Package ‘DPBBM’
September 29, 2016

Type Package
Title Dirichlet Process Beta-Binomial Mixture
Version 0.2.5
Date 2016-09-21
Author Lin Zhang
Maintainer Lin Zhang <lin.zhang@cumt.edu.cn>
Depends R (>= 3.1.0)
Imports tmvtnorm, VGAM, gplots, CEoptim
Description Beta-binomial Mixture Model is used to infer the pattern from count data.
It can be used for clustering of RNA methylation sequencing data.
License GPL (>= 2)
NeedsCompilation no
Repository CRAN
Date/Publication 2016-09-29 08:02:30

R topics documented:

DPBBM-package ..................................................... 1
bbm_data_generate ............................................... 2
BCubed_metric ....................................................... 3
dpbbm_mc_iterations ............................................. 4

Index

DPBBM-package

Description
This package is developed for the beta-binomial mixture model based clustering

Details
bbm_data_generate

Package: DPBBM
Type: Package
Version: 1.0.1
Date: 2016-06-26
License: GPL-2

References
Coming soon!

Examples

# Please check the main function of the package
?dpbbm_mc_iterations

bbm_data_generate  bbm_data_generate

Description
This is to generate the simulation data based on Beta-bionomial mixture model

Usage

bbm_data_generate(S=3, G=50, K=3, prob=rep(1, times=3),
                   alpha_band=c(2, 6),
                   beta_band=c(2, 6),
                   nb_mu=100, nb_size=0.2, plotf = FALSE,
                   max_cor=0.5)

Arguments

S  Number of samples in the simulated data
G  Number of sites in the simulated data
K  Number of clusters that exist in the simulated data
prob  the cluster weight for each cluster
alpha_band  the region used to generate the parameter of beta distribution alpha
beta_band  the region used to generate the parameter of beta distribution beta
nb_mu  alternative parametrization via mean for Negative Binomial distribution
nb_size  target for number of successful trials, or dispersion parameter (the shape parameter of the gamma mixing distribution) for Negative binomial distribution. Must be strictly positive, need not be integer.
BCubed_metric

plotf  option for whether plot the generated data according to clusters or not
max_cor  The maximized correlation allowed for the simulated data, which used to guarantee the data is not highly correlated.

Details
The Dirichlet Process based beta-binomial mixture model clustering

Value
The function returns simulation data generated based on beta binomial mixture model

Author(s)
Lin Zhang, PhD <lin.zhang@cumt.edu.cn>

References
Reference coming soon!

Examples
```r
set.seed(123455)
S <- 4
G <- 100
K <- 3
nb_mu <- 100
nb_size <- 0.8
prob <- c(1,1,1)
mat <- bbm_data_generate(S=S,G=G,K=K,prob=prob,alpha_band=c(2,6),beta_band=c(2,6),
            nb_mu=nb_mu,nb_size=nb_size, plotf = TRUE, max_cor=0.5)
table(mat$gamma)
pie(mat$gamma)
id <- order(mat$gamma);
c <- mat$gamma[id]
mat_ratio <- (mat$k+1)/(mat$n+1);
heatmap(mat_ratio[id,], Rowv = NA, Colv = NA, scale="none", RowSideColors=as.character(c),
            xlab = "4 samples", ylab="100 RNA methylation sites")
```

BCubed_metric  BCubed_metric

Description
This is to evaluate the clustering method with Bcubed F score.

Usage
```r
BCubed_metric(L, C, alpha)
```
Arguments

- `L` real label of classes
- `C` classification label of classes drawn by clustering method
- `alpha` F metric parameter which used to average precision and recall

Details

The clustering evaluation method based on Bcubed F metric.

Value

The function returns Bcubed F score of the clustering method. The higher the value is, the better performance the clustering method can get.

Author(s)

Lin Zhang, PhD <lin.zhang@cumt.edu.cn>

References

Reference coming soon!

Examples

```r
L <- c(1,1,2,1,1,2,2)
C <- c(2,2,1,2,1,1,1)
alpha <- 0.5
Bcubed_score <- BCubed_metric(L, C, alpha)
Bcubed_score
```

Description

This is the Markov Chain Monte Carlo iterations for DPBBM

Usage

```r
dpbbm_mc_iterations(x, size.x, m = 1, max_iter = 2000,
a = 0.1, b = 1, tau = 1,
sig_alpha = 25/9, sig_beta = 25/9,
tau.method = "auto", debug = FALSE)
```
Arguments

x a matrix of k for clustering, referring to IP reads in m6A seq data
size.x a matrix of n for clustering, referring to the summation of IP reads and input reads in m6A seq data
m a value indicating the auxiliary clusters used in DPBBM
max_iter maximized iterations in DPBBM
a Hyperparameter a for tau
b Hyperparameter b for tau
tau Prior for tau
sig_alpha variation for parameter alpha of beta distribution
sig_beta variation for parameter beta of beta distribution
tau.method tau.method should be set to "auto" or "stable", refer to tau for detail description.
debug whether DPBBM print the debug info or not. Default: FALSE

Details

The Dirichlet Process based beta-binomial mixture model clustering

Value

The function returns the cluster label withdrawn by DPBBM

Author(s)

Lin Zhang, PhD <lin.zhang@cumt.edu.cn>

References

Reference coming soon!

Examples

# generate a simulated dataset
set.seed(123455)
S <- 4
G <- 100
K <- 3
nb_mu <- 100
nb_size <- 0.8
prob <- c(1,1,1)
mat <- bbm_data_generate(S=S,G=G,K=K,prob=prob, alpha_band=c(2,6), beta_band=c(2,6),
                          nb_mu=nb_mu, nb_size=nb_size, plotf = FALSE, max_cor=0.5)
# check generated data
id <- order(mat$gamma);
c <- mat$gamma[id]
mat_ratio <- (mat$k+1)/(mat$n+1);
heatmap(mat_ratio[id,], Rowv = NA, Colv = NA, scale="none", RowSideColors=as.character(c),
xlab = "4 samples", ylab="100 RNA methylation sites")

## Run the DPBBM result. This step takes a really long time.
## You are suggested to check the pre-prepared example for a quick start
F=system.file("extdata", "DPBBM_example.html", package="DPBBM")
browseURL(url=F)

## Alternatively
# cluster_label <- dpbbm_mc_iterations(mat$k, mat$n)
# Show the clustering result.
# table(cluster_label)
# pie(table(mat$gamma))
#
# # Compare the clustering result with the true clustering IDs.
# id <- order(mat$gamma);
# c <- cluster_label
# r <- rainbow(3, start = 0, end = 0.3)
# mat_ratio <- (mat$k + 1)/(mat$n + 1);
# heatmap(mat_ratio[id,], Rowv = NA, Colv = NA, scale="none",
# RowSideColors = as.character(cluster_label[id]),
# margins = c(3,25))
Index

*Topic statistical Inference
  bbm_data_generate, 2
  BCubed_metric, 3
  dpbbm_mc_iterations, 4

bbm_data_generate, 2
BCubed_metric, 3

DPBBM (DPBBM-package), 1
DPBBM-package, 1
dpbbm_mc_iterations, 4