

# Package ‘tiltmod’

June 11, 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, limma

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 6.0.1

**URL** <https://github.com/chongma1989/tiltmod>

**NeedsCompilation** yes

**Archs** i386, x64

## R topics documented:

etilt	2
fnet	2
MBM	3
tqvalue	4
UBMM	5

Index	6
-------	---

---

etilt	<i>The exponential tilting function</i>
-------	---

---

### 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, ...)
```

### 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 <a href="#">UBMM</a> . Default is NULL.
h	The conditioning function. If h is NULL, $h = -\log(f(x))$ . Default is NULL.
m	The constant is used to find the optimal theta such that $E(h(x))=m$ . If m is NULL, $m = \text{mean}(h(xt))$ . Default is NULL.
interval	The interval is used to search the optimal theta. Default is (-0.5,10).
...	Arguments to be passed to <a href="#">uniroot</a> .

### 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	<i>Frequency network</i>
------	--------------------------

---

### Description

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

### Usage

```
fnet(x, Simplify = FALSE, threshold = 0.05, max.ew = 2,
      directed = FALSE, ...)
```

**Arguments**

<code>x</code>	A list of discovered differentially expressed genes
<code>Simplify</code>	logical indicating whether to discard the genes with lower relative frequency than the threshold. Default is FALSE.
<code>threshold</code>	a numeric value determining the cutoff point, where the genes are discarded with lower relative frequency than it. Default is 0.05.
<code>max.ew</code>	a numeric value. The maximum edge width in the network plot.
<code>directed</code>	logical indicating whether the edges are shown in directions. Default is FALSE.
<code>...</code>	Arguments to be passed to <code>plot.igraph</code> .

**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)
```

---

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**

<code>x</code>	A vector of numeric values
<code>w</code>	A vector of two numeric values, representing the weights of two Beta distributions. Default values are 0.5, respectively.
<code>a0</code>	Initial values of the alpha and beta for Beta distribution f0. Default values are 1 and 1, respectively.
<code>a1</code>	Initial values of the alpha and beta for Beta distribution f1. Default values are 0.5 and 0.5, respectively.
<code>precision</code>	The tolerance for convergence. Default value is 1e-6.
<code>MaxIter</code>	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)
```

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 = 10000,
  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^0.6, tol = 1e-06)
```

**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 <a href="#">UBMM</a> .
a	A vector of two initial parameter values for Beta distribution. See <a href="#">UBMM</a> .
precision	The precision for convergence. Default value is 1e-8.
MaxIter	The maximum iteration for the EM algorithm.
interval	A vector of two numeric values, which determines the range to search the optimal theta. Default is c(-100L,100L).
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 <a href="#">integrate</a> .
tol	the accuracy used in <a href="#">uniroot</a> .

**Value**

A dataframe includes `x1`, `xt`, `fdr`, `FDR`, `tldr`, and `tFDR`, respectively. `fdr` and `FDR` are the local and global false discovery rate for each value of `xt`. `tldr` and `tFDR` are the corresponding tilted local and global false discovery rate, respectively.

The optimal  $\theta$  calculated by solving  $\log(E(\exp(\theta h(x)))) - c\theta$ , where  $c = \text{mean}(h(xt))$ .

**Examples**

```
x1=c(rbeta(50,0.2,0.2),runif(950))
xt=c(rbeta(50,0.1,0.1),runif(950))

## Not run:
tqvalue(x1,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**

<code>x</code>	A vector of numeric values
<code>w</code>	A vector of two numeric values, representing the weights of the uniform and Beta distributions. Default values are 0.5, respectively.
<code>a</code>	Initial values of the alpha and beta for the Beta distribution. Defaults are obtained from MOM estimators.
<code>precision</code>	The tolerance for convergence. Default value is 1e-8.
<code>MaxIter</code>	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

etilt, [2](#)

fnet, [2](#)

integrate, [4](#)

MBM, [3](#)

plot.igraph, [3](#)

tqvalue, [4](#)

UBMM, [2](#), [4](#), [5](#)

uniroot, [2](#), [4](#)