Package 'cobiclust'

July 22, 2025

Type Package

Title Biclustering via Latent Block Model Adapted to Overdispersed Count Data

Version 0.1.2

Description Implementation of a probabilistic method for biclustering adapted to overdispersed count data. It is a Gamma-Poisson Latent Block Model. It also implements two selection criteria in order to select the number of biclusters.

License GPL-3

URL https://github.com/julieaubert/cobiclust

BugReports https://github.com/julieaubert/cobiclust/issues

Depends R (>= 3.5.0)

Imports assertthat, cluster, stats, testthat

Suggests spelling

Encoding UTF-8

Language en-US

RoxygenNote 7.2.3

NeedsCompilation no

Author Julie Aubert [aut, cre] (ORCID: <https://orcid.org/0000-0001-5203-5748>), INRAE [cph]

Maintainer Julie Aubert <julie.aubert@inrae.fr>

Repository CRAN

Date/Publication 2024-02-16 12:10:02 UTC

Contents

`cobiclust` *Perform a biclustering adapted to overdispersed count data.*

Description

Perform a biclustering adapted to overdispersed count data.

Usage

```
cobiclust(
  x,
  K = 2,
  G = 3,
  nu_j = NULL,
  a = NULL,
  akg = FALSE,
  cvg_lim = 1e-05,
  nbiter = 5000,
  tol = 1e-04
)
```

Arguments

`x`	the input matrix of observed data.
`K`	an integer specifying the number of groups in rows.
`G`	an integer specifying the number of groups in columns.
`nu_j`	a vector of numeric, corresponding of a column (sampling effort) effect.
`a`	a numeric dispersion parameter (parameter of the gamma distribution).
`akg`	a logical variable indicating whether to use a common dispersion parameter (`akg = FALSE`) or not.
`cvg_lim`	a number specifying the threshold used for convergence criterion.
`nbiter`	the maximal number of iterations for the global loop of variational EM algorithm (`nbiter = 5000` by default).
`tol`	the level of relative iteration convergence tolerance (`tol = 1e-04` by default).

Value

An object of class cobiclustering

See Also

`cobiclustering` for the cobiclustering class.

Examples

```
npc <- c(50, 40) # nodes per class
KG <- c(2, 3) # classes
nm <- npc * KG # nodes
Z <- diag(KG[1]) %x% matrix(1, npc[1], 1)
W <- diag(KG[2]) %x% matrix(1, npc[2], 1)
L <- 70*matrix(runif(KG[1] * KG[2]), KG[1], KG[2])
M_in_expectation <- Z %*% L %*% t(W)
size <- 50
M <- matrix(
  rnbinom(
    n = length(as.vector(M_in_expectation)),
    mu = as.vector(M_in_expectation), size = size
  ),
  nm[1], nm[2]
)
rownames(M) <- paste('OTU', 1:nrow(M), sep = '_')
colnames(M) <- paste('S', 1:ncol(M), sep = '_')
res <- cobiclust(M, K = 2, G = 3, nu_j = rep(1, 120), a = 1 / size, cvg_lim = 1e-5)
```

`selection_criteria` *Calculate selection criteria.*

Description

Calculate selection criteria.

Usage

```
selection_criteria(x, K = NULL, G = NULL)
```

Arguments

`x`	The output of the cobiclust function.
`K`	The number of groups in rows.
`G`	The number of groups in columns.

Value

A dataframe with 7 columns.

`vICL` the vICL selection criterion.

`BIC` the BIC selection criterion.

`penKG` the value of the BIC penalty.

`lb` the value of the lower bound of the log-likelihood.

`entZW` the value of the entropy of the latent variables Z and W.

`K` the number of groups in rows.

`G` the number of groups in columns.

Index