Package 'npSeq'

September 7, 2011

Title A non-parametric method for significance analysis of sequencing data

Version 1.1

Author Jun Li

Description This package implements all methods used in paper Jun Li and Robert Tibshirani (2011). Finding consistent patterns: a nonparametric approach for identifying differential expression in RNA-Seq data. To appear, Statistical Methods in Medical Research.

Maintainer Jun Li <junli07@stanford.edu>

Depends combinat

License LGPL

R topics documented:

`npSeq.Main` *Discover differentially expressed genes using a nonparametric method*

Description

Discover significant genes and estimate false discovery rates using the method described in Jun Li and Robert Tibshirani (2011).
This is the main (key) function of this package.

Usage

```
npSeq.Main(dat, para=list())
```

Arguments

dat	a list with elements: n: count matrix. row: counts from a gene, column: counts from an experiment. each element should be a non-negative integer. original count matrix (not normalized.) y: outcome vector. twoclass data: '1', '2' for two classes. multiclass data: '1', '2', ..., 'K' for K classes. quantitative data: real numbers. survi: real numbers (survival times). type: "twoclass", "multiclass", "quant", or "survi". gname(optional): the names of the genes. gamma(optional): censoring statuses. '1' for observed (died), '0' for censored. delta(optional): true significance. TRUE for significance. FALSE for insignificance. This can only be known in simulated data. When delta is not null, true false discovery rates will be calculated and returned.
para	a list with elements (all of them are optional): npermu: number of permutations used to estimate FDR. Default value: 100. nsam: number of resamplings. Default value: 20. sam.meth: resampling method: '1' for subsampling, '2' for Poisson sampling. Default value: 2. seed: random seed for resampling. Default value: 20. ct.sum: if the total number of reads of a gene across all experiments <= ct.sum, this gene will not be considered for differential expression detection. Default value: 5. ct.mean: if the mean number of reads of a gene across all experiments <= ct.mean, this gene will not be considered for differential expression detection. Default value: 0.5.

Value

a data frame (table) containing the following columns. Each row stands for a gene. The genes are sorted from the most significant to the most insignificant.

nc	number of significant genes called.
gname	the sorted gene names.
tt	The statistics of the genes.
pval	Permutation-based p-values of the genes.
fdr	Estimated false discovery rate.
log.fc	Estimated log fold change of the genes. Only available for twoclass outcomes.
tfdr	True false discovery rate. Only available when dat$delta is not NULL.

Author(s)

Jun Li

References

Jun Li and Robert Tibshirani (2011). Finding consistent patterns: a nonparametric approach for identifying differential expression in RNA-Seq data. To appear, Statistical Methods in Medical Research.

Jun Li, Daniela M. Witten, Iain Johnstone, Robert Tibshirani (2011). Normalization, testing, and false discovery rate estimation for RNA-sequencing data. To appear, Biostatistics.

Examples

```
## two class negative binomial-distributed data with outliers,
## 12 samples in each class
dat <- npSeq.Simu.Data(list(type='twoclass', NGENE=1000, option=4, NSAM=c(8, 8)))
np.fdr <- npSeq.Main(dat)

## 4 class Poisson-distributed data with outliers,
## 6 samples in each class
dat <- npSeq.Simu.Data(list(type='multiclass', NGENE=1000, option=3, NSAM=c(3, 3, 3, 3)))
np.fdr <- npSeq.Main(dat)

## quantitative negative binomial-distributed data with outliers,
## 24 samples totally
dat <- npSeq.Simu.Data(list(type='quant', NGENE=1000, option=4, NSAM=12))
np.fdr <- npSeq.Main(dat)

## survival negative binomial-distributed data with outliers,
## 24 samples totally
dat <- npSeq.Simu.Data(list(type='survi', NGENE=1000, option=4, NSAM=12))
np.fdr <- npSeq.Main(dat)
```

---

npSeq.Simu.Data *Simulate sequencing data*

Description

Simulate sequencing data with two class, multiclass, quantitative or survival outcomes.

Usage

```
npSeq.Simu.Data(dat, seed=10)
```

Arguments

dat	a list with elements (the first three are required): `type`: "twoclass", "multiclass", "quant", or "survi". `option`: "1" for Poisson, "2" for negative binomial with dispersion 0.25, "3" for Poisson with outliers, "4" for negative binomial with outliers.

NSAM: number of samples. an integer for quant and survi, and a vector of integers for twoclass and multiclass.
NGENE: number of genes. default value 20000.
psig: percentage of significant genes. default value 0.3.
up.perc: in the significant genes, how many percent are up-regulated. Default value: 0.8.

seed — random seed

Details

This function generate all simulated data for the paper. Different outcome type: two class, multiple class, quantitative, or survival. Different distribution: Poisson, negative binomial, with/without outliers.

Value

a list with all elements in the input dat, and

rmean	gene expression levels.
cmean	sequencing depths.
mu	means of Poission/negative binomial distribution.
y	the outcome vector.
fold.change	the log fold change.
n	the count matrix
delta	TRUE/FALSE indicating whether a gene is differentially expressed.
gamma	for survival data. observed (1) or censored (0).

Author(s)

Jun Li

References

Jun Li and Robert Tibshirani (2011). Finding consistent patterns: a nonparametric approach for identifying differential expression in RNA-Seq data. To appear, Statistical Methods in Medical Research.

Jun Li, Daniela M. Witten, Iain Johnstone, Robert Tibshirani (2011). Normalization, testing, and false discovery rate estimation for RNA-sequencing data. To appear, Biostatistics.

Examples

```
## two class Poisson-distributed data with 12 samples in each class
dat <- npSeq.Simu.Data(list(type='twoclass', option=1, NSAM=c(12, 12)))

## two class negative binomial-distributed data with outliers,
```

```
## 12 samples in each class
dat <- npSeq.Simu.Data(list(type='twoclass', option=4, NSAM=c(12, 12)))

## 4 class Poisson-distributed data with outliers,
## 6 samples in each class
dat <- npSeq.Simu.Data(list(type='multiclass', option=3, NSAM=c(6, 6, 6, 6)))

## quantitative negative binomial-distributed data with outliers,
## 24 samples totally
dat <- npSeq.Simu.Data(list(type='quant', option=4, NSAM=24))

## survival negative binomial-distributed data with outliers,
## 24 samples totally
dat <- npSeq.Simu.Data(list(type='survi', option=4, NSAM=24))
```

Index