Package ‘TargetScore’

May 25, 2024

**Type**  Package  
**Title**  TargetScore: Infer microRNA targets using microRNA-overexpression data and sequence information  
**Version**  1.42.0  
**Date**  2013-Oct-17  
**Author**  Yue Li  
**Maintainer**  Yue Li <yueli@cs.toronto.edu>  
**Description**  Infer the posterior distributions of microRNA targets by probabilistically modelling the likelihood microRNA-overexpression fold-changes and sequence-based scores. Variational Bayesian Gaussian mixture model (VB-GMM) is applied to log fold-changes and sequence scores to obtain the posteriors of latent variable being the miRNA targets. The final targetScore is computed as the sigmoid-transformed fold-change weighted by the averaged posteriors of target components over all of the features.  
**Depends**  pracma, Matrix  
**Suggests**  TargetScoreData, gplots, Biobase, GEOquery  
**License**  GPL-2  
**URL**  http://www.cs.utoronto.ca/~yueli/software.html  
**Lazyload**  yes  
**biocViews**  miRNA  
**git_url**  https://git.bioconductor.org/packages/TargetScore  
**git_branch**  RELEASE_3_19  
**git_last_commit**  c2ccdb4  
**git_last_commit_date**  2024-04-30  
**Repository**  Bioconductor 3.19  
**Date/Publication**  2024-05-24
**TargetScore-package**

**Contents**

TargetScore-package .................................................. 2  
bsxfun.se ................................................................. 3  
dot.ext ................................................................. 4  
getTargetScores ....................................................... 5  
initialization ........................................................... 6  
logmvgamma ............................................................. 7  
logsumexp ............................................................... 8  
sort_components ....................................................... 9  
targetScore ............................................................ 10  
vbgmm ................................................................. 11  
vbound ................................................................. 13  
vexp ................................................................. 14  
vmax ................................................................. 15  

**Index**

17

---

**Description**

Infer the posterior distributions of microRNA targets by probabilistically modeling the likelihood microRNA-overexpression fold-changes and sequence-based scores. Variational Bayesian Gaussian mixture model (VB-GMM) is applied to log fold-changes and sequence scores to obtain the posteriors of latent variable being the miRNA targets. The final targetScore is computed as the sigmoid-transformed fold-change weighted by the averaged posteriors of target components over all of the features.

**Details**

<table>
<thead>
<tr>
<th>Package</th>
<th>TargetScore</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type</td>
<td>Package</td>
</tr>
<tr>
<td>Version</td>
<td>1.1.5</td>
</tr>
<tr>
<td>Date</td>
<td>2013-10-15</td>
</tr>
<tr>
<td>License</td>
<td>GPL-2</td>
</tr>
</tbody>
</table>

The front-end main function **targetScore** should be used to obtain the probabilistic score of miRNA target. The workhorse function is **vbgmm**, which implementates multivariate variational Bayesian Gaussian mixture model.

**Author(s)**

Yue Li <yueli@cs.toronto.edu>
bsxfun.se

References
Lim, L. P., Lau, N. C., Garrett-Engele, P., Grimson, A., Schelter, J. M., Castle, J., Bartel, D. P.,
Linsley, P. S., and Johnson, J. M. (2005). Microarray analysis shows that some microRNAs down-
215-233.
and Statistics. NY, USA. (p474-486)

See Also
targetScore

Examples
library(TargetScore)
ls("package:TargetScore")

bsxfun.se

bsxfun with single expansion (real Matlab style) (Internal function)

Description
Depending on the dimension of x, repeat y in either by row or by column and apply element-wise
operation defined by func.

Usage
bsxfun.se(func, x, y, expandByRow = TRUE)

Arguments
func function with two or more input parameters.
x, y two vectors, matrices, or arrays
expandByRow expand by row or by column of x when nrow(x)==ncol(x)==length(y)

Details
The function is used by vbgmm.

Value
func(x, y) A matrix of having the same dimension of x.

Note
Internal function.
Author(s)
    Yue Li

See Also
    bsxfun

Examples
    bsxfun.se("*", matrix(c(1:10), nrow=2), matrix(c(1:5), nrow=5))

---

dot.ext

Elementwise dot product (modified dot function) (Internal function)

Description
    Same as dot but handle single row matrix differently by multiplying each value but not sum them up

Usage
    dot.ext(x, y, mydim)

Arguments
    x          numeric vector or matrix
    y          numeric vector or matrix
    mydim      Elementwise product (if 1); otherwise defined by dot

Details
    Returns the 'dot' or 'scalar' product of vectors or columns of matrices. Two vectors must be of same length, two matrices must be of the same size. If x and y are column or row vectors, their dot product will be computed IF mydim is 1 (only difference from dot).

Value
    A scalar or vector of length the number of columns of x and y.

Author(s)
    Yue Li

See Also
    dot
getTargetScores

Examples

dot.ext(1:5, 1:5)
dot.ext(1:5, 1:5, 1)

getTargetScores

Compute targetScore of an overexpressed human microRNA

Description

Obtain for each gene the targetScore using using pre-computed (logFC) TargetScan context score and PCT as sequence score. TargetScanData package is needed.

Usage

getTargetScores(mirID, logFC, ...)

Arguments

mirID: A character string of microRNA ID (e.g., hsa-miR-1)
logFC: N x D numeric vector or matrix of logFC with D replicates for N genes.
...: Parameters passed to vbgmm

Details

This is a convenient function for computing targetScore for a human miRNA using user-supplied or pre-computed logFC and (if available) two pre-computed sequence scores namely TargetScan context score and PCT (probability of conserved targeting). The function also searches for any validated targets from the MirTarBase human validated target list. The function requires TargetScanData to be installed first.

Value

targetScores: numeric matrix of probabilistic targetScores together with the input variable and a binary vector indicating whether each gene is a validated target (if available).

Author(s)

Yue Li

References

See Also

`targetScore`

Examples

```r
if(interactive()) {
  library(TargetScoreData)
  library(Biobase)
  library(GEOquery)

  # compute targetScore from pre-computed logFC and sequence socres
  # for hsa-miR-1
  mir1.score <- getTargetScores("hsa-miR-1", tol=1e-3, maxiter=200)

  # download fold-change data from GEO for hsa-miR-124 overexpression in HeLa
  gset <- getGEO("GSE2075", GSEMatrix =TRUE, AnnotGPL=TRUE)
  if (length(gset) > 1) idx <- grep("GPL1749", attr(gset, "names")) else idx <- 1
  gset <- gset[[idx]]
  sampleinfo <- as.character(pData(gset)$title)
  geneInfo <- fData(gset)

  # only 24h data are used (discard 12h data)
  logfc.mir124 <- as.matrix(exprs(gset)[,
      grep("HeLa transfected with miR-1 versus control transfected HeLa, 24 hours", sampleinfo)])
  rownames(logfc.mir124) <- geneInfo$`Gene symbol`
  mir124.score <- getTargetScores("hsa-miR-124", logfc.mir124, tol=1e-3, maxiter=200)
  head(mir124.score)
}
```

### initialization

**Initialization of latent variable assignments (responsibility) of the VB-GMM (Internal function)**

**Description**

Initialize latent variables based on the number of components. The function is run before the VB-EM iteration in vbgmm.

**Usage**

`initialization(X, init)`
logmvgamma

Arguments

X  D x N numeric vector or matrix of observations
init  Based on the dimension, init is expected to be one of the followings: scalar: number of components; vector: initial class labels; matrix: initialize with a D x K matrix for D variables and K components.

Details

The function is expected to be used by vbgmm to initialize assignments of latent variables before VM-EM iterations.

Value

R  N by K matrix for N observations and K latent components (defined by init)

Author(s)

Yue Li

References


See Also

vbgmm

Examples

tmp <- initialization(matrix(c(rnorm(100,mean=2), rnorm(100,mean=3)),nrow=1), init=2)

logmvgamma  Logarithmic multivariate Gamma function (Internal function)

Description

Compute logarithm multivariate Gamma function.

Usage

logmvgamma(x, d)

Arguments

x  numeric vector or matrix
d  dimension
Details

\( \Gamma_p(x) = \pi^\frac{p(p-1)}{4} \prod_{j=1}^{p} \Gamma(x+(1-j)/2) \)

\[ \log \Gamma_p(x) = \frac{p(p-1)}{4} \log \pi + \sum_{j=1}^{p} \log \Gamma(x+(1-j)/2) \]

Value

Matrix of the same dimension as \( x \).

Author(s)

Yue Li

References

Mo Chen (2012). Matlab code for Variational Bayesian Inference for Gaussian Mixture Model. 

See Also

\( \text{lgamma} \)

Examples

\[
\logmvgamma(\text{matrix}(1:6, nrow=3), 2)
\]

---

### logsumexp

**Compute \( \log(\text{sum}(\exp(x), \text{dim})) \) while avoiding numerical underflow**

*(Internal function)*

Description

Compute \( \log(\text{sum}(\exp(x), \text{dim})) \) while avoiding numerical underflow.

Usage

\[
\text{logsumexp}(x, \text{margin} = 1)
\]

Arguments

- \( x \) numeric vector or matrix
- \( \text{margin} \) dimension to apply summation

Value

numeric vector or matrix of the same columns or rows (depending on \( \text{margin} \)) as \( x \)
sort_components

Author(s)
Yue Li

References

Examples
logsumexp(matrix(c(1:5)), 2)

sort_components | Sort mixture components in increasing order of averaged means (Internal function)

Description
Sort Gaussian mixture components with model parameters in increasing order of averaged means of d variables.

Usage
sort_components(model)

Arguments
model | A list containing trained parameters of the Bayesian GMM (see Value section in vbgmm).

Value
VB-GMM model list in increasing order of averaged means.

Author(s)
Yue Li

See Also
vbgmm

Examples
tmp <- vbgmm(c(rnorm(100, mean=2), rnorm(100, mean=3)), tol=1e-3)
tmp$mu
| targetScore | Probabilistic score of genes being the targets of an overexpressed microRNA |

**Description**

Given the overexpression fold-change and sequence-scores (optional) of all of the genes, calculate for each gene the TargetScore as a probability of miRNA target.

**Usage**

```
targetScore(logFC, seqScores, ...)  
```

**Arguments**

- **logFC**  
  numeric vector of log fold-changes of N genes in treatment (miRNA overexpression) vs control (mock).
- **seqScores**  
  N x D numeric vector or matrix of D sequence-scores for N genes. Each score vector is expected to be equal to or less than 0. The more negative the scores, the more likely the corresponding target.
- **...**  
  Parameters passed to `vbgmm`.

**Details**

Given expression fold-change (due to miRNA transfection), we use a three-component VB-GMM to infer down-regulated targets accounting for genes with little or positive log fold-change (due to off-target effects (Khan et al., 2009). Otherwise, two-component VB-GMM is applied to unsigned sequence scores (seqScores). The parameters for the VB-GMM are optimized using Variational Bayesian Expectation-Maximization (VB-EM) algorithm. Presumably, the mixture component with the largest absolute means of observed negative fold-change or sequence score is associated with miRNA targets and denoted as "target component". The other components correspond to the "background component". It follows that inferring miRNA-mRNA interactions most likely explained by the observed data is equivalent to inferring the posterior distribution of the target component given the observed variables. The targetScore is computed as the sigmoid-transformed fold-change weighted by the averaged posteriors of target components over all of the features. Specifically, we define the targetScore as a composite probabilistic score of a gene being the target t of a miRNA:

\[
\text{sigmoid}(-\text{logFC}) \left( \frac{1}{K+1} \sum_{x} p(t \mid x) \right),
\]

where \( \text{sigmoid}(-\text{logFC}) = \frac{1}{1 + \exp(\text{logFC})} \) and \( p(t \mid x) \) is the posterior of the first component computed by `vbgmm`.

**Value**

- **targetScore**  
  numeric vector of probabilistic targetScores for N genes

**Author(s)**

Yue Li
References


See Also

vbgmm

Examples

# A toy example:
# 10 down-reg, 1000 unchanged, 90 up-reg genes
# due to overexpressing a miRNA
trtm <- c(rnorm(10,mean=0.01), rnorm(1000,mean=1), rnorm(90,mean=2)) + 1e3
tcrl <- c(rnorm(1100,mean=1)) + 1e3
logFC <- log2(trtm) - log2(tcrl)

# 8 out of the 10 down-reg genes have prominent seq score A
seqScoreA <- c(rnorm(8,mean=-2), rnorm(1092,mean=0))

# 10 down-reg genes plus 10 more genes have prominent seq score B
seqScoreB <- c(rnorm(20,mean=-2), rnorm(1080,mean=0))

seqScores <- cbind(seqScoreA, seqScoreB)
p.targetScore <- targetScore(logFC, seqScores, tol=1e-3)

vbgmm

Variational Bayesian Gaussian mixture model (VB-GMM)

Description

Given a N x D matrix of N observations and D variables, compute VB-GMM via VB-EM.

Usage

vbgmm(data, init = 2, prior, tol = 1e-20, maxiter = 2000, mirprior = TRUE, expectedTargetFreq = 0.01, verbose = False)
Arguments

data: N x D numeric vector or matrix of N observations (rows) and D variables (columns)

init: Based on the dimension, init is expected to be one of the followings: scalar: number of components; vector: initial class labels; matrix: initialize with a D x K matrix for D variables and K components.

prior: A list containing the hyperparameters including alpha (Dirichlet), m (Gaussian mean), kappa (Gaussian variance), v (Wishart degree of freedom), M (Wishart precision matrix).

tol: Threshold that defines termination/convergence of VB-EM when abs(L[t] - L[t-1])/abs(L[t]) < tol

maxiter: Scalar for maximum number of EM iterations

mirprior: Boolean to indicate whether to use expectedTargetFreq to initialize alpha0 for the hyperparameters of Dirichlet.

expectedTargetFreq: Expected target frequency within the gene population. By default, it is set to 0.01, which is consistent with the widely accepted prior knowledge that 200/20000 targets per miRNA.

verbose: Boolean indicating whether to show progress in terms of lower bound (vbound) of VB-EM (default: FALSE)

Details

The function implements variation Bayesian multivariate GMM described in Bishop (2006). Please refer to the reference below for more details. This is the workhorse of targetScore. Alternatively, user can choose to apply this function to other problems other than miRNA target prediction.

Value

A list containing:

label: a vector of maximum-a-posteriori (MAP) assignments of latent discrete values based on the posteriors of latent variables.

R: N x D matrix of posteriors of latent variables

mu: Gaussian means of the latent components

full.model: A list containing posteriors R, logR, and the model parameters including alpha (Dirichlet), m (Gaussian mean), kappa (Gaussian variance), v (Wishart degree of freedom), M (Wishart precision matrix)

L: A vector of variational lower bound at each EM iterations (should be strictly increasing)

Author(s)

Yue Li
**vbound**

---

**References**


**See Also**

*targetScore*

**Examples**

```r
X <- c(rnorm(100, mean=2), rnorm(100, mean=3))
tmp <- vbgmm(X, tol=1e-3)
names(tmp)
```

---

**Description**

Evaluate variational lower bound to determine when to stop VB-EM iteration (convergence).

**Usage**

```r
vbound(X, model, prior)
```

**Arguments**

- **X**: D x N numeric vector or matrix of N observations (columns) and D variables (rows)
- **model**: List containing model parameters (see `vbgmm`)
- **prior**: numeric vector or matrix containing the hyperparameters for the prior distributions

**Value**

A continuous scalar indicating the lower bound (the higher the more converged)

**Note**

X is expected to be D x N for N observations (columns) and D variables (rows)

**Author(s)**

Yue Li
vexp

Variational-Expectation in VB-EM (Internal function)

Description
The E step in VB-EM iteration.

Usage
vexp(X, model)

Arguments
X
D x N numeric vector or matrix of N observations (columns) and D variables (rows)

model
List containing model parameters (see vbgmm)

Value
model
A list containing the updated model parameters including alpha (Dirichlet), m (Gaussian mean), kappa (Gaussian variance), v (Wishart degree of freedom), M (Wishart precision matrix).

Note
X is expected to be D x N for N observations (columns) and D variables (rows)

Author(s)
Yue Li
References


See Also

vbgmm

Examples

```r
X <- c(rnorm(100,mean=2), rnorm(100,mean=3))
tmp <- vbgmm(X, tol=1e-3)
dim(tmp$R); head(tmp$R)
```

**vmax**

Variational-Maximimization in VB-EM (Internal function)

Description

The M step in VB-EM iteration.

Usage

`vmax(X, model, prior)`

Arguments

- `X` D x N numeric vector or matrix of N observations (columns) and D variables (rows)
- `model` List containing model parameters (see `vbgmm`)
- `prior` List containing the hyperparameters defining the prior distributions

Value

- `model` A list containing the updated model parameters including alpha (Dirichlet), m (Gaussian mean), kappa (Gaussian variance), v (Wishart degree of freedom), M (Wishart precision matrix).

Note

- X is expected to be D x N for N observations (columns) and D variables (rows)

Author(s)

Yue Li
References


See Also

vbgmm

Examples

X <- c(rnorm(100,mean=2), rnorm(100,mean=3))
tmp <- vbgmm(X, tol=1e-3)
names(tmp$full.model)
Index

* microRNA target prediction
    TargetScore-package, 2
* package
    TargetScore-package, 2
    TargetScore-package, 2
bsxfun, 4
bsxfun.se, 3
dot, 4
dot.ext, 4
getTargetScores, 5
initialization, 6
lgamma, 8
logmvgamma, 7
logsumexp, 8
sort_components, 9
TargetScore (TargetScore-package), 2
targetScore, 2, 3, 6, 10, 12, 13
TargetScore-package, 2
vbgmm, 2, 5, 7, 9–11, 11, 13–16
vbound, 12, 13
vexp, 14
vmax, 15