Package 'DArand'

October 12, 2022

Title Differential Analysis with Random Reference Genes

Version 0.0.1.2

Description Differential Analysis of short RNA transcripts that can be modeled by either Poisson or Negative binomial distribution. The statistical methodology implemented in this package is based on the random selection of references genes (Desaulle et al. (2021) <arXiv:2103.09872>).

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.1.2

Suggests testthat (>= 3.0.0)

Config/testthat/edition 3

Imports parallel

NeedsCompilation no

Author Dorota Desaulle [aut, cre] (<https://orcid.org/0000-0002-3419-9447>), Yves Rozenholc [aut] (<https://orcid.org/0000-0002-3907-5101>)

Maintainer Dorota Desaulle <dorota.desaulle@u-paris.fr>

Repository CRAN

Date/Publication 2022-02-10 15:40:02 UTC

R topics documented:

`build_example` *Simulation of gene expressions using independant negative binomials*

Description

Simulation of gene expressions using independant negative binomials

Usage

```
build_example(
  m = 500,
  m1,
  n1 = 6,
  n2 = n1,
  fold = 100,
  mu0 = 100,
  use.scales = FALSE,
  nb.size = Inf
)
```

Arguments

`m,`	number of genes
`m1,`	number of differentially expressed genes. In the expression matrix, m1 first columns contain differentially expressed genes.
`n1,`	number of samples under the first condition. The first n1 rows in the expression matrix.
`n2,`	number of samples under the second condition (default n2=n1)
`fold,`	maximal fold change added to the first m1 genes. The fold decreases proportionally to `1/sqrt(1:m1)`.
`mu0,`	mean relative expression
`use.scales,`	if TRUE random scales are used, otherwise all scales are set to 1.
`nb.size,`	number of successful trials in the negative binomial distribution. If nb.size is set to Inf (default), the Poisson model is used.

Details

The function generates a list, of which the first element `X` is a matrix of `n1+n2` and m dimension with simulated expressions under Poisson or Negative Binomial distribution. Lines `1:n1` correspond to the first condition (or sub-group) and lines `(n1+1):(n1+n2)` to the second one. Columns `1:m1` contain counts imitating differential expressions.

In the ideal situation there is no microscopical variability between samples and all scales (so-called scaling factors) would be the same. To simulate examples corresponding to this perfect situation, use argument `use.scales=FALSE` which will set all scales to 1. When `use.scales=TRUE`, scales are simulated under uniform distribution *Unif(0.25,4)*.

The fold is maximal for the first expression and decreases proportionally to `1/sqrt(1:m1)`. The smallest fold `fold/sqrt(m1)` is set to the *m1*-th expression.

Value

A list with components

`X` a two-dimensional array containing the expression table of n individuals in rows and m gene expressions in columns.

`m1` number of differentially expressed genes (as in arguments).

`n1` number of samples under the first condition (as in arguments).

`n2` number of samples under the second condition (as in arguments).

`fold` maximal fold change between the differentally expressed genes and invariant genes (as in arguments).

`scales` vector of simulated scales.

`mu0` mean relative expression (as in arguments).

Examples

```
L = build_example(m=500,m1=25,n1=6,fold=20,mu0=100,use.scales=FALSE,nb.size=Inf)
```

`DArand` *Do Differential Analysis with Random Reference Genes*

Description

Implement the `DArand` procedure for transcriptomic data. The procedure is based on random and repeated selection of subsets of reference genes as described in the paper cited below. Observed counts data are normalized with counts from the subset and a differential analysis is used to detect differentially expressed genes. Thought repetitions, the number times a gene is detected is recorded and the final selection is determined from p-values computed under Binomial distribution and adjusted with the Holm's correction.

Usage

```
DArand(
  X,
  n1,
  k = NULL,
  alpha = 0.05,
  eta = 0.05,
  beta = 0.1,
  r = 1000,
  with.info = FALSE,
  clog = 1,
```

```
  use.multi.core = TRUE,
  step = 0,
  scales = NULL,
  use.Iter = TRUE,
  set.seed = NULL
)
```

Arguments

X	a two-dimensional array (or data.frame) containing the expression table of n individuals in rows and m gene expressions in columns.
n1	integer, number of individuals of the first category, should be smaller than n
k	integer, number of random genes selected (default k = ceiling(log2(m))) as reference genes.
alpha	numeric, global test level (default 0.05)
eta	numeric, inner test level (default 0.05)
beta	numeric, inner type II error (default 0.1)
r	integer, number of random 'reference' set selected (the default 1000)
with.info	logical, if TRUE results are displayed (the default FALSE)
clog	numeric, constant (default 1) controlling the gaussian approximation of the test statistic (in Negative Binomial and Poisson case) .
use.multi.core	logical, if TRUE (the default) parallel computing with mclapply is used.
step	integer, only used when use.Iter is TRUE to get information on the number of iterations (default 0). Not for use.
scales	numeric, only used for simulation of oracle purpose (default NULL). Not for use.
use.Iter	logical, applies iterative procedure (default FALSE)
set.seed	numeric, set random seed (as is in set.seed function for random number generation), here default is NULL.

Details

The expression table should be organized in such a way that individuals are represented in rows and genes in columns of the X array. Furthermore, in the current version, the procedure provides a differential analysis comparing exactly two experimental conditions. Hence, lines from 1 to n1 should correspond to the first condition and the remaining lines to the second condition.

In the inner part of the procedure, called further *randomization*, scaling factors are estimated using a normalization subset of k genes randomly selected from all m genes. These k genes are used as reference genes. The normalized data are compared between the experimental conditions within an approximately gaussian test for Poisson or negative-binomial counts as proposed in the methodology cited below. For this inner test the type I (eta) and the type II (beta) errors should be specified, otherwise the default values will be used. Since true reference genes (*housekeeping genes*) are unknown, the inner part is repeated r times.

Through all r randomization, for each gene, the number of detections (*i.e.* the number of randomizations when a given gene is identified as differentially expressed) is collected. For these detection

counts, the corresponding p-values are computed under the Binomial distribution. The finale detection uses the p-values and, owing to Holm's correction, controls FWER at specified level `alpha`.

The maximal number of discoveries is limited to Delta - the parameter that is a function of `eta`, `beta` and the probability of selecting a subset containing at least one differentially expressed gene leading to a wrong normalization (see `select_prob`) . If `use.Iter` is TRUE (the default), the maximal number of discoveries is limited (per iteration) to Delta. The procedure is iterated as long as the number of discoveries is equal to the value of Delta computed in the iteration. Starting from step=1, at each iteration the one-type error is halved `alpha=alpha/2` to ensure the overall test level respects the initial alpha.

`clog` is a constant that controls gaussian approximation of the test statistic for the count data arising from Negative Binomial or Poisson distribution. The constant should be ajusted to keep the probability `1-5*n^(-clog)` high while shift term `1+sqrt(clog*n)` low.

Value

position vector of the gene expressions found as differentially expressed.

Author(s)

D. Desaulle and Y. Rozenholc

References

Differential analysis in Transcriptomic: The strengh of randomly picking 'reference' genes, D. Desaulle, C. Hoffman, B. Hainque and Y. Rozenholc. https://arxiv.org/abs/2103.09872

Examples

```
L = build_example(m=500,m1=25,n1=6,fold=20,mu0=100,use.scales=FALSE,nb.size=Inf)
DArand(L$X,L$n1,alpha=0.05)
```

---

select_prob *Probabilities to select a normalization set without DE-gene*

Description

Probabilities to select a normalization set without DE-gene

Usage

```
select_prob(m, k, invariant = TRUE)
```

Arguments

`m,`	number of genes
`k,`	normalization subset size
`invariant,`	boolean, when TRUE, probability of selection is evaluated for invariant gene

Value

a vector of probabilities of having at least one differential expression used as an reference selected in the normalization subset for any number of differential expressions d in the gene collection.

Examples

```
select_prob(500, 10, invariant=TRUE)
```

Index