

Package ‘popgen’

January 2, 2012

Version 1.0-1

Date 2011-06-19

Title Statistical and Population Genetics

Author J L Marchini <marchini@stats.ox.ac.uk>

Maintainer J L Marchini <marchini@stats.ox.ac.uk>

Depends cluster

Description A package that implements a variety of statistical and population genetic methodology.

License GPL (>= 2)

URL <http://www.stats.ox.ac.uk/~marchini/software.html>

Repository CRAN

Date/Publication 2011-06-20 07:16:32

R topics documented:

LDmat	2
nps	3
nps.plot	4
popdiv	5
popdiv.convert	8
ps	9
simMD	13

Index	15
--------------	-----------

LDmat

Calculate LD measures for haplotype data.

Description

Linkage disequilibrium measures (D' and r^2) for a set of genotypes or haplotypes.

Usage

```
LDmat(mat, typ = c("genotype", "haplotype"), pos = NULL, plotmat = TRUE)
```

Arguments

mat	An $n \times p$ matrix of genotypes or haplotypes. In the case of genotypes the matrix should contain a row for each individual. The genotypes should be coded 0, 1 and 2 and use -1 for missing. In the case of haplotypes the matrix should contain a row for each haplotype. The alleles at each site should be coded 0 and 1 and use -1 for missing.
typ	Flag determining whether the data matrix contains genotypes or haplotypes.
pos	Vector with the positions of the markers in the range [0, 1].
plotmat	Flag specifying whether to plot the LD measures.

Value

Returns a $p \times p$ matrix whose upper triangle contains the values of D' between pairs of markers and the lower triangle contains the values of r^2 between the pairs of markers. If `plotmat == TRUE` then an image is plotted in which the upper left segment is D' and the bottom right half is r^2 .

Author(s)

Jonathan Marchini

References

B. S. Weir (1996) Genetic Data Analysis II. Sinauer

nps

*Non-parametric clustering of SNP genotype data.***Description**

Uses classical hierarchical clustering methods and the Gap statistic to identify clusters of genotype data.

Usage

```
nps(X, dmetric = "manhattan", method = "mcquitty", gap.n = 100)
```

Arguments

X	Data array of dimension $c(n, 2, L)$ where n is the number of people and L is the number of loci. Entry $[i, j, k]$ contains the j th allele of the i th person at the k th locus. The function only handles SNP data and the alleles should be coded 0 and 1.
dmetric	The distance measure to be used. This must be one of "euclidean", "maximum", "manhattan", "canberra" or "binary". Any unambiguous substring can be given. For genotype data we recommend the use of the "manhattan" metric.
method	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid".
gap.n	The number of simulations used to compute the Gap statistic.

Details

The `nps` function can be used to cluster genotype data without making any parametric assumptions about the properties of the clusters themselves. In order to do this we use the existing clustering functions available in R and implement the recently proposed Gap statistic [1] to estimate the number of clusters in a given dataset. This method compares the average within-cluster dissimilarity to its expected value under an appropriate null reference distribution for each partition in the hierarchy. In practice the expectation is approximated using samples from the null reference distribution. The null reference distribution used assumes that all the datapoints lie in just one cluster.

Value

A list with components

num	The number of clusters selected by using the Gap statistic.
clust	The output of the function <code>hclust</code> applied to the real dataset.
gap	A vector of Gap statistics for each value of k .
sk	The adjusted standard error of each Gap statistic.

Author(s)

Jonathan Marchini

References

R. Tibshirani and G. Walther and T. Hastie (2001) Estimating the number of clusters in a dataset via the gap statistic. JRSS (B)

See Also

[nps.plot](#), [ps](#)

Examples

```
## NB. The value of gap.n is set unrealistically low in this example so
## that the examples run efficiently at compile time. We suggest
## gap.n be set to 100 (the default) for reliable results.

X <- simMD(100, 3, 100, p = NULL, c.vec1 = c(0.1, 0.2, 0.3), c.vec2 = 1, ac = 2, beta = 1)

res <- nps(X - 1, gap.n = 2)

nps.plot(res, k.max = 2)
```

nps.plot

Gap statistic plot for nps function

Description

Plot the Gap statistic for different numbers of clusters using the output of the SNP genotype clustering function `nps`.

Usage

```
nps.plot(res, k.max)
```

Arguments

<code>res</code>	Object produced by the function <code>nps</code>
<code>k.max</code>	Maximum number of clusters to plot the Gap statistic for.

Details

Plot the Gap statistic for different numbers of clusters using the output of the SNP genotype clustering function `nps`. A red line is used to indicate the selected number of clusters.

Author(s)

Jonathan Marchini

References

R. Tibshirani and G. Walther and T. Hastie (2001) Estimating the number of clusters in a dataset via the gap statistic. JRSS (B)

See Also[nps](#)

popdiv

Population diversity function

Description

This function assesses between population variability using unlinked multi-locus multi-allelic genotype data from the populations of interest.

Usage

```
popdiv(L, P, NUMA, N, X, burnin = 10000, iter = 10000, m = 300, csd = 0.01, outc = FALSE)
```

Arguments

L	Number of loci
P	Number of populations
NUMA	Vector of the number of alleles at each loci
N	LxP matrix containing the number of genotypes in each population at each locus.
X	Matrix with L rows, one for each locus. Each row contains a list of allele counts from each population. For example, if there are 2 populations and the locus has 3 alleles then the first 3 entries in the row specify the allele counts in population 1 and the next 3 entries specify the allele counts in population 2. The rest of the row is set to 0.
burnin	Number of burn-in iterations in MCMC run
iter	Number of sampling iterations in MCMC run
m	Scale parameter of Dirichlet distribution used to update global allele frequencies (p).
csd	Standard deviation of Normal distribution used to update the population variance parameters (c).
outc	Flag that specifies whether the samples of c are returned (Logical)

Details

The model fitted has a Multinomial-Dirichlet structure with a specific parameterisation suitable for the situation.

x_{il_j} = number of type j alleles observed at locus l in population i .

n_{il_j} = number of type j alleles observed at locus l in population i .

α_{il_j} = locus l type j allele frequency of the population i .

π_{l_j} = locus l type j allele frequency of the ‘global’ population.

c_i = variance parameter of population i .

J_l = number of alleles at locus l .

$\{x_{il_1}, \dots, x_{il_{J_l}}\} \sim \text{Multinom}(\alpha_{il_1}, \dots, \alpha_{il_{J_l}})$

$\{\alpha_{il_1}, \dots, \alpha_{il_{J_l}}\} \sim \text{Dirichlet}(\pi_{l_1} \frac{(1-c_i)}{c_i}, \dots, \pi_{l_{J_l}} \frac{(1-c_i)}{c_i})$

The model is specified in a Bayesian framework by placing uniform priors on the c and π parameters.

A Metropolis-Hastings algorithm is used to sample from the posterior distribution of the parameters c and π .

The proposal distribution for the c parameters is

$c^{\text{new}} \sim N(c^{\text{old}}, \text{csd}^2)$

The proposal distribution for the π parameter is

$\pi_l^{\text{new}} \sim \text{Dirichlet}(m\pi_{l_1}, \dots, m\pi_{l_{J_l}})$

The conjugate nature of the model allows the α parameters to be integrated out of the likelihood so that these parameters are not sampled.

The c parameters are analogous to Fst estimates which are traditional measures of population diversity. See [1] for more details.

Value

List with components

C.sample	An (iter x P) matrix containing the samples of the c parameters.
muc	The posterior mean of the c parameters.
sdc	The posterior standard deviation of the c parameters.
mup	The posterior mean of the π parameters.
sdp	The posterior standard deviation of the π parameters.

Author(s)

Jonathan Marchini

References

- [1] Nicholson et al. (2002), Assessing population differentiation and isolation from single-nucleotide polymorphism data. *JRSS(B)*, 64, 695–715
- [2] Marchini and Cardon (2002) Discussion of Nicholson et al. (2002). *JRSS(B)*, 64, 740–741
- [3] Nichols and Balding (1995) A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica*, 96, 3–12.

Examples

```
## Note : In the examples below the number of burn-in and sampling
## iterations is set to 100 so that these examples run in a reasonable
## time. In practice we recommend setting these parameters much higher so
## that convergence of the Markov chain to the stationary distribution is
## more likely.

#####
## EXAMPLE 1 ##
#####

## SNP dataset from Nicholson et al. (2002). 66 loci, 4 populations

ex1 <- read.table(paste(.path.package("popgen"), "example_data1", sep = .Platform$file.sep))
ex1 <- matrix(unlist(ex1), 66, 13, byrow = FALSE)
NUMA <- ex1[,1]
N <- ex1[, 2:5]
X <- ex1[, 6:13]
L <- 66
P <- 4

res <- popdiv(L, P, NUMA, N, X, burnin = 100, iter = 100, m = 10, csd = 0.01, outc = TRUE)

plot(c(0, 100), c(0, max(res$C.sample)), type = "n")
lines(1:100, res$C.sample[, 1])
lines(1:100, res$C.sample[, 2], col = 2)
lines(1:100, res$C.sample[, 3], col = 3)
lines(1:100, res$C.sample[, 4], col = 4)

#####
## EXAMPLE 2 ##
#####

## Simulated dataset from the Multinomial-Dirichlet model.
## 100 loci, 3 populations.
## Variance parameters set to 0.01, 0.02, 0.04.
## Each loci has a variable number of alleles (between 3 and 8).
## Dataset stored in the following format :-
## - each individuals genotypes are stored in 2 rows of the file.
## - the first entry in each row specifies the individuals population.
## - each row has (L + 1) entries where L is the number of loci.
## To examine the datafile use edit(ex2)
```

```

ex2 <- read.table(system.file("example_data2", package = "popgen"))
X <- matrix(unlist(ex2), nrow(ex2), ncol(ex2), byrow = FALSE)

X1 <- popdiv.convert(X)

res1 <- popdiv(L = nrow(X1$N), P = ncol(X1$N), NUMA = X1$NUMA, N = X1$N, X = X1$X, burnin = 100, iter = 100, m = 10, c

plot(c(0, 100), c(0, max(res1$C.sample)), type = "n")
lines(1:100, res1$C.sample[, 1])
lines(1:100, res1$C.sample[, 2], col = 2)
lines(1:100, res1$C.sample[, 3], col = 3)

```

popdiv.convert

Convert a dataset for use in function popdif.

Description

Convert a matrix (in which the multi-locus genotypes of each individual are stored in 2 rows) to a format used by the function popdif.

Usage

```
popdiv.convert(x, miss = 0)
```

Arguments

x	(2 * N, L + 1) matrix where N is the number of individuals and L is the number of loci. Each individual's genotypes are contained in 2 rows. The first entry of each specifies the individuals population. Each row has (L + 1) entries where L is the number of loci.
miss	The code for missing values.

Value

List with components

NUMA	Vector of the number of alleles at each loci
N	LxP matrix containing the number of genotypes in each population at each locus.
X	Matrix with L rows, one for each locus. Each row contains a list of allele counts from each population. For example, if there are 2 populations and the locus has 3 alleles then the first 3 entries in the row specify the allele counts in population 1 and the next 3 entries specify the allele counts in population 2. The rest of the row is set to 0.

Author(s)

Jonathan Marchini

See Also[popdiv](#)**Examples**

```
ex <- read.table(system.file("example_data2", package = "popgen"))
X <- matrix(unlist(ex), nrow(ex), ncol(ex), byrow = FALSE)

X1 <- popdiv.convert(X)
```

ps

*Parametric model-based inference of population structure.***Description**

Parametric model-based inference of population structure for unlinked SNP datasets.

Usage

```
ps(X, K, burn.in = 10^4, num.sample = 10^4, thin = 1, alpha.start = 1.0, alpha.sd = 0.05, alpha.step = 10,
```

Arguments

X	Data array of dimension $c(n, 2, L)$ where n is the number of people and L is the number of loci. Entry $[i, j, k]$ contains the j th allele of the i th person at the k th locus.
K	The number of populations.
burn.in	The length of the burn-in.
num.sample	The number of sampling iterations after the burn-in.
thin	The thinning frequency (default = 1)
alpha.start	The initial value of the admixture parameter α (default = 1)
alpha.sd	The standard deviation of the proposal distribution used to update α (default = 0.05)
alpha.step	The frequency at which α is updated (default = 10)
alpha.max	The upper limit for α (default = 20)
z.init	Array with the same dimension as X which contains the initial values of the population labels for each allele at each locus in each person (the default is NULL which implies that Z is set randomly)

<code>sample.type</code>	Vector of 0/1 flags that specify which parameter samples are recorded. The order of the flags is (P, Q, Z, pi, c). The default is to sample all the parameters i.e. <code>sample.type = rep(1, 5)</code> .
<code>na.lab</code>	Label for missing data.
<code>c.init</code>	Vector of length K containing the initial values of the c parameters. If NULL then set randomly.
<code>pi.init</code>	Vector of length L containing the initial values of the pi parameters. If NULL then set randomly.
<code>c.prior.mu</code>	Mean of the prior for c (default = 0.01)
<code>c.prior.sd</code>	Standard deviation of the prior for pi (default = 0.05)
<code>c.sd</code>	The standard deviation of the proposal distribution used to update c (default = 0.01)
<code>fix.alpha</code>	Flag that specifies that alpha be fixed at its initial value.
<code>fix.pi</code>	Flag that specifies that pi be fixed at its initial value.
<code>popdif.flag</code>	Flag to turn on the model of population differentiation used in the function <code>popdif</code> .
<code>beta</code>	Parameter that scales the dirichlet proposal for pi (default = 1)
<code>fix.z</code>	Flag that specifies that Z be fixed at its initial value i.e. the structure is fixed and the other parameters inferred.

Details

The `ps` function is an implementation of a model-based clustering algorithm for genotype data developed in [1]. The implementation here includes an extension which uses a more realistic prior structure for each subpopulations allele frequencies (as developed in [2], [3], and [4]). At present the function should only be used with unlinked SNP data.

The model is specified in a Bayesian framework and samples from the posterior distribution are obtained from an MCMC algorithm. Details of the algorithm can be found in the references below.

The `ps` function clusters a given dataset for a given number of populations K and it is often desirable to obtain inference on the number K. In [1] a novel approximation to the evidence was used to approximate the posterior distribution on K by combining the results of several runs. This can be achieved by combining the results of several runs using different values of K. The details are given in [1] and an example is given below in the examples section.

Value

List with components

Alpha	Sample of the alpha parameter.
Z	Mean of the (n, 2, L) array Z i.e. the mean population labels for each allele.
P	Mean of the (K, 2, L) array P i.e. the mean allele frequencies at each locus in each population.
Q	Mean of the (n, K) matrix Q i.e. the mean proportion of each persons genome from each population.

mu	Mean value of log-likelihood
v	Variance of log-likelihood
log.PX.K	Estimated log-probability of the data given K
PX	Sample of the log-likelihood
pi	Sample of the pi parameters.
c	Sample of the c parameters.

Author(s)

Jonathan Marchini

References

- [1] Pritchard, Stephens and Donnelly (2000) Inference of population structure using multilocus genotype data. *Genetics* 155:945-959
- [2] Nichols and Balding (1995) A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica*, 96, 3–12.
Marchini and Cardon (2002) Discussion of Nicholson et al. (2002). *JRSS(B)*, 64, 740–741
- [3] Nicholson et al. (2002), Assessing population differentiation and isolation from single-nucleotide polymorphism data. *JRSS(B)*, 64, 695–715
- [4] Marchini and Cardon (2002) Discussion of Nicholson et al. (2002). *JRSS(B)*, 64, 740–741

Examples

```
X <- simMD(60, 3, 100, p = NULL, c.vec1 = c(0.1, 0.2, 0.3), c.vec2 =
1, ac = 2, beta = 1)

## infer the population structure
## NB in this example the burn-in and no. of iterations have been set
## way too low so that the examples run in a reasonable time at time
## of installation. To get reliable answers set these parameters
## much higher and check convergence of the chain using multiple runs.

res2 <- ps(X,
  2,
  num.sample = 100,
  burn.in = 100,
  na.lab = (-1),
  c.init = rep(0.1, 2),
  c.prior.mu = 0.01,
  c.prior.sd = 0.05,
  c.sd = 0.05,
  alpha.start = 1.0,
  fix.alpha = FALSE,
  fix.pi = FALSE,
  popdif.flag = TRUE,
  beta = 1,
  z.init = NULL,
  fix.z = FALSE)
```

```
res3 <- ps(X,
  3,
  num.sample = 100,
  burn.in = 100,
  na.lab = (-1),
  c.init = rep(0.1, 3),
  c.prior.mu = 0.01,
  c.prior.sd = 0.05,
  c.sd = 0.05,
  alpha.start = 1.0,
  fix.alpha = FALSE,
  fix.pi = FALSE,
  popdif.flag = TRUE,
  beta = 1,
  z.init = NULL,
  fix.z = FALSE)

res4 <- ps(X,
  4,
  num.sample = 100,
  burn.in = 100,
  na.lab = (-1),
  c.init = rep(0.1, 4),
  c.prior.mu = 0.01,
  c.prior.sd = 0.05,
  c.sd = 0.05,
  alpha.start = 1.0,
  fix.alpha = FALSE,
  fix.pi = FALSE,
  popdif.flag = TRUE,
  beta = 1,
  z.init = NULL,
  fix.z = FALSE)

k.dist <- c(res2$log.PX.K, res3$log.PX.K, res4$log.PX.K)
k.dist <- k.dist - max(k.dist)
k.dist <- exp(k.dist) / sum(exp(k.dist))
k.dist

## print the mean values of the c parameters
mean(res3$c[, 1])
mean(res3$c[, 2])
mean(res3$c[, 3])

## print out the results
par(mfcol = c(3, 3))
hist(res3$c[, 1], n = 100, xlim = c(0, 0.4))
hist(res3$c[, 2], n = 100, xlim = c(0, 0.4))
hist(res3$c[, 3], n = 100, xlim = c(0, 0.4))

plot(res3$c[, 1], typ = "l")
plot(res3$c[, 2], typ = "l")
```

```
plot(res3$c[, 3], typ = "l")
image(res3$Q)
```

simMD	<i>Simulate multi-population unlinked genotype data from a Multinomial-Dirichlet model.</i>
-------	---

Description

Simulate multi-population unlinked genotype data from a Multinomial-Dirichlet model.

Usage

```
simMD(N, P, L, p = NULL, c.vec1, c.vec2 = 1, ac = 2, beta = 1)
```

Arguments

N	The number of people per population.
P	The number of populations.
L	The number of unlinked loci.
p	A (ac x L) with column 1 containing the ac global allele frequencies for locus 1. If NULL the frequencies are generated at random from a Dirichlet distribution with parameter beta (default is beta = 1 which is a uniform distribution).
c.vec1	A vector of length P which contains the level 1 variance parameter for each subpopulation.
c.vec2	An (optional) vector which contains the level 2 variance parameters for each subpopulation of the level 1 subpopulation.
ac	The number of alleles at each locus.
beta	The parameter of Dirichlet distribution used to simulate the global allele frequencies.

Details

The data is simulated from a Multinomial-Dirichlet structure with a specific parameterisation suitable for the situation.

x_{ipal} = Allele of the a th chromosome from the i th person in the p th population at locus l .

α_{plj} = locus l type j allele frequency of the population p .

π_{lj} = locus l type j allele frequency of the 'global' population.

c_p = variance parameter of population p .

ac = number of alleles at each locus.

$x_{ipal} \sim \text{Multinom}(\alpha_{pl_1}, \dots, \alpha_{pl_{ac}})$

$\{\alpha_{pl_1}, \dots, \alpha_{pl_{ac}}\} \sim \text{Dirichlet}(\pi_{l_1} \frac{(1-c_p)}{c_p}, \dots, \pi_{l_{ac}} \frac{(1-c_p)}{c_p})$

If the global allele frequencies are not specified then $\{\pi_{l_1}, \dots, \pi_{l_{ac}}\} \sim \text{Dirichlet}(\beta, \dots, \beta)$

Value

If the `c.vec2` parameter is left as its default value then the data is simulated from the standard multinomial-dirichlet model (see above) and the result is stored in an array with dimensions $(N * P, 2, L)$ containing the data. Entry $((p - 1) * P + i, a, l)$ contains the simulated allele from the i th person in the p th population of the a th chromosome at the l th locus.

If the `c.vec2` vector is specified then each of the P populations has $\text{length}(\text{c.vec2})$ subpopulations with the `c.vec2` parameters containing the variance parameters of the subpopulations.

Author(s)

Jonathan Marchini

References

Nicholson et al. (2002), Assessing population differentiation and isolation from single-nucleotide polymorphism data. *JRSS(B)*, 64, 695–715

Marchini and Cardon (2002) Discussion of Nicholson et al. (2002). *JRSS(B)*, 64, 740–741

Nichols and Balding (1995) A method for quantifying differentiation between populations at multi-allelic loci and its implications for investigating identity and paternity. *Genetica*, 96, 3–12.

Examples

```
X <- simMD(60, 3, 100, p = NULL, c.vec1 = c(0.1, 0.2, 0.3), c.vec2 =  
1, ac = 2, beta = 1)
```

Index

*Topic **utilities**

LDmat, [2](#)

nps, [3](#)

nps.plot, [4](#)

popdiv, [5](#)

popdiv.convert, [8](#)

ps, [9](#)

simMD, [13](#)

LDmat, [2](#)

nps, [3](#), [5](#)

nps.plot, [4](#), [4](#)

popdiv, [5](#), [9](#)

popdiv.convert, [8](#)

ps, [4](#), [9](#)

simMD, [13](#)