Package ‘CAEN’

October 15, 2025

Type Package

Title Category encoding method for selecting feature genes for the classification of single-cell RNA-seq

Version 1.17.0

Description With the development of high-throughput techniques, more and more gene expression analysis tend to replace hybridization-based microarrays with the revolutionary technology.The novel method encodes the category again by employing the rank of samples for each gene in each class. We then consider the correlation coefficient of gene and class with rank of sample and new rank of category. The highest correlation coefficient genes are considered as the feature genes which are most effective to classify the samples.

License GPL-2

Encoding UTF-8

LazyData true

biocViews DifferentialExpression, Sequencing, Classification, RNASeq, ATACSeq, SingleCell, GeneExpression, RIPSeq

Depends R (>= 4.1)

Suggests knitr,rmarkdown,BiocManager,SummarizedExperiment,BiocStyle

VignetteBuilder knitr

RoxygenNote 7.1.1

Imports stats,PoiClaClu,SummarizedExperiment,methods

git_url https://git.bioconductor.org/packages/CAEN

git_branch devel

git_last_commit 92ae3a0

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-10-15

Author Zhou Yan [aut, cre]

Maintainer Zhou Yan <2160090406@email.szu.edu.cn>

Contents

CAEN 2
estimatep 3
newCountDataSet 4
realData 5
ZIPLDA 5
ZIPLDA.cv 7

Index **9**

`CAEN` *Compute the correlation coefficient of gene with category number to identify differentially expressed genes*

Description

To Compute the correlation coefficient of gene with category number to identify differentially expressed genes.

Usage

```
CAEN(dataTable, y, K, gene_no_list)
```

Arguments

`dataTable`	would be a SummarizedExperiment Bioconductor object, then it would be transformed into a p times n matrix - i.e. features on the rows and observations on the columns in the function, or dataTable would be a p times n matrix.
`y`	the category for each sample
`K`	the number of class
`gene_no_list`	the number of differentially expressed genes you want to select

Value

list(.) A list of computed correlation coefficient and the first some differentially expressed genes , where "r" represents correlation coefficient between gene and category number, and "np" represents the top differential feature label.

Examples

```
library(SummarizedExperiment)
dat <- newCountDataSet(n=40,p=500,sdsignal=0.1,K=4,param=10,drate=0.4)
x <- dat$sim_train_data
y <- as.numeric(colnames(dat$sim_train_data))
myscore <- CAEN(dataTable=x, y=y, K=4, gene_no_list=100)
```

`estimatep` *Estimate the probability that the read is 0 in a Zero-inflated Poisson model.*

Description

Estimate the probability that the read is 0 in a Zero-inflated Poisson model.

Usage

```
estimatep(x, y, xte=NULL, beta=1, type=c("mle","deseq","quantile"),
prior=NULL)
```

Arguments

`x`	would be a SummarizedExperiment Bioconductor object, then it would be transformed into a n times p matrix - i.e. observations on the rows and features on the columns in the function, or x would be a n times p matrix.
`y`	A numeric vector of class labels of length n: 1, 2, ..., K if there are K classes.Each element of y corresponds to a row of x; i.e. these are the class labels for the observations in x.
`xte`	would be a SummarizedExperiment Bioconductor object, then would be transformed into a m-by-p data matrix: m test observations and p features, or xte would be a m-by-p data matrix. The classifier fit on the training data set x will be tested on this data set. If NULL, then testing will be performed on the training set.
`beta`	A standardized parameter
`type`	the method of normality
`prior`	vector of length equal to the number of classes, representing prior probabilities for each class.If NULL then uniform priors are used (i.e. each class is equally likely)

Value

p the probability that the read is 0 in a Zero-inflated Poisson model

Examples

```
library(SummarizedExperiment)
dat <- newCountDataSet(n=40,p=500, K=4, param=10, sdsignal=0.1, drate=0.4)
x <- dat$sim_train_data
y <- as.numeric(colnames(dat$sim_train_data))
xte <- dat$sim_test_data
prob <- estimatep(x=x, y=y, xte=x, beta=1, type="mle", prior=NULL)
```

`newCountDataSet` *Generate a simulated sequencing data set using a negative binomial model*

Description

Generate two nxp data sets: a training set and a test set, as well as outcome vectors y and yte of length n indicating the class labels of the training and test observations.

Usage

```
newCountDataSet(n, p, K, param, sdsignal,drate)
```

Arguments

`n`	Number of observations desired.
`p`	Number of features desired. Note that drate of the features will differ between classes, though some of those differences may be small.
`K`	Number of classes desired. Note that the function requires that n be at least equal to 4K.i.e. there must be at least 4 observations per class on average.
`param`	The dispersion parameter for the negative binomial distribution. The negative binomial distribution is parameterized using "mu" and "size" in the R function "rnbinom". That is, Y ~ NB(mu, param) means that E(Y)=mu and Var(Y) = mu+mu^2/param.So when param is very large this is essentially a Poisson distribution, and when param is smaller then there is a lot of overdispersion relative to the Poisson distribution.
`sdsignal`	The extent to which the classes are different. If this equals zero then there are no class differences and if this is large then the classes are very different.
`drate`	The proportion of differentially expressed genes

Value

list(.) A list of output, "sim_train_data" represents training data of q*n data matrix. "sim_test_data" represents test data of q*n data matrix. The colnames of this two matrix are class labels for the n observations May have q<p because features with 0 total counts are removed. The q features are those with >0 total counts in dataset. So q <= p. "truesf" denotes size factors for training observations."isDE" represnts the differential gene label.

Examples

```
dat <- newCountDataSet(n=40,p=500, K=4, param=10, sdsignal=0.1,drate=0.4)
```

realData — *A real dataset of gene expression RNA-seq.*

Description

Gene expression is compared between two mouse models, Pkd1f/f: HoxB7-cre mice and Pkd2f/f: HoxB7-cre mice. Each group includes 18 samples, and there are total 29996 transcripts in this dataset.

Usage

```
realData
```

Format

A SummarizedExperiment Bioconductor object containing 29996 transcripts.

Source

Woo, Y., Kim, D., Koo, N., Kim, Y., Lee, S., Ko, J., et al. (2017). Profiling of mirnas and target genes related to cystogenesis in adpkd mouse models. scientific reports 7, 14151.

ZIPLDA — *Classify observations using a Zero-inflated Poisson model.*

Description

Classify observations using a Zero-inflated Poisson model.

Usage

```
ZIPLDA(x, y, xte=NULL, rho = 0, beta = 1, rhos = NULL, prob0=NULL,
type=c("mle","deseq","quantile"),prior = NULL, transform=TRUE, alpha=NULL)
```

Arguments

x	would be a SummarizedExperiment Bioconductor object, then it would be transformed into a n times p matrix - i.e. observations on the rows and features on the columns in the function, or x would be a n times p matrix.
y	A numeric vector of class labels of length n: 1, 2, ..., K if there are K classes.Each element of y corresponds to a row of x; i.e. these are the class labels for the observations in x.
xte	would be a SummarizedExperiment Bioconductor object, then it would be transformed into a m-by-p data matrix: m test observations and p features, or xte would be a m-by-p data matrix. The classifier fit on the training data set x will be tested on this data set. If NULL, then testing will be performed on the training set.

`rho` Tuning parameter controlling the amount of soft thresholding performed, i.e. the level of sparsity, i.e. number of nonzero features in classifier. Rho=0 means that there is no soft-thresolding, i.e. all features used in classifier. Larger rho means that fewer features will be used.

`beta` A smoothing term. A Gamma(beta,beta) prior is used to fit the Zero-inflated Poisson model.Recommendation is to just leave it at 1, the default value.

`rhos` A vector of tuning parameters that control the amount of soft thresholding performed. If "rhos" is provided then a number of models will be fit (one for each element of "rhos"), and a number of predicted class labels will be output (one for each element of "rhos").

`prob0` The probability that the read is 0

`type` How should the observations be normalized within the Zero-inflated Poisson model, i.e. how should the size factors be estimated? Options are "quantile" or "deseq" (more robust) or "mle" (less robust). In greater detail: "quantile" is quantile normalization approach of Bullard et al 2010 BMC Bioinformatics,"deseq" is median of the ratio of an observation to a pseudoreference obtained by taking the geometric mean, described in Anders and Huber 2010 Genome Biology and implemented in Bioconductor package "DESeq", and "mle" is the sum of counts for each sample; this is the maximum likelihood estimate under a simple Zero-inflated Poisson model.

`prior` vector of length equal to the number of classes, representing prior probabilities for each class.If NULL then uniform priors are used (i.e. each class is equally likely)

`transform` should data matrices x and xte first be power transformed so that it more closely fits the Zero-inflated Poisson model? TRUE or FALSE. Power transformation is especially useful if the data are overdispersed relative to the Zero-inflated Poisson model.

`alpha` if transform=TRUE, this determines the power to which the data matrices x and xte are transformed.If alpha=NULL then the transformation that makes the Zero-inflated Poisson model best fit the data matrix x is computed.(Note that alpha is computed based on x, not based on xte). Or a value of alpha, 0<alpha<=1, can be entered by the user.

Value

list(.) A list of output, "ytehat" represents The predicted class labels for each of the test observations(rows of xte)."discriminant" represents A m-by-K matrix, where K is the number of classes. The (i,k) element is large if the ith element of xte belongs to class k."ds" A K-by-p matrix indicating the extent to which each feature is under-or over-expressed in each class. The (k,j) element is >1 if feature j is over-expressed in class k, and is <1 if feature j is under-expressed in class k. When rho is large then many of the elemtns of this matrix are shrunken towards 1(no over- or under-expression)."alpha" represents Power transformation used (if transform=TRUE).

Examples

```
library(SummarizedExperiment)
dat <- newCountDataSet(n=40,p=500, K=4, param=10, sdsignal=0.1, drate=0.4)
x <- dat$sim_train_data
y <- as.numeric(colnames(dat$sim_train_data))
xte <- dat$sim_test_data
prob <- estimatep(x=x, y=y, xte=x, beta=1, type="mle", prior=NULL)
```

```
prob0 <- estimatep(x=x, y=y, xte=xte, beta=1, type="mle", prior=NULL)
cv.out <- ZIPLDA.cv(x=x, y=y, prob0=t(prob))
out <- ZIPLDA(x=x, y=y, xte=xte, rho=cv.out$bestrho, prob0=t(prob0))
```

`ZIPLDA.cv` *Function to do cross-validation for zero-inflated Poisson classification.*

Description

Perform cross-validation for the function that implements the "sparse zero-inflated Poisson linear discriminant analysis classifier",which is similar to linear discriminant analysis but assumes a zero-inflated Poisson model rather than a Gaussian model for the data. The classifies soft-thresholds the estimated effect of each feature in order to achieve sparsity. This cross-validation function selects the proper value of the tuning parameter that controls the level of soft-thresholding.

Usage

```
ZIPLDA.cv(x, y, rhos = NULL, beta = 1, nfolds = 5, prob0=NULL,
type=c("mle","deseq","quantile"),folds = NULL, transform=TRUE, alpha=NULL,
prior=NULL)
```

Arguments

`x`	would be a SummarizedExperiment Bioconductor object, then it would be transformed into a n times p matrix - i.e. observations on the rows and features on the columns in the function, or x would be a n times p matrix.
`y`	A numeric vector of class labels of length n: 1, 2, ..., K if there are K classes.Each element of y corresponds to a row of x; i.e. these are the class labels for the observationsin x.
`rhos`	A vector of tuning parameters to try out in cross-validation. Rho controls the level of shrinkage performed, i.e. the number of features that are not involved in the classifier. When rho=0 then all features are involved in the classifier, and when rho is very large no features are involved. If rhos=NULL then a vector of rho values will be chosen automatically.
`beta`	A smoothing term. A Gamma(beta,beta) prior is used to fit the zero-inflated Poisson model.Recommendation is to leave it at 1, the default value.
`nfolds`	The number of folds in the cross-validation; default is 5-fold cross-validation.
`prob0`	The probability that the read is 0
`type`	How should the observations be normalized within the zero-inflated Poisson model, i.e. how should the size factors be estimated? Options are "quantile" or "deseq" (more robust) or "mle" (less robust). In greater detail: "quantile" is quantile normalization approach of Bullard et al 2010 BMC Bioinformatics, "deseq" is median of the ratio of an observation to a pseudoreference obtained by taking the geometric mean, described in Anders and Huber 2010 Genome Biology and implemented in Bioconductor package "DESeq", and "mle" is the sum of counts for each sample; this is the maximum likelihood estimate under a simple Poisson model.

`folds` Instead of specifying the number of folds in cross-validation, one can explicitly specify the folds. To do this, input a list of length r(to perform r-fold cross-validation). The rth element of the list should be vector containing the indices of the test observations in the rth fold.

`transform` Should data matrices x and xte first be power transformed so that it more closely fits the zero-inflated Poisson model? TRUE or FALSE. Power transformation is especially useful if the data are overdispersed relative to the zero-inflated Poisson model.

`alpha` If transform=TRUE, this determines the power to which the data matrices x and xte are transformed. If alpha=NULL then the transformation that makes the zero-inflated Poisson model best fit the data matrix x is computed. (Note that alpha is computed based on x, not based on xte). Or a value of alpha, 0<alpha<=1, can be entered by the user.

`prior` Vector of length equal to the number of classes, representing prior probabilities for each class. If NULL then uniform priors are used (i.e.each class is equally likely).

Value

list(.) A list of output, "errs" represents A matrix of dimension (number of folds)-by-(length of rhos)."bestrho" represents The tuning parameter value resulting in the lowest overall cross-validation error rate for. "rhos" represent the vector of rho values used in cross-validation."nnonzero" represents A matrix of dimension (number of folds)-by-(length of rhos)."folds" represents Cross-validation folds used. "alpha" represents Power transformation used (if transform=TRUE).

Examples

```
library(SummarizedExperiment)
dat <- newCountDataSet(n=40,p=500, K=4, param=10, sdsignal=0.1,drate=0.4)
x <- dat$sim_train_data
y <- as.numeric(colnames(dat$sim_train_data))
xte <- dat$sim_test_data
prob <- estimatep(x=x, y=y, xte=x, beta=1, type="mle", prior=NULL)
prob0 <- estimatep(x=x, y=y, xte=xte, beta=1,type="mle", prior=NULL)
cv.out <- ZIPLDA.cv(x=x, y=y, prob0=t(prob))
out <- ZIPLDA(x=x, y=y, xte=xte, rho=cv.out$bestrho, prob0=t(prob0))
```

Index

∗ **datasets**
    realData, 5

CAEN, 2

estimatep, 3

newCountDataSet, 4

realData, 5

ZIPLDA, 5
ZIPLDA.cv, 7