Package ‘tiltmod’

June 24, 2018

Type Package

Title Exponential Tilting Method for Reproducible Screening of Large Scale Testing Problem

Version 0.0.1

Author Chong Ma

Maintainer Chong Ma <chongm@email.sc.edu>

Description Develop an exponential tilting method by conditioning on the false discovery rate to tilt a two component mixture of Beta distributions, yielding an tilted mixture model. Use a Boosted EM algorithm to fit a two-component mixture of Beta distributions for p-values or left tail areas of test statistics. The Boosted EM algorithm for the mixture model fitting is built in C++, which is quite fast and stable. The package also includes two utility functions to generate tilted false discovery rates and frequency network.

Depends R (>= 3.3.1)

License GPL (>= 2)

LinkingTo Rcpp, BH

Imports Rcpp (>= 0.12.15), stats, foreach, doParallel, plyr, utils, igraph, edgeR

Encoding UTF-8

LazyData true

RoxygenNote 6.0.1

URL https://github.com/chongma1989/tiltmod

NeedsCompilation yes

Archs i386, x64

R topics documented:

etilt *The exponential tilting function*

Description

This function tilts the mixture model fitted from the training tail-areas (or p-values) by conditioning on the average of local fdr's from the testing tail-areas (or p-values)

Usage

```
etilt(xl, xt, f = NULL, h = NULL, m = NULL, interval = NULL,
  rel.tol = .Machine$double.eps^0.25, ...)
```

Arguments

`xl`	The training left-tail areas (or p-values)
`xt`	The testing left-tail areas (or p-values)
`f`	The objective function is tilted. If either xl or f is NULL, f is fitted by `UBMM`. Default is NULL.
`h`	The conditioning function. By default, h = (1-p)/f(x), where $f(x) = (1-p) \times duniform(x) + p \times dbeta(x, \alpha, \beta)$.
`m`	The constant is used to find the optimal theta such that E(h(x))=m. If m is NULL, m = mean(h(xt)). Default is NULL.
`interval`	The interval is used to search the optimal theta. Default is (-100L,100L).
`rel.tol`	the accuracy used in `integrate`.
`...`	Arguments to be passed to `uniroot`.

Value

A list includes theta, tau, tilt_tau, tilt_f, tilt_f0, tilt_f1, respectively.

Examples

```
xl=c(rbeta(100,0.5,0.5),runif(900))
xt=c(rbeta(300,2,3),runif(700))

## Not run:
etilt(xl,xt)

## End(Not run)
```

fnet *Frequency network*

Description

This function displays the frequency network of discovered differentially expressed genes.

Usage

```
fnet(x, Simplify = FALSE, threshold = 0.05, max.ew = 2, max.size = 18,
  node = TRUE, directed = FALSE, print.graph = TRUE, ...)
```

Arguments

`x`	A list of discovered differentially expressed genes
`Simplify`	Logical indicating whether to discard the genes with lower relative freqency than the threshold. Default is FALSE.
`threshold`	A numeric value determining the cutoff point, where the genes are discarded with lower relative frequency than it. Default is 0.05.
`max.ew`	A numeric value. The maximum edge width in the network plot.
`max.size`	A numeric value, displays the maximum node size.
`node`	A logical value, illustrates the feature name in a circle. Default is TRUE.
`directed`	A logical value, indicates whether the edges are shown in directions. Default is FALSE.
`print.graph`	A logical value. Default is TRUE. If FALSE, then do not print the frequency network.
`...`	Arguments to be passed to `plot.igraph`.

Value

A network plot.

Examples

```
x=list(c(1,3,4),c(2,4,5),c(3,5,1),c(4,1,2),c(5,2,3))
## Not run:
fnet(x)
fnet(x,layout=layout.fruchterman.reingold,vertex.color="grey60")

## End(Not run)
```

Marfan *Marfan*

Description

Gene expression data includes the treatment group and 4132 gene expressions.

Usage

```
Marfan
```

Format

A data frame for 101 samples with 4133 variables: `treatment`, `X1`,...,`X4132`. `treatment` contains 41 samples from the control group and 60 samples from the Marfan group.

Source

`https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2174953/`.

MBM *Fit a two-point mixture of Beta distributions*

Description

Fit a two-point mixture of Beta distributions

Usage

```
MBM(x, w = as.numeric(c()), a0 = as.numeric(c()), a1 = as.numeric(c()),
  precision = 1e-06, MaxIter = 10000L)
```

Arguments

`x`	A vector of numeric values
`w`	A vector of two numeric values, representing the weights of two Beta distributions. Default values are 0.5, respectively.
`a0`	Initial values of the alpha and beta for Beta distribution f0. Default values are 1 and 1, respectively.
`a1`	Initial values of the alpha and beta for Beta distribution f1. Default values are 0.5 and 0.5, respectively.
`precision`	The tolerance for convergence. Default value is 1e-6.
`MaxIter`	The maximum iteration for the EM algorithm. Default value is 10000L.

Value

A list of four components, including the converged weight, parameters for Beta distribution f0, parameters for Beta distribution f1, and the convergence iteration, respectively.

Examples

```
x0=rbeta(900,0.8,0.8)
x1=rbeta(100,0.2,0.2)
## Not run:
   MBM(c(x0,x1),w=c(0.8,0.2),a0=c(1,1),a1=c(0.5,0.5))

## End(Not run)
```

pickrell *pickrell*

Description

The RNA-Seq profiles were made of cell lines derived from lymphoblastoid cells from 69 different Yoruba individuals from Ibadan, Nigeria. Pickrell data consists of 40 females and 29 males for 17310 gene counts data, which are well annotated and being at least 1 count-per-million (cpm) in at least 20 samples. The raw RNA-Seq data for pickrell is available in R package *tweeDEseqCountData*.

Usage

```
pickrell
```

Format

A DGEList S4 class, contains the gene count data, sample information, and gene annotation data.

Source

https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3089435/.

power *Empirical power analysis*

Description

This function deals with power analysis by calculating the relevant TypeI error, power, and probability of being significant by given global false discovery rate.

Usage

```
power(q, x, w = NULL, a = NULL, precision = 1e-08, MaxIter = 10000L,
  theta = NULL, alpha = 0.9, type = c("left tail area", "pvalue"),
  rel.tol = .Machine$double.eps^0.25, tol = .Machine$double.eps^0.5)
```

Arguments

`q` A numeric value or a vector of numerical value, represent global false discovery rate used for power analysis.

`x` A vector of numeric values, represent p-values or left-tail areas of test statistics from a differential gene expression study.

`w` A vector of two numeric values, represent the weights of the uniform and Beta distributions. See `UBMM`.

`a` A vector of two initial parameter values for Beta distribution. See `UBMM`.

`precision` The precision for convergence. Default value is 1e-8.

`MaxIter` The maximum iteration for the EM algorhthm.

`theta` A numerical value, represents the exponential tilting parameter for the fitted mixture model from x. Defualt is NULL.

`alpha` A numeric value, used to determine the probably null region in method “m1" (see `tqvalue`). Default is 0.9.

`type` A character value, chosen from “left tail area” and “pvalue”. Default is “left tail area”.

`rel.tol` the accuracy used in `integrate`.

`tol` the accuracy used in `uniroot`.

Value

A dataframe consists of q, TypeI, Power, ProbS, respectively. TypeI, Power, and ProbS are calculated based on the rejection region R(q) and the empirical mixture model for x.

q The global false discovery rates provided in arguments.

TypeI $P(R(q)|H_0)$

power $P(R(q)|H_1)$

ProbS $P(R(q))$

If theta is provided, then the results contain two data frames as above, one is calculated from the non-exponential tilted mixture model and the other from the exponential tilted mixture model, respectively.

Examples

```
x=c(rbeta(50,0.5,0.5),runif(950))
q=seq(0.05,0.95,0.05)
## Not run:
power(q,x,alpha=0.9,type="left tail area”)
power(q,x,theta=2,alpha=0.9,type="left tail area”)

## End(Not run)
```

tqvalue *The exponential tilting mixture model*

Description

This function tilts the mixture model fitted from the training tail-areas (or p-values) by conditioning on the average of local fdr's from the testing tail-areas (or p-values)

Usage

```
tqvalue(xl, xt, w = NULL, a = NULL, precision = 1e-08, MaxIter = 10000L,
  interval = NULL, adjust = TRUE, method = c("m1", "m2"),
  type = c("left tail area", "pvalue"), alpha = 0.9, q = 0.1,
  ncores = 1, rel.tol = .Machine$double.eps^0.25,
  tol = .Machine$double.eps^0.5)
```

Arguments

`xl`	The training left-tail areas (or p-values)
`xt`	The testing left-tail areas (or p-values)
`w`	A vector of two numeric values, representing the weights of the uniform and Beta distributions. See `UBMM`.
`a`	A vector of two initial parameter values for Beta distribution. See `UBMM`.
`precision`	The precision for convergence. Default value is 1e-8.
`MaxIter`	The maximum iteration for the EM algorhthm.
`interval`	A vector of two numeric values, which determines the range to search the optimal theta. Default is c(-1000L,1000L).
`adjust`	Whether or not to do the model adjustment. Default is TRUE.
`method`	A character chosen from m1, m2. Default is m1.
`type`	A character value, chosen from “left tail area” and “pvalue”. Default is “left tail area”.
`alpha`	A numeric value. Used in method “m1” to determine the probably null region. Default is 0.9.
`q`	A numeric value. The global false discovery rate used in method “m2”, to determine the probable null region. Default is 0.1.
`ncores`	The number of cpus used for implementing this function.
`rel.tol`	the accuracy used in `integrate`.
`tol`	the accuracy used in `uniroot`.

Value

A dataframe includes xl, xt, fdr, FDR, tfdr, and tFDR, respectively. fdr and FDR are the local and global false discovery rate for each value of xt. tfdr and tFDR are the corresponding tilted local and global false discovery rate, respectively.

The optimal theta calculated by solving log(E(exp(thetah(x))))-ctheta, where c=mean(h(xt)).

Examples

```
xl=c(rbeta(50,0.2,0.2),runif(950))
xt=c(rbeta(50,0.1,0.1),runif(950))

## Not run:
tqvalue(xl,xt,ncores=4,adjust=FALSE,type="left tail area")

## End(Not run)
```

UBMM *Fit a mixture of uniform and Beta distribution*

Description

Fit a mixture of uniform and Beta distribution

Usage

```
UBMM(x, w = as.numeric(c()), a = as.numeric(c()), precision = 1e-08,
  MaxIter = 10000L)
```

Arguments

`x` A vector of numeric values

`w` A vector of two numeric values, representing the weights of the uniform and Beta distributions. Default values are 0.5, respectively.

`a` Initial values of the alpha and beta for the Beta distribution. Defaults are obtained from MOM estimators.

`precision` The tolerance for convergence. Default value is 1e-8.

`MaxIter` The maximum iteration for the EM algorithm. Default value is 10000L.

Value

A list of three components, including the converged weight, parameters for Beta distribution, and the convergence iteration, respectively.

Examples

```
x0=runif(900)
x1=rbeta(100,0.5,0.5)
UBMM(c(x0,x1),w=c(0.8,0.2),a=c(0.7,0.8))
```

Index