

Package ‘StAMPP’

September 1, 2014

Type Package

Title Statistical Analysis of Mixed Ploidy Populations

Depends R (>= 2.14.0), pegas

Imports parallel, doParallel, foreach, adegenet

Version 1.3

Date 2014-08-25

Author LW Pembleton

Maintainer LW Pembleton <luke.pembleton@depi.vic.gov.au>

Description

StAMPP allows users to calculate pairwise Nei's Genetic Distances (Nei 1972), pairwise Fixation Indexes (Fst) (Weir & Cockerham 1984) and also Genomic Relationship matrixes following Yang et al. (2010) in mixed and single ploidy populations. Bootstrapping across loci is implemented during Fst calculation to generate confidence intervals and p-values around pairwise Fst values. StAMPP utilises SNP genotype data of any ploidy level (with the ability to handle missing data) and is coded to utilise multithreading where available to allow efficient analysis of large datasets. StAMPP is able to handle genotype data from genlight objects allowing integration with other packages such as adegenet. Please refer to LW Pembleton, NOI Coogan & JW Forster, 2013, Molecular Ecology Resources, 13(5), 946-952. doi:10.1111/1755-0998.12129 for the appropriate citation and user manual. Thank you in advance.

License GPL-3

NeedsCompilation no

Repository CRAN

Date/Publication 2014-09-01 12:16:42

R topics documented:

StAMPP-package	2
potato	4
potato.mini	6
stamp2genlight	8
stampAmova	9
stampConvert	10
stampFst	11
stampGmatrix	12
stampNeisD	13
stampPhylip	14

Index	16
--------------	-----------

StAMPP-package	<i>Statistical Analysis of Mixed Ploidy Populations</i>
----------------	---

Description

StAMPP allows users to calculate pairwise Nei's Genetic Distances (Nei 1972), pairwise Fixation Indexes (Fst) (Weir & Cockerham 1984) and also Genomic Relationship matrixes following Yang et al. (2010) in mixed and single ploidy populations. Bootstrapping across loci is implemented during Fst calculation to generate confidence intervals and p-values around pairwise Fst values. StAMPP utilises SNP genotype data of any ploidy level (with the ability to handle missing data) and is coded to utilise multithreading where available to allow efficient analysis of large datasets.

Details

Package: StAMPP
 Type: Package
 Version: 1.1
 Date: 2013-06-06
 License: GPL-3

Biallelic AB genotype or frequency of allele A genotype data can be imported into R and converted into allele frequencies and stored in a data frame using the stampConvert() function. The resulting allele frequency data frame can be used as the genotype information for; the calculation of Nei's genetic distance of populations and individuals using the stampNeisD() function. The resulting genetic distance matrix can be exported in Phylip format with the stampPhylip() function. the calculation of pairwise Fst values, confidence intervals and p-values using the stampFst() function. the calculation of a genomic relationship matrix using the stampGmatrix() function.

All StAMPP functions can also work with genotype data stored in genlight objects rather than the data frame generated by stampConvert()

Please refer to; LW Pembleton, NOI Cogan & FW Forster, 2013, Molecular Ecology Resources, doi:10.1111/1755-0998.12129 for the appropriate citation and user manual. Thank you in advance.

Author(s)

LW Pembleton

Maintainer: LW Pembleton <luke.pembleton@dpi.vic.gov.au>

References

Nei M (1972) Genetic Distance between Populations. *The American Naturalist* 106, 283-292.

Perrier X, Jacquemoud-Collet JP (2006) DARWin - Dissimilarity Analysis and Representation for Windows. Agricultural Research for Development

Weir BS, Cockerham CC (1984) Estimating F Statistics for the Analysis of Population Structure. *Evolution* 38, 1358-1370.

Wright S (1949) The Genetical Structure of Populations. *Annals of Human Genetics* 15, 323-354.

Yang J, Benyamin B, McEvoy BP, et al (2010) Common SNPs explain a large proportion of the heritability for human height. *Nat Genet* 42, 565-569

Paradis E (2010) pegas: an R package for population genetics with an integrated-modular approach. *Bioinformatics* 26, 419-420.

Jombart T, Ahmed I (2011) adegenet 1.3-1: new tools for the analysis of genome-wide SNP data. *Bioinformatics* 27, 3070-3071.

See Also

[stampConvert](#) [stampNeisD](#) [stampFst](#) [stampGmatrix](#) [stampAmova](#) [stampPhylip](#) [stamp2genlight](#)

Examples

```
# import genotype data and convert to allele frequencies
data(potato.mini, package="StAMPP")
potato.freq <- stampConvert(potato.mini, "r")

# Calculate genetic distance between populations
potato.D.pop <- stampNeisD(potato.freq, TRUE)

# Calculate genetic distance between individuals
potato.D.ind <- stampNeisD(potato.freq, FALSE)

# Calculate AMOVA
stampAmova(potato.D.ind, potato.freq, 100)

# Export the genetic distance matrix in Phylip format
stampPhylip(potato.D.pop, file="potato_distance.txt")

# Calculate pairwise Fst values between each population
potato.fst <- stampFst(potato.freq, 100, 95, 1)

# Calculate genomic relationship values between each individual
potato.gmat <- stampGmatrix(potato.freq)

# Convert allele frequency genotype data frame to a genlight object for use in other packages
potato.genlight <- stamp2genlight(potato.freq, TRUE)
```

 potato

Example genotype input format

Description

A data frame containing Solcap potato genotype data in tetraploid and diploid format as an example of the input format required by StAMPP

Usage

```
data(potato)
```

Format

A data frame with 30 observations on the following 48 variables.

Sample a factor with levels ind1 ind106 ind140 ind141 ind142 ind143 ind144 ind15 ind16 ind161 ind162 ind17 ind18 ind2 ind257 ind258 ind259 ind290 ind291 ind292 ind293 ind294 ind295 ind296 ind3 ind4 ind5 ind6 ind94 ind95

Pop a factor with levels 2x_A_Population 4x_B_Population 4x_C_Population

Ploidy a numeric vector

Format a factor with levels BiA

solcap_snp_c1_1 a factor with levels AA AAAA AAAB AABB AB

solcap_snp_c1_1000 a factor with levels AAAB AABB AB AB BB BB BB

solcap_snp_c1_10000 a factor with levels AA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10001 a factor with levels AAAA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10011 a factor with levels AA AB AB BB BB BB

solcap_snp_c1_10012 a factor with levels AAAA AAAB AB BB

solcap_snp_c1_10031 a factor with levels AA AAAA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10042 a factor with levels AB AB BB BB BB BB

solcap_snp_c1_10050 a factor with levels AB BB BB BB

solcap_snp_c1_10054 a factor with levels AA AAAA AAAB AABB AB BB

solcap_snp_c1_10109 a factor with levels AA AAAA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10130 a factor with levels -9 AA AAAA AAAB AABB AB AB BB

solcap_snp_c1_10157 a factor with levels AABB AB AB BB BB BB

solcap_snp_c1_10202 a factor with levels AABB AB AB BB BB BB

solcap_snp_c1_10252 a factor with levels AA AAAA AAAB AABB AB BB

solcap_snp_c1_10253 a factor with levels AA AAAA AAAB AABB AB BB

solcap_snp_c1_10255 a factor with levels AAAB AABB AB AB BB BB BB

solcap_snp_c1_1029 a factor with levels AA AAAB AABB AB BB

solcap_snp_c1_10295 a factor with levels AA AAAA AAAB AABB
solcap_snp_c1_10297 a factor with levels -9 AA AAAB AABB AB AB BB BB
solcap_snp_c1_10351 a factor with levels AAAB AABB AB BB BB
solcap_snp_c1_10384 a factor with levels AA AAAA AAAB AABB AB AB BB
solcap_snp_c1_10397 a factor with levels AA AAAA AAAB AABB AB BB
solcap_snp_c1_10457 a factor with levels AAAB AABB AB BB BB
solcap_snp_c1_10491 a factor with levels AABB AB AB BB BB
solcap_snp_c1_10492 a factor with levels AABB AB AB BB BB
solcap_snp_c1_10494 a factor with levels AAAA AAAB AB BB BB
solcap_snp_c1_10579 a factor with levels AA AAAB AABB AB AB BB BB
solcap_snp_c1_10646 a factor with levels AA AAAA AAAB AABB
solcap_snp_c1_10669 a factor with levels AA AAAA AAAB AABB AB
solcap_snp_c1_10715 a factor with levels AA AAAA AAAB AABB AB BB
solcap_snp_c1_10737 a factor with levels AABB AB AB BB BB
solcap_snp_c1_10743 a factor with levels -9 AABB AB AB BB BB
solcap_snp_c1_10762 a factor with levels AA AAAA AAAB AABB AB AB BB
solcap_snp_c1_10855 a factor with levels AA AAAB AABB AB AB BB BB
solcap_snp_c1_10873 a factor with levels AA AAAA AAAB AABB AB AB BB BB
solcap_snp_c1_10879 a factor with levels -9 AAAB AABB AB AB BB BB
solcap_snp_c1_10900 a factor with levels AA AAAA AAAB AABB AB AB BB
solcap_snp_c1_10932 a factor with levels AA AAAA AAAB AABB AB BB
solcap_snp_c1_1094 a factor with levels AAAA AAAB AABB AB AB BB BB
solcap_snp_c1_11137 a factor with levels AA AAAA AAAB AABB AB BB
solcap_snp_c1_11144 a factor with levels AA AAAA AAAB AABB AB BB
solcap_snp_c1_11196 a factor with levels AA AAAA AAAB AABB AB
solcap_snp_c1_11206 a factor with levels AAAB AABB AB BB BB

Source

The example genotype data is a subset of data from the publically available Solcap potato dataset (http://solcap.msu.edu/potato_infinium.shtml) which was re-scored in GenomeStudio in diploid and tetraploid formats

Examples

```
data(potato, package="StAMPP")
```

 potato.mini

Smaller example genotype input format

Description

A data frame containing Solcap potato genotype data in tetraploid and diploid format as an small example of the input format required by StAMPP

Usage

```
data(potato.mini)
```

Format

A data frame with 6 observations on the following 48 variables.

Sample a factor with levels ind1 ind106 ind140 ind141 ind142 ind143 ind144 ind15 ind16 ind161 ind162 ind17 ind18 ind2 ind257 ind258 ind259 ind290 ind291 ind292 ind293 ind294 ind295 ind296 ind3 ind4 ind5 ind6 ind94 ind95

Pop a factor with levels 2x_A_Population 4x_B_Population 4x_C_Population

Ploidy a numeric vector

Format a factor with levels BiA

solcap_snp_c1_1 a factor with levels AA AAAA AAAB AABB AB

solcap_snp_c1_1000 a factor with levels AAAB AABB AB AB BB BB BB

solcap_snp_c1_10000 a factor with levels AA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10001 a factor with levels AAAA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10011 a factor with levels AA AB AB BB BB BB

solcap_snp_c1_10012 a factor with levels AAAA AAAB AB BB

solcap_snp_c1_10031 a factor with levels AA AAAA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10042 a factor with levels AB AB BB BB BB BB

solcap_snp_c1_10050 a factor with levels AB BB BB BB

solcap_snp_c1_10054 a factor with levels AA AAAA AAAB AABB AB BB

solcap_snp_c1_10109 a factor with levels AA AAAA AAAB AABB AB AB BB BB BB

solcap_snp_c1_10130 a factor with levels -9 AA AAAA AAAB AABB AB AB BB

solcap_snp_c1_10157 a factor with levels AABB AB AB BB BB BB

solcap_snp_c1_10202 a factor with levels AABB AB AB BB BB BB

solcap_snp_c1_10252 a factor with levels AA AAAA AAAB AABB AB BB

solcap_snp_c1_10253 a factor with levels AA AAAA AAAB AABB AB BB

solcap_snp_c1_10255 a factor with levels AAAB AABB AB AB BB BB BB

solcap_snp_c1_1029 a factor with levels AA AAAB AABB AB BB

solcap_snp_c1_10295 a factor with levels AA AAAA AAAB AABB
 solcap_snp_c1_10297 a factor with levels -9 AA AAAB AABB AB AB BB BB
 solcap_snp_c1_10351 a factor with levels AAAB AABB AB BB BB
 solcap_snp_c1_10384 a factor with levels AA AAAA AAAB AABB AB AB BB
 solcap_snp_c1_10397 a factor with levels AA AAAA AAAB AABB AB BB
 solcap_snp_c1_10457 a factor with levels AAAB AABB AB BB BB
 solcap_snp_c1_10491 a factor with levels AABB AB AB BB BB
 solcap_snp_c1_10492 a factor with levels AABB AB AB BB BB
 solcap_snp_c1_10494 a factor with levels AAAA AAAB AB BB BB
 solcap_snp_c1_10579 a factor with levels AA AAAB AABB AB AB BB BB
 solcap_snp_c1_10646 a factor with levels AA AAAA AAAB AABB
 solcap_snp_c1_10669 a factor with levels AA AAAA AAAB AABB AB
 solcap_snp_c1_10715 a factor with levels AA AAAA AAAB AABB AB BB
 solcap_snp_c1_10737 a factor with levels AABB AB AB BB BB
 solcap_snp_c1_10743 a factor with levels -9 AABB AB AB BB BB
 solcap_snp_c1_10762 a factor with levels AA AAAA AAAB AABB AB AB BB
 solcap_snp_c1_10855 a factor with levels AA AAAB AABB AB AB BB BB
 solcap_snp_c1_10873 a factor with levels AA AAAA AAAB AABB AB AB BB BB
 solcap_snp_c1_10879 a factor with levels -9 AAAB AABB AB AB BB BB
 solcap_snp_c1_10900 a factor with levels AA AAAA AAAB AABB AB AB BB
 solcap_snp_c1_10932 a factor with levels AA AAAA AAAB AABB AB BB
 solcap_snp_c1_1094 a factor with levels AAAA AAAB AABB AB AB BB BB
 solcap_snp_c1_11137 a factor with levels AA AAAA AAAB AABB AB BB
 solcap_snp_c1_11144 a factor with levels AA AAAA AAAB AABB AB BB
 solcap_snp_c1_11196 a factor with levels AA AAAA AAAB AABB AB
 solcap_snp_c1_11206 a factor with levels AAAB AABB AB BB BB

Source

The example genotype data is a subset of data from the publically available Solcap potato dataset (http://solcap.msu.edu/potato_infinium.shtml) which was re-scored in GenomeStudio in diploid and tetraploid formats

Examples

```
data(potato.mini, package="StAMPP")
```

`stamp2genlight`*Convert StAMPP genotype data to genlight object*

Description

Converts a StAMPP formatted allele frequency data frame generated from the stampConvert function to a genlight object for use in other packages

Usage

```
stamp2genlight(geno, pop = TRUE)
```

Arguments

<code>geno</code>	a data frame containing allele frequency data generated from stampConvert
<code>pop</code>	logical. True if population IDs are present in the StAMPP genotype data, False if population IDs are absent.

Details

StAMPP only exports to genlight objects as they are able to handle mixed ploidy datasets unlike genpop and genloci objects.

The genlight object allows the intergration between StAMPP and other common R packages such as ADEGENET

Value

A object of class genlight which contains genotype data, individual IDs, population IDs (if present) and ploidy levels.

Author(s)

LW Pembleton

Examples

```
# import genotype data and convert to allele frequcies
data(potato.mini, package="StAMPP")
potato.freq <- stampConvert(potato.mini, "r")

# Convert the StAMPP formatted allele frequency data frame to a genlight object
potato.genlight <- stamp2genlight(potato.freq, TRUE)
```


Description

Calculates an AMOVA based on the genetic distance matrix from stampNeisD() using the amova() function from the package PEGAS for exploring within and between population variation.

Usage

```
stampAmova(dist.mat, geno, perm = 100)
```

Arguments

dist.mat	the matrix of genetic distances between individuals generated from stampNeisD()
geno	a data frame containing allele frequency data generated from stampConvert, or a genlight object containing genotype data, individual IDs, population IDs and ploidy levels.
perm	the number of permutations for the tests of hypotheses.

Details

Uses the formula distance ~ populations, to calculate an AMOVA for population differentiation and within & between population variation.

This function uses the amova function from the PEGAS package.

Value

An object of class "amova" which is a list containing a table of sum of square deviations (SSD), mean square deviations (MSD) and the number of degrees of freedom as well as the variance components

Author(s)

LW Pembleton

References

Paradis E (2010) pegas: an R package for population genetics with an integrated-modular approach. *Bioinformatics* 26, 419-420.

Examples

```
# import genotype data and convert to allele frequencies
data(potato.mini, package="StAMPP")
potato.freq <- stamppConvert(potato.mini, "r")

# Calculate genetic distance between individuals
potato.D.ind <- stamppNeisD(potato.freq, FALSE)

# Calculate AMOVA
stamppAmove(potato.D.ind, potato.freq, 100)
```

stamppConvert

Import and Convert

Description

Imports biallelic AB formatted or allele A frequency genotype data. If the data is imported in biallelic AB format this function also converts it to allele frequencies.

Usage

```
stamppConvert(genotype.file, type = "csv")
```

Arguments

`genotype.file` the genotype input file. This should be a R matrix object or a file path for a csv file containing the genotype data in either biallelic AB format or allele 'A' frequency format, or a genlight object containing genotype data.

`type` the type of file the genotype data is being imported from; "csv" = comma separated file, "r" = data frame in the R workspace, "genlight" = genlight object.

Details

If importing from a csv file or the R workspace; The first column of the genotype input file or matrix should contain genotype IDs, the second column should contain population IDs, the third column should contain a number indicating the ploidy level of each individual (ie 2=diploid, 4=tetraploid etc.), the fourth column should contain a code indicating the format of the genotype data ('BiA' for biallelic AB format, 'freq' for allele A frequency format) and each consecutive column should contain the genotype at a give locus coded in the corresponding format (for example Biallelic format; AA or AB or BB or AAAB or AAAA etc. or allele A freq format; 1, 0.5, 0, 0.75 etc). Each column should have a header (eg. Inds, Pop, Format, Locus1, Locus2 etc.).

Value

An object of class data.frame which contains allele frequency data for use in other StAMPP functions

Author(s)

LW Pembleton

Examples

```
# Import example data into the R workspace
data(potato.mini, package="StAMPP")

# Convert to allele frequencies
potato.freq <- stampConvert(potato.mini, "r")
```

stampFst

*Fst Computation***Description**

This function calculates pairwise Fst values along with confidence intervals and p-values between populations according to the method proposed by Wright(1949) and updated by Weir and Cockerham (1984).

Usage

```
stampFst(geno, nboots = 100, percent = 95, nclusters = 1)
```

Arguments

geno	a data frame containing allele frequency data generated from stampConvert, or a genlight object containing genotype data, individual IDs, population IDs and ploidy levels.
nboots	number of bootstraps to perform across loci to generate confidence intervals and p-values
percent	the percentile to calculate the confidence interval around
nclusters	the number of processor threads or cores to use during calculations.

Details

If possible, using multiple processing threads or cores is recommended to assist in calculating Fst values over a large number of bootstraps.

Value

An object list with the components:

Fsts	a matrix of pairwise Fst values between populations
Pvalues	a matrix of p-values for each of the pairwise Fst values contained in the 'Fsts' matrix

Bootstraps a dataframe of each Fst value generated during Bootstrapping and the associated confidence intervals

If nboots<2, no bootstrapping is performed and therefore only a matrix of Fst values is returned.

Author(s)

LW Pembleton

References

Wright S (1949) The Genetical Structure of Populations. *Annals of Human Genetics* 15, 323-354.

Weir BS, Cockerham CC (1984) Estimating F Statistics for the ANalysis of Population Structure. *Evolution* 38, 1358-1370.

Examples

```
# import genotype data and convert to allele frequencies
data(potato.mini, package="StAMPP")
potato.freq <- stampConvert(potato.mini, "r")

# Calculate pairwise Fst values between each population
potato.fst <- stampFst(potato.freq, 100, 95, 1)
```

stampGmatrix

Genomic Relationship Calculation

Description

This function calculates a genomic relationship matrix following the method described by Yang et al (2010).

Usage

```
stampGmatrix(geno)
```

Arguments

geno a data frame containing allele frequency data generated from stampConvert, or a genlight object containing genotype data, individual IDs, population IDs and ploidy levels.

Value

An object of class matrix which contains the genomic relationship values between each individual.

Author(s)

LW Pembleton

References

Yang J, Benyamin B, McEvoy BP, et al (2010) Common SNPs explain a large proportion of the heritability for human height. Nat Genet 42, 565-569

Examples

```
# import genotype data and convert to allele frequencies
data(potato.mini, package="StAMPP")
potato.freq <- stampConvert(potato.mini, "r")

# Calculate genomic relationship values between each individual
potato.fst <- stampGmatrix(potato.freq)
```

stampNeisD

Genetic Distance Calculation

Description

This function calculates Nei's genetic distance (Nei 1972) between populations or individuals.

Usage

```
stampNeisD(geno, pop = TRUE)
```

Arguments

geno	a data frame containing allele frequency data generated from stampConvert, or a genlight object containing genotype data, individual IDs, population IDs and ploidy levels.
pop	logical. True if genetic distance should be calculated between populations, false if it should be calculated between individual.

Value

A object of class matrix which contains the genetic distance between each population or individual.

Author(s)

LW Pembleton

References

Nei M (1972) Genetic Distance between Populations. The American Naturalist 106, 283-292.

Examples

```
# import genotype data and convert to allele frequencies
data(potato.mini, package="StAMPP")
potato.freq <- stampConvert(potato.mini, "r")

# Calculate genetic distance between individuals
potato.D.ind <- stampNeisD(potato.freq, FALSE)

# Calculate genetic distance between populations
potato.D.pop <- stampNeisD(potato.freq, TRUE)
```

stampPhylip

Export to Phylip Format

Description

Converts the genetic distance matrix generated with stampNeisD into Phylip format and exports it as a text file.

Usage

```
stampPhylip(distance.mat, file = "")
```

Arguments

distance.mat	the matrix containing the genetic distances generated from stampNeisD to be converted into Phylip format
file	the file path and name to save the Phylip format matrix as

Details

The exported Phylip formatted text file can be easily imported into software packages such as DARWin (Perrier & Jacquemound-Collet 2006) to be used to generate neighbour joining trees.

Author(s)

LW Pembleton

References

Perrier X, Jacquemound-Collet JP (2006) DARWin - Dissimilarity Analysis and Representation for Windows. Agricultural Research for Development

Examples

```
# import genotype data and convert to allele frequencies
data(potato.mini, package="StAMPP")
potato.freq <- stamppConvert(potato.mini, "r")

# Calculate genetic distance between populations
potato.D.pop <- stamppNeisD(potato.freq, TRUE)

# Export the genetic distance matrix in Phylip format
stamppPhylip(potato.D.pop, file="potato_distance.txt")
```

Index

*Topic **datasets**

potato, [4](#)

potato.mini, [6](#)

*Topic **package**

StAMPP-package, [2](#)

potato, [4](#)

potato.mini, [6](#)

StAMPP (StAMPP-package), [2](#)

StAMPP-package, [2](#)

stamp2genlight, [3](#), [8](#)

stampAmova, [3](#), [9](#)

stampConvert, [3](#), [10](#)

stampFst, [3](#), [11](#)

stampGmatrix, [3](#), [12](#)

stampNeisD, [3](#), [13](#)

stampPhylip, [3](#), [14](#)