Package 'BHTSpack'

September 10, 2018

Type Package

Title Bayesian Multi-Plate High-Throughput Screening of Compounds

Version 0.5

Description Can be used for joint identification of candidate compound hits from multiple assays, in drug discovery. This package implements the framework of I. D. Shterev, D. B. Dunson, C. Chan and G. D. Sempowski. ``Bayesian Multi-Plate High-Throughput Screening of Compounds'', Scientific Reports 8(1):9551, 2018. This project was funded by the Division of Allergy, Immunology, and Transplantation, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under contract No. HHSN272201400054C entitled ``Adjuvant Discovery For Vaccines Against West Nile Virus and Influenza'', awarded to Duke University and lead by Drs. Herman Staats and Soman Abraham.

Depends R (>= 3.2.3), R2HTML (>= 2.3.2), xtable (>= 1.8-2)

VignetteBuilder knitr

Suggests knitr

License GPL-3

LazyLoad yes

NeedsCompilation yes

Author Ivo D. Shterev [aut, cre],
David B. Dunson [aut],
Cliburn Chan [aut],
Gregory D. Sempowski [aut]

Maintainer Ivo D. Shterev <i.shterev@duke.edu>

Repository CRAN

Date/Publication 2018-09-10 16:10:02 UTC

R topics documented:

BHTSpack-package *Bayesian Multi-Plate High-Throughput Screening of Compounds*

Description

Can be used for joint identification of candidate hits from multiple assays, in drug discovery. This package implements the framework of I. D. Shterev, D. B. Dunson, C. Chan and G. D. Sempowski. "Bayesian Multi Plate High Throughput Screening of Compounds", arXiv:1709.10041, September 2017. This project was funded by the Division of Allergy, Immunology, and Transplantation, National Institute of Allergy and Infectious Diseases, National Institutes of Health, Department of Health and Human Services, under contract No. HHSN272201400054C entitled "Adjuvant Discovery For Vaccines Against West Nile Virus and Influenza", awarded to Duke University and lead by Drs. Herman Staats and Soman Abraham.

Details

The DESCRIPTION file:

Package:	BHTSpack
Type:	Package
Title:	Bayesian Multi-Plate High-Throughput Screening of Compounds
Version:	0.5
Authors@R:	c(person(c("Ivo", "D."), "Shterev", role = c("aut", "cre"), email = "i.shterev@duke.edu"), person(c("David'
Description:	Can be used for joint identification of candidate compound hits from multiple assays, in drug discovery. Th
Depends:	R (>= 3.2.3), R2HTML (>= 2.3.2), xtable (>= 1.8-2)
VignetteBuilder:	knitr
Suggests:	knitr

License:	GPL-3
LazyLoad:	yes
Author:	Ivo D. Shterev [aut, cre], David B. Dunson [aut], Cliburn Chan [aut], Gregory D. Sempowski [aut]
Maintainer:	Ivo D. Shterev <i.shterev@duke.edu>

Index of help topics:

```
BHTSpack-package        Bayesian Multi-Plate High-Throughput Screening
                        of Compounds
abfun                   package internal function
alpha.u                 package internal function
b.u                     package internal function
bhts                    Bayesian High-Throughput Screening
bhts2HTML               Convert to HTML
data.create             Create Synthetic Data
fdr.r                   package internal function
h.pr.u                  package internal function
hatpai.u                package internal function
ind.u                   package internal function
lambda.u                package internal function
lg.mu.sig               package internal function
mu.k.u                  package internal function
nu.u                    package internal function
pai.u                   package internal function
ptrace                  Trace (ACF) Plots
r.fdr                   Significant Hits
sig.k.u                 package internal function
tau.u                   package internal function
z.pr.u                  package internal function
```

Author(s)

I. D. Shterev, D. B. Dunson, C. Chan and G. D. Sempowski

Maintainer: I. D. Shterev <i.shterev@duke.edu>

References

I. D. Shterev, D. B. Dunson, C. Chan and G. D. Sempowski. "Bayesian Multi-Plate High-Throughput Screening of Compounds". *Scientific Reports*, 8(1):9551, 2018.

`abfun` *package internal function*

Description

package internal function

Usage

```
abfun(m, v)
```

Arguments

m	Description
v	Description

Examples

```
abfun(0.26, 10^-4)
```

alpha.u *package internal function*

Description

package internal function

Usage

```
alpha.u(nu, a0, b0, H)
```

Arguments

nu	Description
a0	Description
b0	Description
H	Description

Examples

```
  M = 5
  H = 10
  a = 10^-6
  b = 10^-6
  nu = lapply(1:M, function(x){rbeta(H, a, b)})
  alpha.u(nu, a, b, H)
```

`b.u` *package internal function*

Description

package internal function

Usage

```
b.u(hatpai)
```

Arguments

`hatpai`	Description

Examples

```
pai = 0.5
M = 10
H = 10
K = 5
n = 100

z = abs(rnorm(n))

sigma1 = abs(rnorm(K))
sigma0 = abs(rnorm(K))

mu1 = abs(rnorm(K))
mu0 = abs(rnorm(K))

hk0 = matrix(sample(K, M*H, replace=TRUE), M, H)
hk1 = matrix(sample(K, M*H, replace=TRUE), M, H)

nu.h0 = lapply(1:H, function(x){rbeta(1,5,5)})
nu.h1 = lapply(1:H, function(x){rbeta(1,5,5)})

ph0 = lapply(nu.h0, lambda.u)
ph1 = lapply(nu.h1, lambda.u)

b.u(hatpai.u(z, hk1, hk0, ph1, ph0, sigma1, sigma0, mu1, mu0, pai, H, n))
```

`bhts` *Bayesian High-Throughput Screening*

Description

This is the package main function.

Usage

```
bhts(Z, iters, H, K, mu00=NULL, mu10=NULL, a.alpha, b.alpha, a.tau, b.tau,
pnorm=FALSE, s=NULL, store=FALSE)
```

Arguments

`Z`	A list of compounds.
`iters`	Number of iterations to perform.
`H`	Number of local DP components.
`K`	Number of global DP components.
`mu00`	Activity level (mean) of non-hit compounds
`mu10`	Activity level (mean) of hit compounds
`a.alpha`	Gamma shape parameter specifying local DP concentration prior.
`b.alpha`	Gamma rate parameter specifying local DP concentration prior.
`a.tau`	Gamma shape parameter specifying global DP concentration prior.
`b.tau`	Gamma rate parameter specifying global DP concentration prior.
`pnorm`	Plate normalization. If $TRUE$, each plate is normalized to zero mean and unit variance, prior to analysis. Default is $FALSE$.
`s`	Random seed (for reproducibility purposes). Default is $NULL$.
`store`	If $TRUE$, all samples of certain latent variables are stored in the output object. Default is $FALSE$.

Value

This function returns a list consisting of the following elements:

`hatpai`	A list of vectors of posterior probabilities, estimating the probability of a compound being a hit.
`dat.store`	If $store = TRUE$ (default is $FALSE$), the output contains a list of $iters \times K$ matrices of samples. Each matrix contains the samples of a separate latent variable. At each iteration, the following six variables are stored in a different row of their corrpsponding matrix, $(\lambda_1^{(0)}, \dots, \lambda_K^{(0)})$, $(\lambda_1^{(1)}, \dots, \lambda_K^{(1)})$, $(\mu_{01}, \dots, \mu_{0K})$, $(\mu_{11}, \dots, \mu_{1K})$, $(\sigma_{01}^2, \dots, \sigma_{0K}^2)$ and $(\sigma_{11}^2, \dots, \sigma_{1K}^2)$.

Examples

```
 set.seed(1234)
 Nmax = 100
 M = 100
 n = sample(Nmax, M, replace=TRUE)
 Z = lapply(n, function(x){abs(rnorm(x))})
bhts(Z, iters=100, H=10, K=5, mu00=0, mu10=10, a.alpha=10, b.alpha=5, a.tau=10, b.tau=5)
```

`bhts2HTML` *Convert to HTML*

Description

This function creates an HTML file.

Usage

```
bhts2HTML(dat, dir, fname, title=NULL, bgcolor="#BBBBEE")
```

Arguments

`dat`	An object which is the output of $bhts()$.
`dir`	Directory in which to store the file.
`fname`	File name.
`title`	The title of the html file.
`bgcolor`	Color for the html background.

Examples

```
#See package vignette
```

`data.create` *Create Synthetic Data*

Description

This function generates synthetic compound data.

Usage

```
data.create(N, nr, nc, M, p, s=NULL, covrow=NULL, covcol=NULL, c=0.0001, mat=FALSE)
```

Arguments

`N` Number of compounds per plate.

`nr` Number of plate rows.

`nc` Number of plate columns.

`M` Number of plates.

`p` Probability of a compound being a hit.

`s` Random seed (for reproducibility perposes). Default is $NULL$.

`covrow` Noise plate row-covariance matrix. Default is $NULL$.

`covcol` Noise plate column-covariance matrix. Default is $NULL$.

`c` Constant for scaling plate noise. Default is 0.0001.

`mat` Specifies a matrix ($TRUE$) or a vector ($FALSE$) plate format. Default is $FALSE$.

Value

This function returns a list consisting of the following elements:

`Z` A list of matrices ($mat = TRUE$) or vectors ($mat = FALSE$) of compounds.

`B` A list of compound indicators specifying a hit (1) or a non-hit (0).

`I` A list of compound indicators specifying the muxture component (from 1 to K).

Examples

```
#See package vignette
```

`fdr.r` *package internal function*

Description

package internal function

Usage

```
fdr.r(r, hatpai, fdr)
```

Arguments

`r` Description

`hatpai` Description

`fdr` Description

Examples

```
#See package vignette
```

h.pr.u *package internal function*

Description

package internal function

Usage

```
h.pr.u(z, ih, mu, sigma, pk, K, H, n)
```

Arguments

z

ih

mu

sigma

pk

K

H

n

Examples

```
Nmax = 100
K = 5
H = 10
M = 20
n = sample(Nmax, M, replace=TRUE)

z = abs(rnorm(sum(n)))
mu = abs(rnorm(K))
sigma = 1/rgamma(n=K, shape=10, rate=10)

ih = sample(H,sum(n), replace=TRUE)

pk = lambda.u(rbeta(K, 1, 1))

h.pr.u(z, ih, mu, sigma, pk, K, H, n)
```

hatpai.u *package internal function*

Description

package internal function

Usage

```
hatpai.u(z, hk1, hk0, ph1, ph0, sigma1, sigma0, mu1, mu0, pai, H, n)
```

Arguments

z

hk1

hk0

ph1

ph0

sigma1

sigma0

mu1

mu0

pai

H

n

Value

value

Examples

```
pai = 0.5
M = 10
H = 10
K = 5
n = 100

z = abs(rnorm(n))

sigma1 = abs(rnorm(K))
sigma0 = abs(rnorm(K))

mu1 = abs(rnorm(K))
mu0 = abs(rnorm(K))
```

```
hk0 = matrix(sample(K, M*H, replace=TRUE), M, H)
hk1 = matrix(sample(K, M*H, replace=TRUE), M, H)

nu.h0 = lapply(1:H, function(x){rbeta(1,5,5)})
nu.h1 = lapply(1:H, function(x){rbeta(1,5,5)})

ph0 = lapply(nu.h0, lambda.u)
ph1 = lapply(nu.h1, lambda.u)

hatpai.u(z, hk1, hk0, ph1, ph0, sigma1, sigma0, mu1, mu0, pai, H, n)
```

ind.u *package internal function*

Description

package internal function

Usage

```
ind.u(pr)
```

Arguments

pr

Examples

```
  Nmax = 100
  K = 5
  H = 10
  M = 20
  n = sample(Nmax, M, replace=TRUE)

  z = abs(rnorm(sum(n)))
  mu = abs(rnorm(K))
  sigma = 1/rgamma(n=K, shape=10, rate=10)

  hk = sample(K,M*H, replace=TRUE)

  ph = as.vector(sapply(1:M, function(x){lambda.u(rbeta(H, 1, 1))}))

  ind.u(z.pr.u(z, hk, mu, sigma, ph, H, n))
```

lambda.u *package internal function*

Description

package internal function

Usage

```
lambda.u(nu)
```

Arguments

nu

Examples

```
H = 5
nu = rbeta(H, 1, 1)
lambda.u(nu)
```

lg.mu.sig *package internal function*

Description

package internal function

Usage

```
lg.mu.sig(m, v)
```

Arguments

m	Description
v	Description

Examples

```
#See package vignette
```

mu.k.u *package internal function*

Description

package internal function

Usage

```
mu.k.u(k, ik, z, sigma, mu0)
```

Arguments

k

ik

z

sigma

mu0

Value

Describe

Examples

```
  K = 5
  n = 100
  z = abs(rnorm(n))
  sigma = sapply(1:K, function(x){1/rgamma(n=1, shape=10, rate=10)})
  mu0 = 0
  ik = sample(K, n, replace=TRUE)
  sapply(1:K, mu.k.u, ik, z, sigma, mu0)
```

nu.u *package internal function*

Description

package internal function

Usage

```
nu.u(ind, tau, H)
```

Arguments

`ind`

`tau`

`H`

Value

Describe

Examples

```
H = 5
n = 100
tau = rgamma(1, 1, 1)
ind = sample(H, n, replace=TRUE)
nu.u(ind, tau, H)
```

---

pai.u *package internal function*

Description

package internal function

Usage

```
pai.u(b, a.pai, b.pai)
```

Arguments

`b`

`a.pai`

`b.pai`

Value

Describe

Examples

```
n = 100
b = rbinom(n, 1, 0.5)
a.pai = 10^-6
b.pai = 10^-6
pai.u(b, a.pai, b.pai)
```

`ptrace` *Trace (ACF) Plots*

Description

This function outputs trace plots of certain latent variables.

Usage

```
ptrace(res, var, ndisc, nr, nc, type="trace")
```

Arguments

`res` An output object from $bhts()$.

`var` Variable for which to display convergence diagnostic plots. Current options are $mu0$ (displaying $\mu_{01}, \ldots, \mu_{0K}$), $mu1$ (displaying $\mu_{11}, \ldots, \mu_{1K}$), $sigma0$ (displaying $\sigma^2_{01}, \ldots, \sigma^2_{0K}$), $sigma1$ (displaying $\sigma^2_{11}, \ldots, \sigma^2_{1K}$), $pk0$ (displaying $\lambda^{(0)}_1, \ldots, \lambda^{(0)}_K$) and $pk1$ (displaying $\lambda^{(1)}_1, \ldots, \lambda^{(1)}_K$).

`ndisc` Number of iterations for which to discard samples.

`nr` Number of rows in the resulting composite plot.

`nc` Number of columns in the resulting composite plot.

`type` Type of convergence diagnostic. Currently implemented are trace plots (default $type = "trace"$) and ACF plots ($type = "acf"$)

Examples

```
#See package vignette
```

`r.fdr` *Significant Hits*

Description

This function determines significant hits, based on a specified expected FDR.

Usage

```
r.fdr(res, fdr=0.05)
```

Arguments

`res` An output object from $bhts()$.

`fdr` Expected FDR (default is 0.05).

Value

This function returns a list consisting of the following elements:

`res` A data frame containing significant hits and their probabilities.

`r` The computed significant hit probability threshold.

Examples

```
#See package vignette
```

`sig.k.u` *package internal function*

Description

package internal function

Usage

```
sig.k.u(k, ik, z, mu0, a0, b0)
```

Arguments

`k`

`ik`

`z`

`mu0`

`a0`

`b0`

Examples

```
K = 5
n = 100
z = abs(rnorm(n))
mu0 = 0
ik = sample(K, n, replace=TRUE)
a0 = 5
b0 = 5
sapply(1:K, sig.k.u, ik, z, mu0, a0, b0)
```

`tau.u` *package internal function*

Description

package internal function

Usage

```
tau.u(nu, a0, b0)
```

Arguments

`nu`	Description
`a0`	Description
`b0`	Description

Examples

```
K = 5
a = 10^-6
b = 10^-6
nu = rbeta(K, a, b)
tau.u(nu, a, b)
```

`z.pr.u` *package internal function*

Description

package internal function

Usage

```
z.pr.u(z, hk, mu, sigma, ph, H, n)
```

Arguments

`z`

`hk`

`mu`

`sigma`

`ph`

`H`

`n`

Examples

```
Nmax = 100
K = 5
H = 10
M = 20
n = sample(Nmax, M, replace=TRUE)

z = abs(rnorm(sum(n)))
mu = abs(rnorm(K))
sigma = 1/rgamma(n=K, shape=10, rate=10)

hk = sample(K,M*H, replace=TRUE)

ph = as.vector(sapply(1:M, function(x){lambda.u(rbeta(H, 1, 1))}))

z.pr.u(z, hk, mu, sigma, ph, H, n)
```

Index