Package 'countsplit'

August 24, 2023

Title Splitting a Count Matrix into Independent Folds

Version 4.0.0

Description Implements the count splitting methodology from Neufeld et al. (2022) <doi:10.1093/biostatistics/kxac047> and Neufeld et al. (2023) <arXiv:2307.12985>. Intended for turning a matrix of single-cell RNA sequencing counts, or similar count datasets, into independent folds that can be used for training/testing or cross validation. Assumes that the entries in the matrix are from a Poisson or a negative binomial distribution.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.2.3

Imports Matrix, methods, Rcpp

Suggests knitr, rmarkdown

URL https://github.com/anna-neufeld/countsplit

Depends R (>= 2.10)

BugReports https://github.com/anna-neufeld/countsplit/issues

LinkingTo Rcpp

NeedsCompilation yes

Author Anna Neufeld [aut, cre, cph],
Mischko Heming [ctb],
Joshua Popp [ctb]

Maintainer Anna Neufeld <aneufeld@fredhutch.org>

Repository CRAN

Date/Publication 2023-08-24 10:30:09 UTC

R topics documented:

countsplit *Count splitting*

Description

Takes one matrix of counts and splits it into a specified number of folds. Each fold is a matrix of counts with the same dimension as the original matrix. Summing element-wise across the folds yields the original data matrix.

Usage

```
countsplit(X, folds = 2, epsilon = rep(1/folds, folds), overdisps = NULL)
```

Arguments

X	A cell-by-gene matrix of integer counts
folds	An integer specifying how many folds you would like to split your data into.
epsilon	A vector, which has length folds, that stores non-zero elements that sum to one. Determines the proportion of information from X that is allocated to each fold. When folds is not equal to 2, the recommended (and default) setting is to allocate equal amounts of information to each fold, such that each element is 1/folds. When folds=2, the default is still (1/2, 1/2), but other values may be beneficial.
overdisps	If NULL, then Poisson count splitting will be performed. Otherwise, this parameter should be a vector of non-negative numbers whose length is equal to the number of columns of X. These numbers are the overdispersion parameters for each column in X. If these are unknown, they can be estimated with a function such as vst in the package sctransform.

Details

When the argument overdisps is set to NULL, this function performs the Poisson count splitting methodology outlined in Neufeld et al. (2022). With this setting, the folds of data are independent only if the original data were drawn from a Poisson distribution.

If the data are thought to be overdispersed relative to the Poisson, then we may instead model them as coming from a negative binomial distribution, If we assume that $X_{ij} \sim NB(\mu_{ij}, b_j)$, where this parameterization means that $E[X_{ij}] = \mu_{ij}$ and $Var[X_{ij}] = \mu_{ij} + \mu_{ij}^2/b_j$, then we should pass in overdisps = $c(b_1, \ldots, b_j)$. If this is the correct assumption, then the resulting folds of data will be independent. This is the negative binomial count splitting method of Neufeld et al. (2023).

Please see our tutorials and vignettes for more details.

Value

A list of length folds. Each element in the list stores a sparse matrix with the same dimensions as the data X. Each list element is a fold of data.

References

reference

Examples

```
library(countsplit)
library(Matrix)
library(Rcpp)
# A Poisson count splitting example.
n=400
p=2
X <- matrix(rpois(n*p, 7), nrow=n, ncol=p)
split <- countsplit(X, folds=2)
Xtrain <- split[[1]]
Xtest <- split[[2]]
cor(Xtrain[,1], Xtest[,1])
cor(Xtrain[,2], Xtest[,2])

# A negative binomial count splitting example.
X <- matrix(rnbinom(n*p, mu=7, size=7), nrow=n, ncol=p)
split <- countsplit(X, folds=2, overdisps=c(7,7))
Xtrain <- split[[1]]
Xtest <- split[[2]]
cor(Xtrain[,1], Xtest[,1])
cor(Xtrain[,2], Xtest[,2])
```

Index