Package 'bnpsd'

January 16, 2018

Title Simulate Genotypes from the BN-PSD Admixture Model

Version 1.0.1

Description The Pritchard-Stephens-Donnelly (PSD) admixture model has k intermediate subpopulations from which n individuals draw their alleles dictated by their individual-specific admixture proportions. The BN-PSD model additionally imposes the Balding-Nichols (BN) allele frequency model to the intermediate populations, which therefore evolved independently from a common ancestral population T with subpopulation-specific FST (Wright's fixation index) parameters. The BN-PSD model can be used to yield complex population structures. Method described in Ochoa and Storey (2016) <doi:10.1101/083923>.

Depends

Imports stats

Suggests popkin, testthat, knitr, rmarkdown, RColorBrewer

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1.9000

VignetteBuilder knitr

URL https://github.com/StoreyLab/bnpsd/

BugReports https://github.com/StoreyLab/bnpsd/issues

NeedsCompilation no

Author Alejandro Ochoa [aut, cre],
John D. Storey [aut]

Maintainer Alejandro Ochoa <ochoa@princeton.edu>

Repository CRAN

Date/Publication 2018-01-16 10:07:50 UTC

R topics documented:

---

bnpsd *A package for modeling and simulating an admixed population*

Description

The underlying model is called the BN-PSD admixture model, which combines the Balding-Nichols (BN) allele frequency model for the intermediate subpopulations with the Pritchard-Stephens-Donnelly (PSD) model of individual-specific admixture proportions. The BN-PSD model enables the simulation of complex population structures, ideal for illustrating challenges in kinship coefficient and F_{ST} estimation. Note that simulated loci are drawn independently (in linkage equilibrium).

Author(s)

Maintainer: Alejandro Ochoa <ochoa@princeton.edu>

Authors:

- John D. Storey <jstorey@princeton.edu>

See Also

Useful links:

- https://github.com/StoreyLab/bnpsd/
- Report bugs at https://github.com/StoreyLab/bnpsd/issues

Examples

```
# dimensions of data/model
m <- 10 # number of loci
n <- 5 # number of individuals
k <- 2 # number of intermediate subpops

# define population structure
```

```
F <- c(0.1, 0.3) # FST values for k=2 subpopulations
sigma <- 1 # dispersion parameter of intermediate subpops
Q <- q1d(n, k, sigma) # admixture proportions from 1D geography

# get pop structure parameters of the admixed individuals
Theta <- coanc(Q,F) # the coancestry matrix
Fst <- fst(Q,F) # Fst

# draw all random allele freqs and genotypes
out <- rbnpsd(Q, F, m)
X <- out$X # genotypes
P <- out$P # IAFs (individual-specific AFs)
B <- out$B # intermediate AFs
pAnc <- out$Pa # ancestral AFs

# OR... draw each vector or matrix separately
# provided for additional flexibility
pAnc <- rpanc(m) # "anc"estral AFs
B <- rpint(pAnc, F) # "int"ermediate AFs
P <- rpiaf(B, Q) # "IAF"s (individual-specific AFs)
X <- rgeno(P) # "geno"types
```

---

coanc *Construct the coancestry matrix of an admixture model*

Description

In the most general case, the $n \times n$ coancestry matrix Θ of admixed individuals is determined by the $n \times k$ admixture proportion matrix Q and the $k \times k$ intermediate subpopulation coancestry matrix Ψ, given by

$$\Theta = Q\Psi Q^T$$

In the BN-PSD model Ψ is a diagonal matrix (with F_{ST} values for the intermediate subpopulations along the diagonal, zero values off-diagonal).

Usage

```
coanc(Q, F)
```

Arguments

Q	The $n \times k$ admixture proportion matrix
F	Either the $k \times k$ intermediate subpopulation coancestry matrix (for the complete admixture model), or the length-k vector of intermediate subpopulation F_{ST} values (for the BN-PSD model), or a scalar F_{ST} value shared by all intermediate subpopulations.

Value

The $n \times n$ coancestry matrix Θ

Examples

```
# a trivial case: unadmixed individuals from independent subpopulations
n <- 5 # number of individuals/subpops
Q <- diag(rep.int(1, n)) # unadmixed individuals
F <- 0.2 # equal Fst for all subpops
Theta <- coanc(Q, F) # diagonal coancestry matryx

# a more complicated admixture model
n <- 5 # number of individuals
k <- 2 # number of intermediate subpops
sigma <- 1 # dispersion parameter of intermediate subpops
Q <- q1d(n, k, sigma) # non-trivial admixture proportions
F <- c(0.1, 0.3) # different Fst for each of the k subpops
Theta <- coanc(Q, F) # non-trivial coancestry matrix
```

---

fst *Calculate FST for the admixed individuals*

Description

Given the admixture proportion matrix Q for n individuals and k intermediate subpopulations, the vector of intermediate inbreeding coefficients F (per-subpopulation F_{ST}'s), and weights for individuals, this function returns the F_{ST} of the admixed individuals. This F_{ST} equals the weighted mean of the diagonal of the coancestry matrix (see coanc).

Usage

```
fst(Q, F, w)
```

Arguments

Q	The $n \times k$ admixture proportion matrix
F	The length-k vector of subpopulation inbreeding coefficients
w	The length-n vector of weights for individuals that define F_{ST} (default uniform weights)

Value

The F_{ST} of the admixed individuals

Examples

```
# set desired parameters
n <- 1000 # number of individuals
k <- 10 # number of intermediate subpopulations
s <- 0.5 # desired bias coefficient
sigma <- 1 # for 1D admixture model
# differentiation of intermediate subpopulations
F <- (1:k)/k
# construct final admixture proportions
Q <- q1d(n=n, k=k, sigma=sigma)
# lastly, calculate Fst!!! (uniform weights in this case)
F <- fst(Q, F)
```

---

q1d *Construct admixture proportion matrix for 1D geography*

Description

Assumes k intermediate subpopulations placed along a line at locations $1:k$ spread by random walks, then n individuals sampled equally spaced in $[a, b]$ (default $[0.5, k + 0.5]$) draw their admixture proportions relative to the Normal density that models the random walks of each of these intermediate subpopulations. The spread of the random walks (the σ of the Normal densities) is set to `sigma` if not missing, otherwise σ is found numerically to give the desired bias coefficient `s`, the vector `F` of F_{ST}'s for the intermediate subpopulations up to a scalar factor, and the final F_{ST} of the admixed individuals (see details below).

Usage

```
q1d(n, k, sigma, a = 0.5, b = k + 0.5, s, F, Fst, interval = c(0.1, 10),
  tol = .Machine$double.eps)
```

Arguments

`n`	Number of individuals
`k`	Number of intermediate subpopulations
`sigma`	Spread of intermediate subpopulations (standard deviation of normal densities)
`a`	Location of first individual
`b`	Location of last individual
	OPTIONS FOR BIAS COEFFICIENT VERSION
`s`	The desired bias coefficient, which specifies σ indirectly. Required if `sigma` is missing
`F`	The length-k vector of inbreeding coefficients (or F_{ST}'s) of the intermediate subpopulations, up to a scaling factor (which cancels out in calculations). Required if `sigma` is missing

Fst	The desired final F_{ST} of the admixed individuals. Required if sigma is missing
interval	Restrict the search space of σ to this interval
tol	The numerical tolerance used to declare the solution found

Details

When sigma is missing, the function determines its value using the desired s, F up to a scalar factor, and Fst. Uniform weights for the final generalized F_{ST} are assumed. The scaling factor of the input F is irrelevant because it cancels out in s; after sigma is found, F is rescaled to give the desired final F_{ST}. However, the function stops with a fatal error if the rescaled F takes on any values greater than 1, which are not allowed since F are IBD probabilities.

Value

If sigma was provided, the $n \times k$ admixture proportion matrix Q. If sigma is missing, a named list is returned containing Q, the rescaled F, and the sigma that together give the desired s and final F_{ST} of the admixed individuals.

Examples

```
## admixture matrix for 1000 individuals drawing alleles from 10 subpops
## and a spread of 2 standard deviations along the 1D geography
Q <- q1d(n=1000, k=10, sigma=2)

## a similar model but with a bias coefficient "s" of exactly 1/2
k <- 10
F <- 1:k # Fst vector for intermediate subpops, up to a factor (will be rescaled below)
Fst <- 0.1 # desired final Fst of admixed individuals
obj <- q1d(n=1000, k=k, s=0.5, F=F, Fst=Fst)
## in this case return value is a named list with three items:
Q <- obj$Q # admixture proportions
F <- obj$F # rescaled Fst vector for intermediate subpops
sigma <- obj$sigma # and the sigma that gives the desired s and final Fst
```

---

q1dc *Construct admixture proportion matrix for circular 1D geography*

Description

Assumes k intermediate subpopulations placed along a circumference (the $[0, 2\pi]$ line that wraps around) with even spacing spread by random walks, then n individuals sampled equally spaced in $[a, b]$ (default $[0, 2\pi]$) draw their admixture proportions relative to the Von Mises density that models the random walks of each of these intermediate subpopulations. The spread of the random walks (the $\sigma = 1/\sqrt{\kappa}$ of the Von Mises densities) is set to sigma if not missing, otherwise σ is found numerically to give the desired bias coefficient s, the vector F of F_{ST}'s for the intermediate subpopulations up to a scalar factor, and the final F_{ST} of the admixed individuals (see details below).

Usage

```
q1dc(n, k, sigma, a = 0, b = 2 * pi, s, F, Fst, interval = c(0.1, 10),
  tol = .Machine$double.eps)
```

Arguments

n	Number of individuals
k	Number of intermediate subpopulations
sigma	Spread of intermediate subpopulations (approximate standard deviation of Von Mises densities, see above)
a	Location of first individual
b	Location of last individual
	OPTIONS FOR BIAS COEFFICIENT VERSION
s	The desired bias coefficient, which specifies σ indirectly. Required if sigma is missing
F	The length-k vector of inbreeding coefficients (or F_{ST}'s) of the intermediate subpopulations, up to a scaling factor (which cancels out in calculations). Required if sigma is missing
Fst	The desired final F_{ST} of the admixed individuals. Required if sigma is missing
interval	Restrict the search space of σ to this interval
tol	The numerical tolerance used to declare the solution found

Details

When sigma is missing, the function determines its value using the desired s, F up to a scalar factor, and Fst. Uniform weights for the final generalized F_{ST} are assumed. The scaling factor of the input F is irrelevant because it cancels out in s; after sigma is found, F is rescaled to give the desired final F_{ST}. However, the function stops with a fatal error if the rescaled F takes on any values greater than 1, which are not allowed since F are IBD probabilities.

Value

If sigma was provided, the $n \times k$ admixture proportion matrix Q. If sigma is missing, a named list is returned containing Q, the rescaled F, and the sigma that together give the desired s and final F_{ST} of the admixed individuals.

Examples

```
## admixture matrix for 1000 individuals drawing alleles from 10 subpops
## and a spread of about 2 standard deviations along the circular 1D geography
Q <- q1dc(n=1000, k=10, sigma=2)

## a similar model but with a bias coefficient "s" of exactly 1/2
k <- 10
F <- 1:k # Fst vector for intermediate subpops, up to a factor (will be rescaled below)
Fst <- 0.1 # desired final Fst of admixed individuals
obj <- q1dc(n=1000, k=k, s=0.5, F=F, Fst=Fst)
```

```
## in this case return value is a named list with three items:
Q <- obj$Q # admixture proportions
F <- obj$F # rescaled Fst vector for intermediate subpops
sigma <- obj$sigma # and the sigma that gives the desired s and final Fst
```

`qis` *Construct admixture proportion matrix for independent subpopulations*

Description

This function constructs an admixture proportion matrix where every individuals is actually unadmixed (draws its full ancestry from a single intermediate subpopulation). The inputs are the vector of subpopulation labels `labs` for every individual (length n), and the length-k vector of unique subpopulations `subpops` in the desired order. If `subpops` is missing, the sorted unique subpopulations observed in `labs` is used. This function returns the admixture proportion matrix Q with individuals along the rows and subpopulations along the columns, marking for each individual TRUE for the column corresponding to its subpopulation, FALSE otherwise. Treating the entries of Q as numerical, every individual has an admixture proportion of 1 for its subpopulation and 0 for all other subpopulations.

Usage

```
qis(labs, subpops)
```

Arguments

`labs`	Length-n vector of subpopulation labels
`subpops`	Optional length-k vector of unique subpopulations in desired order. Stops if `subpops` does not contain all unique labels in `labs` (no error if `subpops` contains additional labels).

Value

The $n \times k$ admixture proportion matrix Q. The unique subpopulation labels are given in `colnames(Q)`.

Examples

```
# vector of subpopulation memberships
labs <- c(1,1,1,2,2,3,1)
# admixture matrix with subpopulations (along columns) sorted
Q <- qis(labs)

# declare subpopulations in custom order
subpops <- c(3,1,2)
# columns will be reordered to match subpops as provided
Q <- qis(labs, subpops)
```

```
# declare subpopulations with unobserved labels
subpops <- 1:5
# note columns 4 and 5 will be false for all individuals
Q <- qis(labs, subpops)
```

---

rbnpsd *Simulate random allele frequencies and genotypes from the BN-PSD admixture model*

Description

This function returns simulated ancestral, intermediate, and individual-specific allele frequencies and genotypes given the admixture structure, as determined by the admixture proportions and the vector of intermediate subpopulation F_{ST} values. The function is a wrapper around rpanc, rpint, rpiaf, and rgeno. Below m is the number of loci, n is the number of individuals, and k is the number of intermediate subpopulations.

Usage

```
rbnpsd(Q, F, m, wantX = TRUE, wantP = TRUE, wantB = TRUE, wantPa = TRUE,
  lowMem = FALSE, verbose = FALSE)
```

Arguments

Q	The $n \times k$ matrix of admixture proportions
F	The length-k vector of intermediate subpopulation F_{ST} values
m	The number of loci to draw
wantX	If TRUE (default), calculates and includes the random genotype matrix in the return list
wantP	If TRUE (default), includes the random IAF matrix in the return list
wantB	If TRUE (default), includes the random intermediate pop allele freq matrix in the return list
wantPa	If TRUE (default), includes the random ancestral allele freq matrix in the return list
lowMem	If TRUE, uses a low-memory algorithm to raw genotypes without storing or returning the corresponding IAF matrix.
verbose	If TRUE, prints messages for every stage in the algorithm

Value

A named list that includes the following random matrices: X=genotypes, P=IAFs, B=intermediate pop allele freqs, Pa=vector of ancestral allele frequencies. Items may be omitted depending on the values of wantX, wantP, wantB, or wantPa above.

Examples

```
# dimensions
m <- 10 # number of loci
n <- 5 # number of individuals
k <- 2 # number of intermediate subpops

# define population structure
F <- c(0.1, 0.3) # FST values for k=2 subpopulations
sigma <- 1 # dispersion parameter of intermediate subpops
Q <- q1d(n, k, sigma) # admixture proportions from 1D geography

# draw all random allele freqs and genotypes
out <- rbnpsd(Q, F, m)
X <- out$X # genotypes
P <- out$P # IAFs
B <- out$B # Intermediate AFs
pAnc <- out$Pa # Ancestral AFs
```

---

rgeno *Draw genotypes from the admixture model*

Description

Given the Individual-specific Allele Frequency (IAF) π_{ij} for locus i and individual j, genotypes are drawn binomially:

$$x_{ij}|\pi_{ij} \sim \text{Binomial}(2, \pi_{ij}).$$

Below m is the number of loci, n the number of individuals, and k the number of intermediate subpopulations. If an admixture proportion matrix Q is provided as the second argument, the first argument P is treated as the intermediate subpopulation allele frequency matrix and the IAF matrix is given by

$$PQ^T.$$

If Q is missing, then P is treated as the IAF matrix.

Usage

```
rgeno(P, Q = NULL, lowMem = FALSE)
```

Arguments

P	The $m \times n$ IAF matrix (if Q is missing) or the $m \times k$ intermediate subpopulation allele frequency matrix (if Q is present)
Q	The optional $n \times k$ admixture proportion matrix
lowMem	If TRUE, the low-memory algorithm is used (Q must be present)

Details

To reduce memory, set `lowMem=TRUE` to draw genotypes one locus at the time from P and Q (both must be present). This low-memory algorithm prevents the construction of the entire IAF matrix, but is considerably slower than the standard algorithm.

Value

The $m \times n$ genotype matrix

Examples

```
# dimensions
m <- 10 # number of loci
n <- 5 # number of individuals
k <- 2 # number of intermediate subpops

# define population structure
F <- c(0.1, 0.3) # FST values for k=2 subpops
sigma <- 1 # dispersion parameter of intermediate subpops
Q <- q1d(n, k, sigma) # non-trivial admixture proportions

# draw alelle frequencies
pAnc <- rpanc(m) # random vector of ancestral allele frequencies
B <- rpint(pAnc, F) # matrix of intermediate subpop allele freqs
P <- rpiaf(B,Q) # matrix of individual-specific allele frequencies

# draw genotypes from intermediate subpops (one individual each)
Xb <- rgeno(B)
# and genotypes for admixed individuals
Xp <- rgeno(P)
```

---

rpanc *Draw uniform ancestral allele frequencies*

Description

This is simply a wrapper around `runif` with different defaults and additional validations.

Usage

```
rpanc(m, min = 0.01, max = 0.5)
```

Arguments

`m`	Number of loci
`min`	Minimum allele frequency to draw
`max`	Maximum allele frequency to draw

Value

A length-m vector of ancestral allele frequencies

Examples

```
pAnc <- rpanc(m=10)
```

`rpiaf` *Construct individual-specific allele frequency matrix*

Description

Here m is the number of loci, n the number of individuals, and k the number of intermediate subpopulations. The $m \times n$ Individual-specific Allele Frequency (IAF) matrix P is constructed from the $m \times k$ intermediate subpopulation allele frequency matrix B and the $n \times k$ admixture proportion matrix Q using

$$P = BQ^T.$$

Usage

```
rpiaf(B, Q)
```

Arguments

`B`	The $m \times k$ intermediate subpopulation allele frequency matrix
`Q`	The $n \times k$ admixture proportion matrix

Value

The $m \times n$ IAF matrix P

Examples

```
m <- 10 # number of loci
n <- 5 # number of individuals
k <- 2 # number of intermediate subpops
pAnc <- rpanc(m) # random vector of ancestral allele frequencies
F <- c(0.1, 0.3) # FST values for k=2 subpops
B <- rpint(pAnc, F) # matrix of intermediate subpop allele freqs
sigma <- 1 # dispersion parameter of intermediate subpops
Q <- q1d(n, k, sigma) # non-trivial admixture proportions
P <- rpiaf(B,Q)
```

rpint *Draw intermediate subpopulation allele frequencies*

Description

Intermediate subpopulation allele frequencies $p_i^{S_u}$ for subpopulation S_u at locus i are drawn from the Balding-Nichols distribution with ancestral allele frequency p_i^T and F_{ST} parameter $f_{S_u}^T$ as

$$p_i^{S_u} \sim \text{Beta}(\nu_u p_i^T, \nu_u(1 - p_i^T)),$$

where $\nu_u = 1/f_{S_u}^T - 1$. Below m is the number of loci and k is the number of subpopulations.

Usage

```
rpint(pAnc, F)
```

Arguments

pAnc	The length-m vector of ancestral allele frequencies per locus
F	The length-k vector of subpopulation F_{ST} values

Value

The $m \times k$ matrix of intermediate subpopulation allele frequencies

Examples

```
m <- 10 # number of loci
pAnc <- rpanc(m) # random vector of ancestral allele frequencies
F <- c(0.1, 0.3) # FST values for two subpops
B <- rpint(pAnc, F) # matrix of intermediate subpop allele freqs
```

Index