Package: SNPknock (via r-universe)

November 1, 2024

Type Package

Title Knockoffs for Hidden Markov Models and Genetic Data

Version 0.8.4

Date 2019-11-06

Description Generate knockoffs for genetic data and hidden Markov models. For more information, see the website below and the accompanying papers: ``Gene hunting with hidden Markov model knockoffs", Sesia et al., Biometrika, 2019, (<doi:10.1093/biomet/asy033>). ``Multi-resolution localization of causal variants across the genome", Sesia et al., bioRxiv, 2019, (<doi:10.1101/631390>).

License GPL-3

Language en-US

Depends R (>= 3.3.0)

Imports Rcpp (>= 0.12.13), Rdpack

RdMacros Rdpack

Suggests knitr, testthat, parallel, doParallel

LinkingTo Rcpp, RcppArmadillo, RcppProgress

RoxygenNote 6.1.1

URL https://msesia.github.io/snpknock

BugReports https://github.com/msesia/snpknock/issues

VignetteBuilder knitr

Encoding UTF-8

Repository https://msesia.r-universe.dev

RemoteUrl https://github.com/msesia/snpknock

RemoteRef HEAD

RemoteSha f0d027b4891fa9a0ec2a541c231fb80a8dbbabc8

Contents

`fwGenotypes` *Forward-backward sampling for hidden Markov models for genotypes*

Description

This function samples the latent Markov chain of a hidden Markov model given the observed values, given an HMM for unphased genotypes.

Usage

```
fwGenotypes(X, r, alpha, theta, seed = 123, cluster = NULL,
  display_progress = FALSE)
```

Arguments

`X`	a 0,1,2 matrix of size n-by-p containing the original variables.
`r`	a vector of length p containing the "r" parameters estimated by fastPHASE.
`alpha`	a matrix of size p-by-K containing the "alpha" parameters estimated by fastPHASE.
`theta`	a matrix of size p-by-K containing the "theta" parameters estimated by fastPHASE.
`seed`	an integer random seed (default: 123).
`cluster`	a computing cluster object created by makeCluster (default: NULL).
`display_progress`	whether to show progress bar (default: FALSE).

Value

A 0,1,2 matrix of size n-by-p containing the knockoff variables.

References

Sesia M, Katsevich E, Bates S, Candès E, Sabatti C (2019). "Multi-resolution localization of causal variants across the genome." *bioRxiv*. doi:10.1101/631390.

See Also

Other hmm: `fwHMM`, `fwHaplotypes`

Examples

```
# Problem size
p = 10
n = 100
# Load HMM to generate data
r_file = system.file("extdata", "haplotypes_rhat.txt", package = "SNPknock")
alpha_file = system.file("extdata", "haplotypes_alphahat.txt", package = "SNPknock")
theta_file = system.file("extdata", "haplotypes_thetahat.txt", package = "SNPknock")
char_file = system.file("extdata", "haplotypes_origchars", package = "SNPknock")
hmm.data = loadHMM(r_file, alpha_file, theta_file, char_file, compact=FALSE, phased=FALSE)
hmm.data$Q = hmm.data$Q[1:(p-1),,]
hmm.data$pEmit = hmm.data$pEmit[1:p,,]
# Sample X from this HMM
X = sampleHMM(hmm.data$pInit, hmm.data$Q, hmm.data$pEmit, n=n)
# Load HMM to generate knockoffs
hmm = loadHMM(r_file, alpha_file, theta_file, char_file)
hmm$r = hmm$r[1:p]
hmm$alpha = hmm$alpha[1:p,]
hmm$theta = hmm$theta[1:p,]
# Forward-backward sampling
H = fwGenotypes(X, hmm$r, hmm$alpha, hmm$theta)
```

`fwHaplotypes` *Forward-backward sampling for hidden Markov models for haplotypes*

Description

This function samples the latent Markov chain of a hidden Markov model given the observed values, given an HMM for phased haplotypes.

Usage

```
fwHaplotypes(X, r, alpha, theta, seed = 123, cluster = NULL,
  display_progress = FALSE)
```

Arguments

X	a binary matrix of size n-by-p containing the original variables.
r	a vector of length p containing the "r" parameters estimated by fastPHASE.
alpha	a matrix of size p-by-K containing the "alpha" parameters estimated by fastPHASE.
theta	a matrix of size p-by-K containing the "theta" parameters estimated by fastPHASE.
seed	an integer random seed (default: 123).
cluster	a computing cluster object created by makeCluster (default: NULL).
display_progress	whether to show progress bar (default: FALSE).

Value

A binary matrix of size n-by-p containing the knockoff variables.

References

Sesia M, Katsevich E, Bates S, Candès E, Sabatti C (2019). "Multi-resolution localization of causal variants across the genome." *bioRxiv*. doi:10.1101/631390.

See Also

Other hmm: fwGenotypes, fwHMM

Examples

```
# Problem size
p = 10
n = 100
# Load HMM to generate data
r_file = system.file("extdata", "haplotypes_rhat.txt", package = "SNPknock")
alpha_file = system.file("extdata", "haplotypes_alphahat.txt", package = "SNPknock")
theta_file = system.file("extdata", "haplotypes_thetahat.txt", package = "SNPknock")
char_file = system.file("extdata", "haplotypes_origchars", package = "SNPknock")
hmm.data = loadHMM(r_file, alpha_file, theta_file, char_file, compact=FALSE, phased=TRUE)
hmm.data$Q = hmm.data$Q[1:(p-1),,]
hmm.data$pEmit = hmm.data$pEmit[1:p,,]
# Sample X from this HMM
X = sampleHMM(hmm.data$pInit, hmm.data$Q, hmm.data$pEmit, n=n)
# Load HMM to generate knockoffs
hmm = loadHMM(r_file, alpha_file, theta_file, char_file)
hmm$r = hmm$r[1:p]
hmm$alpha = hmm$alpha[1:p,]
hmm$theta = hmm$theta[1:p,]
# Forward-backward sampling
H = fwHaplotypes(X, hmm$r, hmm$alpha, hmm$theta)
```

`fwHMM` *Forward-backward sampling for hidden Markov models*

Description

This function samples the latent Markov chain of a hidden Markov model given the observed values.

Usage

```
fwHMM(X, pInit, Q, pEmit, seed = 123, cluster = NULL,
  display_progress = FALSE)
```

Arguments

`X`	an integer matrix of size n-by-p containing the original variables.
`pInit`	an array of length K, containing the marginal distribution of the states for the first variable.
`Q`	an array of size (p-1,K,K), containing a list of p-1 transition matrices between the K states of the Markov chain.
`pEmit`	an array of size (p,M,K), containing the emission probabilities for each of the M possible emission states, from each of the K hidden states and the p variables.
`seed`	an integer random seed (default: 123).
`cluster`	a computing cluster object created by makeCluster (default: NULL).
`display_progress`	whether to show progress bar (default: FALSE).

Details

Each element of the matrix X should be an integer value between 0 and M-1. The transition matrices contained in Q are defined with the same convention as in knockoffDMC. The emission propability matrices contained in pEmit are defined such that $P[X_j = k|H_j = l] = \text{pEmit}[j, k, l]$, where H_j is the latent variable associated to X_j.

Value

An integer matrix of size n-by-p containing the knockoff variables.

References

Sesia M, Sabatti C, Candès EJ (2019). "Gene hunting with hidden Markov model knockoffs." *Biometrika*, **106**, 1–18. doi:10.1093/biomet/asy033.

See Also

Other hmm: `fwGenotypes`, `fwHaplotypes`

Examples

```
# Generate data
p=10; K=5; M=3;
pInit = rep(1/K,K)
Q = array(stats::runif((p-1)*K*K),c(p-1,K,K))
for(j in 1:(p-1)) { Q[j,,] = Q[j,,] / rowSums(Q[j,,]) }
pEmit = array(stats::runif(p*M*K),c(p,M,K))
for(j in 1:p) { pEmit[j,,] = pEmit[j,,] / rowSums(pEmit[j,,]) }
X = sampleHMM(pInit, Q, pEmit, n=20)
# Forward-backward sampling
H = fwHMM(X, pInit, Q, pEmit)
```

`knockoffDMC` *Group knockoffs of discrete Markov chains*

Description

This function constructs knockoffs of variables distributed as a discrete Markov chain.

Usage

```
knockoffDMC(X, pInit, Q, groups = NULL, seed = 123, cluster = NULL,
  display_progress = FALSE)
```

Arguments

`X`	an integer matrix of size n-by-p containing the original variables.
`pInit`	an array of length K, containing the marginal distribution of the states for the first variable.
`Q`	an array of size (p-1,K,K), containing a list of p-1 transition matrices between the K states of the Markov chain.
`groups`	an array of length p, describing the group membership of each variable (default: NULL).
`seed`	an integer random seed (default: 123).
`cluster`	a computing cluster object created by makeCluster (default: NULL).
`display_progress`	whether to show progress bar (default: FALSE).

Details

Each element of the matrix X should be an integer value between 0 and K-1. The transition matrices contained in Q are defined such that $P[X_{j+1} = k | X_j = l] = Q[j, l, k]$.

Value

An integer matrix of size n-by-p containing the knockoff variables.

References

Sesia M, Sabatti C, Candès EJ (2019). "Gene hunting with hidden Markov model knockoffs." *Biometrika*, **106**, 1–18. doi:10.1093/biomet/asy033. Sesia M, Katsevich E, Bates S, Candès E, Sabatti C (2019). "Multi-resolution localization of causal variants across the genome." *bioRxiv*. doi:10.1101/631390.

See Also

Other knockoffs: `knockoffGenotypes`, `knockoffHMM`, `knockoffHaplotypes`

Examples

```
# Generate data
p = 10; K = 5;
pInit = rep(1/K,K)
Q = array(stats::runif((p-1)*K*K),c(p-1,K,K))
for(j in 1:(p-1)) { Q[j,,] = Q[j,,] / rowSums(Q[j,,]) }
X = sampleDMC(pInit, Q, n=20)
# Generate knockoffs
Xk = knockoffDMC(X, pInit, Q)
# Generate group-knockoffs for groups of size 3
groups = rep(seq(p), each=3, length.out=p)
Xk = knockoffDMC(X, pInit, Q, groups=groups)
```

`knockoffGenotypes` *Group-knockoffs of unphased genotypes*

Description

This function efficiently constructs group-knockoffs of 0,1,2 variables distributed according to the fastPHASE model for unphased genotypes.

Usage

```
knockoffGenotypes(X, r, alpha, theta, groups = NULL, seed = 123,
  cluster = NULL, display_progress = FALSE)
```

Arguments

`X`	a 0,1,2 matrix of size n-by-p containing the original variables.
`r`	a vector of length p containing the "r" parameters estimated by fastPHASE.
`alpha`	a matrix of size p-by-K containing the "alpha" parameters estimated by fastPHASE.
`theta`	a matrix of size p-by-K containing the "theta" parameters estimated by fastPHASE.

groups	a vector of length p containing group memberships for each variable. Indices are assumed to be monotone increasing, starting from 1 (default: NULL).
seed	an integer random seed (default: 123).
cluster	a computing cluster object created by makeCluster (default: NULL).
display_progress	whether to show progress bar (default: FALSE).

Details

Generate group-knockoffs of unphased genotypes according to the fastPHASE HMM. The required model parameters can be obtained through fastPHASE and loaded with loadHMM. This function is more efficient than knockoffHMM for haplotype data.

Value

A 0,1,2 matrix of size n-by-p containing the knockoff variables.

References

Sesia M, Katsevich E, Bates S, Candès E, Sabatti C (2019). "Multi-resolution localization of causal variants across the genome." *bioRxiv*. doi:10.1101/631390.

See Also

Other knockoffs: knockoffDMC, knockoffHMM, knockoffHaplotypes

Examples

```
# Problem size
p = 10
n = 100
# Load HMM to generate data
r_file = system.file("extdata", "haplotypes_rhat.txt", package = "SNPknock")
alpha_file = system.file("extdata", "haplotypes_alphahat.txt", package = "SNPknock")
theta_file = system.file("extdata", "haplotypes_thetahat.txt", package = "SNPknock")
char_file = system.file("extdata", "haplotypes_origchars", package = "SNPknock")
hmm.data = loadHMM(r_file, alpha_file, theta_file, char_file, compact=FALSE, phased=FALSE)
hmm.data$Q = hmm.data$Q[1:(p-1),,]
hmm.data$pEmit = hmm.data$pEmit[1:p,,]
# Sample X from this HMM
X = sampleHMM(hmm.data$pInit, hmm.data$Q, hmm.data$pEmit, n=n)
# Load HMM to generate knockoffs
hmm = loadHMM(r_file, alpha_file, theta_file, char_file)
hmm$r = hmm$r[1:p]
hmm$alpha = hmm$alpha[1:p,]
hmm$theta = hmm$theta[1:p,]
# Generate knockoffs
Xk = knockoffGenotypes(X, hmm$r, hmm$alpha, hmm$theta)
# Generate group-knockoffs for groups of size 3
groups = rep(seq(p), each=3, length.out=p)
Xk = knockoffGenotypes(X, hmm$r, hmm$alpha, hmm$theta, groups=groups)
```

knockoffHaplotypes *Group-knockoffs of phased haplotypes*

Description

This function efficiently constructs group-knockoffs of binary variables distributed according to the fastPHASE model for phased haplotypes.

Usage

```
knockoffHaplotypes(X, r, alpha, theta, groups = NULL, seed = 123,
  cluster = NULL, display_progress = FALSE)
```

Arguments

X	a binary matrix of size n-by-p containing the original variables.
r	a vector of length p containing the "r" parameters estimated by fastPHASE.
alpha	a matrix of size p-by-K containing the "alpha" parameters estimated by fastPHASE.
theta	a matrix of size p-by-K containing the "theta" parameters estimated by fastPHASE.
groups	a vector of length p containing group memberships for each variable. Indices are assumed to be monotone increasing, starting from 1 (default: NULL).
seed	an integer random seed (default: 123).
cluster	a computing cluster object created by makeCluster (default: NULL).
display_progress	whether to show progress bar (default: FALSE).

Details

Generate group-knockoffs of phased haplotypes according to the fastPHASE HMM. The required model parameters can be obtained through fastPHASE and loaded with loadHMM. This function is more efficient than knockoffHMM for haplotype data.

Value

A binary matrix of size n-by-p containing the knockoff variables.

References

Sesia M, Katsevich E, Bates S, Candès E, Sabatti C (2019). "Multi-resolution localization of causal variants across the genome." *bioRxiv*. doi:10.1101/631390.

See Also

Other knockoffs: knockoffDMC, knockoffGenotypes, knockoffHMM

Examples

```
# Problem size
p = 10
n = 100
# Load HMM to generate data
r_file = system.file("extdata", "haplotypes_rhat.txt", package = "SNPknock")
alpha_file = system.file("extdata", "haplotypes_alphahat.txt", package = "SNPknock")
theta_file = system.file("extdata", "haplotypes_thetahat.txt", package = "SNPknock")
char_file = system.file("extdata", "haplotypes_origchars", package = "SNPknock")
hmm.data = loadHMM(r_file, alpha_file, theta_file, char_file, compact=FALSE, phased=TRUE)
hmm.data$Q = hmm.data$Q[1:(p-1),,]
hmm.data$pEmit = hmm.data$pEmit[1:p,,]
# Sample X from this HMM
X = sampleHMM(hmm.data$pInit, hmm.data$Q, hmm.data$pEmit, n=n)
# Load HMM to generate knockoffs
hmm = loadHMM(r_file, alpha_file, theta_file, char_file)
hmm$r = hmm$r[1:p]
hmm$alpha = hmm$alpha[1:p,]
hmm$theta = hmm$theta[1:p,]
# Generate knockoffs
Xk = knockoffHaplotypes(X, hmm$r, hmm$alpha, hmm$theta)
# Generate group-knockoffs for groups of size 3
groups = rep(seq(p), each=3, length.out=p)
Xk = knockoffHaplotypes(X, hmm$r, hmm$alpha, hmm$theta, groups=groups)
```

---

knockoffHMM *Group knockoffs of hidden Markov models*

Description

This function constructs knockoffs of variables distributed as a hidden Markov model.

Usage

```
knockoffHMM(X, pInit, Q, pEmit, groups = NULL, seed = 123,
  cluster = NULL, display_progress = FALSE)
```

Arguments

X	an integer matrix of size n-by-p containing the original variables.
pInit	an array of length K, containing the marginal distribution of the states for the first variable.
Q	an array of size (p-1,K,K), containing a list of p-1 transition matrices between the K states of the Markov chain.
pEmit	an array of size (p,M,K), containing the emission probabilities for each of the M possible emission states, from each of the K hidden states and the p variables.

`groups` an array of length p, describing the group membership of each variable (default: NULL).

`seed` an integer random seed (default: 123).

`cluster` a computing cluster object created by makeCluster (default: NULL).

`display_progress` whether to show progress bar (default: FALSE).

Details

Each element of the matrix X should be an integer value between 0 and M-1. The transition matrices contained in Q are defined with the same convention as in knockoffDMC. The emission propability matrices contained in pEmit are defined such that $P[X_j = k|H_j = l] = \text{pEmit}[j, k, l]$, where H_j is the latent variable associated to X_j.

Value

An integer matrix of size n-by-p containing the knockoff variables.

References

Sesia M, Sabatti C, Candès EJ (2019). "Gene hunting with hidden Markov model knockoffs." *Biometrika*, **106**, 1–18. doi:10.1093/biomet/asy033. Sesia M, Katsevich E, Bates S, Candès E, Sabatti C (2019). "Multi-resolution localization of causal variants across the genome." *bioRxiv*. doi:10.1101/631390.

See Also

Other knockoffs: `knockoffDMC`, `knockoffGenotypes`, `knockoffHaplotypes`

Examples

```
# Generate data
p=10; K=5; M=3;
pInit = rep(1/K,K)
Q = array(stats::runif((p-1)*K*K),c(p-1,K,K))
for(j in 1:(p-1)) { Q[j,,] = Q[j,,] / rowSums(Q[j,,]) }
pEmit = array(stats::runif(p*M*K),c(p,M,K))
for(j in 1:p) { pEmit[j,,] = pEmit[j,,] / rowSums(pEmit[j,,]) }
X = sampleHMM(pInit, Q, pEmit, n=20)
# Generate knockoffs
Xk = knockoffHMM(X, pInit, Q, pEmit)
# Generate group-knockoffs for groups of size 3
groups = rep(seq(p), each=3, length.out=p)
Xk = knockoffHMM(X, pInit, Q, pEmit, groups=groups)
```

`loadHMM` *Load HMM parameters fitted by fastPHASE*

Description

This function loads the parameter estimates obtained by fastPHASE (see runFastPhase) and assembles the Li and Stephens HMM, in the format required by the knockoff generation functions knockoffHaplotypes and knockoffGenotypes.

Usage

```
loadHMM(r_file, alpha_file, theta_file, char_file, compact = TRUE,
  phased = FALSE)
```

Arguments

`r_file`	a string with the path of the "_rhat.txt" file produced by fastPHASE.
`alpha_file`	a string with the path of the "_alphahat.txt" file produced by fastPHASE.
`theta_file`	a string with the path of the "_thetahat.txt" file produced by fastPHASE.
`char_file`	a string with the path of the "_origchars" file produced by fastPHASE.
`compact`	whether to assemble the explicit transition and emission matrices for the HMM (default: FALSE).
`phased`	whether to assemble a model for phased haplotypes, if compact==FALSE (default: FALSE).

Details

This function by default returns a structure with three fields:

- "r": a numerical array of length p.
- "alpha": a numerical array of size (p,K).
- "theta": a numerical array of size (p,K).

If the parameter compact is FALSE, this function assembles the HMM model for the genotype data (either unphased or phased), in the format required by the knockoff generation function knockoffHMM.

Value

A structure containing the parameters from the Li and Stephens HMM for phased haplotypes.

References

Scheet P, Stephens M (2006). "A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase." *Am. J. Hum. Genet.*, **78**, 629–644. doi:10.1086/502802.

See Also

Other fastPHASE: runFastPhase, writeXtoInp

Examples

```
# Specify the location of the fastPHASE output files containing the parameter estimates.
# Example files can be found in the package installation directory.
r_file = system.file("extdata", "genotypes_rhat.txt", package = "SNPknock")
alpha_file = system.file("extdata", "genotypes_alphahat.txt", package = "SNPknock")
theta_file = system.file("extdata", "genotypes_thetahat.txt", package = "SNPknock")
char_file = system.file("extdata", "genotypes_origchars", package = "SNPknock")

# Read the parameter files and load the HMM
hmm = loadHMM(r_file, alpha_file, theta_file, char_file)

# Read the parameter files and load the HMM
hmm.large = loadHMM(r_file, alpha_file, theta_file, char_file, compact=FALSE)
```

---

runFastPhase *Fit an HMM to genetic data using fastPHASE*

Description

This function provides a wrapper for the fastPHASE executable in order to fit an HMM to either unphased genotype data or phased haplotype data. The software fastPHASE will fit the HMM to the genotype data and write the corresponding parameter estimates in four separate files. Since fastPHASE is not an R package, this executable must be downloaded separately by the user. Visit http://scheet.org/software.html for more information on how to obtain fastPHASE.

Usage

```
runFastPhase(fp_path, X_file, out_path = NULL, K = 12, numit = 25,
  phased = FALSE, seed = 1)
```

Arguments

fp_path	a string with the path to the directory with the fastPHASE executable.
X_file	a string with the path of the genotype input file containing X in fastPHASE format (as created by writeXtoInp).
out_path	a string with the path of the directory in which the parameter estimates will be saved (default: NULL). If this is equal to NULL, a temporary file in the R temporary directory will be used.
K	the number of hidden states for each haplotype sequence (default: 12).
numit	the number of EM iterations (default: 25).
phased	whether the data are already phased (default: FALSE).
seed	the random seed for the EM algorithm (default: 1).

Details

The software fastPHASE saves the parameter estimates in four separate files whose names begin with the string contained in 'out_path' and end with:

- "_rhat.txt"
- "_alphahat.txt"
- "_thetahat.txt"
- "_origchars"

The HMM for the genotype data can then be loaded from these files by calling loadHMM.

Value

A string containing the path of the directory in which the parameter estimates were saved. This is useful to find the data when the default option for 'out_path' is used and the output is written in an R temporary directory.

References

Scheet P, Stephens M (2006). "A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase." *Am. J. Hum. Genet.*, **78**, 629–644. doi:10.1086/502802.

See Also

Other fastPHASE: `loadHMM`, `writeXtoInp`

Examples

```
fp_path  = "~/bin/fastPHASE" # Path to the fastPHASE executable

# Run fastPHASE on unphased genotypes
# Specify the path to the genotype input file in ".inp" format.
# An example file containing unphased genotypes can be found in the package installation folder.
X_file = system.file("extdata", "genotypes.inp", package = "SNPknock")
fp_outPath = runFastPhase(fp_path, X_file)

# Run fastPHASE on phased haplotypes
# An example file containing phased haplotypes can be found in the package installation folder.
H_file = system.file("extdata", "haplotypes.inp", package = "SNPknock")
fp_outPath = runFastPhase(fp_path, H_file, phased=TRUE)
```

sampleDMC *Sample discrete Markov chains*

Description

This function draws independent random samples of a discrete Markov chain.

Usage

```
sampleDMC(pInit, Q, n = 1)
```

Arguments

pInit	an array of length K, containing the marginal distribution of the states for the first variable.
Q	an array of size (p-1,K,K), containing a list of p-1 transition matrices between the K states of the Markov chain.
n	the number of independent samples to be drawn (default: 1).

Details

Each element of the output matrix is an integer value between 0 and K-1. The transition matrices contained in Q are defined such that $P[X_{j+1} = k|X_j = l] = Q[j, l, k]$.

Value

A matrix of size n-by-p containing the n observed Markov chains of length p.

References

Sesia M, Sabatti C, Candès EJ (2019). "Gene hunting with hidden Markov model knockoffs." *Biometrika*, **106**, 1–18. doi:10.1093/biomet/asy033.

See Also

Other models: sampleHMM

Examples

```
p=10; K=5;
pInit = rep(1/K,K)
Q = array(stats::runif((p-1)*K*K),c(p-1,K,K))
for(j in 1:(p-1)) { Q[j,,] = Q[j,,] / rowSums(Q[j,,]) }
X = sampleDMC(pInit, Q, n=20)
```

`sampleHMM` *Sample hidden Markov models*

Description

This function draws independent random samples of an hidden Markov model.

Usage

```
sampleHMM(pInit, Q, pEmit, n = 1)
```

Arguments

`pInit`	an array of length K, containing the marginal distribution of the states for the first variable.
`Q`	an array of size (p-1,K,K), containing a list of p-1 transition matrices between the K states of the Markov chain.
`pEmit`	an array of size (p,M,K), containing the emission probabilities for each of the M possible emission states, from each of the K hidden states and the p variables.
`n`	the number of independent samples to be drawn (default: 1).

Details

Each element of the output matrix is an integer value between 0 and K-1. The transition matrices contained in Q are defined with the same convention as in sampleDMC. The emission propability matrices contained in pEmit are defined such that $P[X_j = k|H_j = l] = \mathrm{pEmit}[j, k, l]$, where H_j is the latent variable associated to X_j.

Value

A matrix of size n-by-p containing the n observed Markov chains of length p.

References

Sesia M, Sabatti C, Candès EJ (2019). "Gene hunting with hidden Markov model knockoffs." *Biometrika*, **106**, 1–18. doi:10.1093/biomet/asy033.

See Also

Other models: `sampleDMC`

Examples

```
p=10; K=5; M=3;
pInit = rep(1/K,K)
Q = array(stats::runif((p-1)*K*K),c(p-1,K,K))
for(j in 1:(p-1)) { Q[j,,] = Q[j,,] / rowSums(Q[j,,]) }
pEmit = array(stats::runif(p*M*K),c(p,M,K))
for(j in 1:p) { pEmit[j,,] = pEmit[j,,] / rowSums(pEmit[j,,]) }
X = sampleHMM(pInit, Q, pEmit, n=20)
```

`writeXtoInp` *Convert genotypes X into the fastPHASE input format*

Description

This function converts a genetic matrix X into the fastPHASE input format and saves it to a user-specified file. Then, an HMM can be fitted by calling fastPHASE with runFastPhase.

Usage

```
writeXtoInp(X, phased = FALSE, out_file = NULL)
```

Arguments

`X`	either a matrix of size n-by-p containing unphased genotypes for n individuals, or a matrix of size 2n-by-p containing phased haplotypes for n individuals.
`phased`	whether the data are phased (default: FALSE). If this is equal to TRUE, each pair of consecutive rows will be assumed to correspond to phased haplotypes from the same individual.
`out_file`	a string containing the path of the output file onto which X will be written (default: NULL). If this is equal to NULL, a temporary file in the R temporary directory will be used.

Value

A string containing the path of the output file onto which X was written. This is useful to find the data when the default option for 'out_file' is used and X is written onto a temporary file in the R temporary directory.

References

Scheet P, Stephens M (2006). "A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase." *Am. J. Hum. Genet.*, **78**, 629–644. doi:10.1086/502802.

See Also

Other fastPHASE: `loadHMM`, `runFastPhase`

Examples

```
# Convert unphased genotypes
# Load an example data matrix X from the package installation directory.
X_file = system.file("extdata", "genotypes.RData", package = "SNPknock")
load(X_file)
# Write X in a temporary file
Xinp_file = writeXtoInp(X)

# Convert phased haplotypes
# Load an example data matrix H from the package installation directory.
H_file = system.file("extdata", "haplotypes.RData", package = "SNPknock")
load(H_file)
# Write H in a temporary file
Hinp_file = writeXtoInp(H, phased=TRUE)
```

Index