Package 'SNPknock'

May 30, 2018

Type Package

Title Knockoffs for Hidden Markov Models and Genetic Data

Version 0.7.1

Date 2018-05-30

Author Matteo Sesia

Maintainer Matteo Sesia <msesia@stanford.edu>

Description Generates knockoff variables from discrete Markov chains and hidden Markov models, with specific support for genetic data.
For more information, see the website below and the accompanying paper: Sesia et al.,
``Gene Hunting with Knockoffs for Hidden Markov Models",
2017, <arXiv:1706.04677>.

URL https://web.stanford.edu/group/candes/knockoffs/software/snpknock/

License GPL (>= 2)

Depends R (>= 3.3.0)

Imports Rcpp (>= 0.12.13)

Suggests knitr, testthat, parallel, doParallel

LinkingTo Rcpp, RcppProgress

RoxygenNote 6.0.1

VignetteBuilder knitr

NeedsCompilation yes

Repository CRAN

Date/Publication 2018-05-30 17:51:41 UTC

R topics documented:

`SNPknock.fp.loadFit` *Load the parameter estimates obtained by fastPHASE*

Description

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

Usage

```
SNPknock.fp.loadFit(r_file, alpha_file, theta_file, char_file)
```

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.

Details

This function 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).

Value

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

References

Scheet and Stephens, A fast and flexible statistical model for large-scale population genotype data, Am J Hum Genet (2006). http://www.sciencedirect.com/science/article/pii/S000292970763701X

See Also

Other fastPHASE: `SNPknock.fp.loadFit_hmm`, `SNPknock.fp.runFastPhase`, `SNPknock.fp.writeX`

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 build the HMM
hmm = SNPknock.fp.loadFit(r_file, alpha_file, theta_file, char_file)
```

`SNPknock.fp.loadFit_hmm`

Load the parameter estimates obtained by fastPHASE and assembles the HMM model for genotype data

---

Description

This function loads the parameter estimates obtained by fastPHASE (see SNPknock.fp.runFastPhase) and assembles the HMM model for the genotype data (either unphased or phased), in the format required by the knockoff generation function SNPknock.knockoffHMM.

Usage

```
SNPknock.fp.loadFit_hmm(r_file, alpha_file, theta_file, char_file,
  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.
`phased`	whether to assemble a model for phased haplotypes (default: FALSE).

Details

This function returns a structure with three fields:

- "pInit": a numerical array of length K, containing the marginal distribution of the hidden states for the first SNP.
- "Q": a numerical array of size (p-1,K,K), containing a list of p-1 transition matrices between the K latent states of the HMM.
- "pEmit": a numerical array of size (p,K,3), containing the emission probabilities of the hidden states for each of the p SNPs.

Value

A structure describing the HMM fitted by fastPHASE.

References

Scheet and Stephens, A fast and flexible statistical model for large-scale population genotype data, Am J Hum Genet (2006). http://www.sciencedirect.com/science/article/pii/S000292970763701X

See Also

Other fastPHASE: SNPknock.fp.loadFit, SNPknock.fp.runFastPhase, SNPknock.fp.writeX

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 build the HMM for unphased genotypes
hmm = SNPknock.fp.loadFit_hmm(r_file, alpha_file, theta_file, char_file)

# Read the parameter files and build the HMM for phased haplotypes
hmm = SNPknock.fp.loadFit_hmm(r_file, alpha_file, theta_file, char_file, phased=TRUE)
```

SNPknock.fp.runFastPhase

Calls fastPHASE to fit an HMM to genotype data

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

```
SNPknock.fp.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 SNPknock.fp.writeX).
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 SNPknock.fp.loadFit.

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 and Stephens, A fast and flexible statistical model for large-scale population genotype data, Am J Hum Genet (2006). http://www.sciencedirect.com/science/article/pii/S000292970763701X

See Also

Other fastPHASE: SNPknock.fp.loadFit_hmm, SNPknock.fp.loadFit, SNPknock.fp.writeX

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 = SNPknock.fp.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 = SNPknock.fp.runFastPhase(fp_path, H_file, phased=TRUE)
```

SNPknock.fp.writeX *Convert a genetic matrix X into the fastPHASE input format*

Description

This function convert 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 SNPknock.fp.runFastPhase.

Usage

```
SNPknock.fp.writeX(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 and Stephens, A fast and flexible statistical model for large-scale population genotype data, Am J Hum Genet (2006). http://www.sciencedirect.com/science/article/pii/S000292970763701X

See Also

Other fastPHASE: SNPknock.fp.loadFit_hmm, SNPknock.fp.loadFit, SNPknock.fp.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 = SNPknock.fp.writeX(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 = SNPknock.fp.writeX(H, phased=TRUE)
```

SNPknock.knockoffDMC *Knockoff copies of a discrete Markov chain*

Description

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

Usage

```
SNPknock.knockoffDMC(X, pInit, Q, seed = 123, cluster = NULL,
  display_progress = TRUE)
```

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.
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: TRUE).

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 et al., Gene Hunting with Knockoffs for Hidden Markov Models, arXiv:1706.04677 (2017).
https://statweb.stanford.edu/~candes/papers/HMM_Knockoffs.pdf

See Also

Other knockoffs: SNPknock.knockoffGenotypes, SNPknock.knockoffHMM, SNPknock.knockoffHaplotypes

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 = SNPknock.models.sampleDMC(pInit, Q, n=20)
Xk = SNPknock.knockoffDMC(X, pInit, Q)
```

SNPknock.knockoffGenotypes

Knockoff copies of unphased genotypes

Description

This function efficiently constructs knockoff copies of 0,1,2 variables distributed according to the Li and Stephens model for unphased genotypes.

Usage

```
SNPknock.knockoffGenotypes(X, r, alpha, theta, seed = 123, cluster = NULL,
  display_progress = TRUE)
```

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: TRUE).

Details

Generate knockoff copies of unphased genotypes according to the Li and Stephens HMM. The required model parameters can be obtained through fastPHASE and loaded with SNPknock.fp.loadFit. This function is more efficient than SNPknock.knockoffHMM for genotype data.

Value

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

References

Sesia et al., Gene Hunting with Knockoffs for Hidden Markov Models, arXiv:1706.04677 (2017). https://statweb.stanford.edu/~candes/papers/HMM_Knockoffs.pdf

Scheet and Stephens, A fast and flexible statistical model for large-scale population genotype data, Am J Hum Genet (2006). http://www.sciencedirect.com/science/article/pii/S000292970763701X

See Also

Other knockoffs: SNPknock.knockoffDMC, SNPknock.knockoffHMM, SNPknock.knockoffHaplotypes

Examples

```
# Load an example dataset of unphased genotypes from the package installation directory.
X_file = system.file("extdata", "genotypes.RData", package = "SNPknock")
load(X_file)

# 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")
theta_file = system.file("extdata", "genotypes_thetahat.txt", package = "SNPknock")
alpha_file = system.file("extdata", "genotypes_alphahat.txt", package = "SNPknock")
char_file = system.file("extdata", "genotypes_origchars", package = "SNPknock")

# Read the parameter files and build the HMM
hmm = SNPknock.fp.loadFit(r_file, theta_file, alpha_file, char_file)

# Generate the knockoffs
Xk = SNPknock.knockoffGenotypes(X, hmm$r, hmm$alpha, hmm$theta)
```

SNPknock.knockoffHaplotypes

Knockoff copies of phased haplotypes

Description

This function efficiently constructs knockoff copies of binary variables distributed according to the Li and Stephens model for phased haplotypes.

Usage

```
SNPknock.knockoffHaplotypes(X, r, alpha, theta, seed = 123, cluster = NULL,
  display_progress = TRUE)
```

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: TRUE).

Details

Generate knockoff copies of phased haplotypes according to the Li and Stephens HMM. The required model parameters can be obtained through fastPHASE and loaded with SNPknock.fp.loadFit. This function is more efficient than SNPknock.knockoffHMM for haplotype data.

Value

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

References

Sesia et al., Gene Hunting with Knockoffs for Hidden Markov Models, arXiv:1706.04677 (2017). https://statweb.stanford.edu/~candes/papers/HMM_Knockoffs.pdf

Scheet and Stephens, A fast and flexible statistical model for large-scale population genotype data, Am J Hum Genet (2006). http://www.sciencedirect.com/science/article/pii/S000292970763701X

See Also

Other knockoffs: SNPknock.knockoffDMC, SNPknock.knockoffGenotypes, SNPknock.knockoffHMM

Examples

```
# Load an example dataset of phased haplotypes from the package installation directory.
H_file = system.file("extdata", "haplotypes.RData", package = "SNPknock")
load(H_file)

# 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", "haplotypes_rhat.txt", package = "SNPknock")
theta_file = system.file("extdata", "haplotypes_thetahat.txt", package = "SNPknock")
```

```
alpha_file = system.file("extdata", "haplotypes_alphahat.txt", package = "SNPknock")
char_file = system.file("extdata", "haplotypes_origchars", package = "SNPknock")

# Read the parameter files and build the HMM
hmm = SNPknock.fp.loadFit(r_file, theta_file, alpha_file, char_file)

# Generate the knockoffs
Hk = SNPknock.knockoffHaplotypes(H, hmm$r, hmm$alpha, hmm$theta)
```

SNPknock.knockoffHMM *Knockoff copies of a hidden Markov model*

Description

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

Usage

```
SNPknock.knockoffHMM(X, pInit, Q, pEmit, seed = 123, cluster = NULL,
  display_progress = TRUE)
```

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: TRUE).

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 SNPknock.knockoffDMC. 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

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

References

Sesia et al., Gene Hunting with Knockoffs for Hidden Markov Models, arXiv:1706.04677 (2017). https://statweb.stanford.edu/~candes/papers/HMM_Knockoffs.pdf

See Also

Other knockoffs: SNPknock.knockoffDMC, SNPknock.knockoffGenotypes, SNPknock.knockoffHaplotypes

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 = SNPknock.models.sampleHMM(pInit, Q, pEmit, n=20)
Xk = SNPknock.knockoffHMM(X, pInit, Q, pEmit)
```

SNPknock.models.sampleDMC

Sample discrete Markov chains

Description

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

Usage

```
SNPknock.models.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 et al., Gene Hunting with Knockoffs for Hidden Markov Models, arXiv:1706.04677 (2017). https://statweb.stanford.edu/~candes/papers/HMM_Knockoffs.pdf

See Also

Other models: SNPknock.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 = SNPknock.models.sampleDMC(pInit, Q, n=20)
```

SNPknock.models.sampleHMM

Sample hidden Markov models

Description

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

Usage

```
SNPknock.models.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 SNPknock.models.sampleDMC. 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

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

References

Sesia et al., Gene Hunting with Knockoffs for Hidden Markov Models, arXiv:1706.04677 (2017). https://statweb.stanford.edu/~candes/papers/HMM_Knockoffs.pdf

See Also

Other models: SNPknock.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 = SNPknock.models.sampleHMM(pInit, Q, pEmit, n=20)
```

Index