Package 'PAIRADISE'

May 6, 2024

Title PAIRADISE: Paired analysis of differential isoform expression

Version 1.20.0

Author Levon Demirdjian, Ying Nian Wu, Yi Xing

Maintainer Qiang Hu <Qiang.Hu@roswellpark.org>, Levon Demirdjian <levondem@ucla.edu>

Description This package implements the PAIRADISE procedure for detecting differential isoform expression between matched replicates in paired RNA-Seq data.

Depends R (>= 3.6), nloptr

Imports SummarizedExperiment, S4Vectors, stats, methods, abind, BiocParallel

License MIT + file LICENSE

RoxygenNote 7.1.1

Encoding UTF-8

VignetteBuilder knitr

Suggests testthat, knitr, rmarkdown, BiocStyle

biocViews RNASeq, DifferentialExpression, AlternativeSplicing, StatisticalMethod, ImmunoOncology

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

git_branch RELEASE_3_19

git_last_commit 6367989

git_last_commit_date 2024-04-30

Repository Bioconductor 3.19

Date/Publication 2024-05-05

Contents

`clean.data` *clean.data*

Description

Removes missing data and invalid pairs from the matched pair data to be analyzed by PAIRADISE.

Usage

```
clean.data(my.data)
```

Arguments

`my.data`	Data frame containing grouped data to be analyzed.

Details

The data frame has 7 columns, arranged as follows: Column 1 contains the ID of the exons/events. Column 2 contains counts of isoform 1 corresponding to the first group. Column 3 contains counts of isoform 2 corresponding to the first group. Column 4 contains counts of isoform 1 corresponding to the second group. Column 5 contains counts of isoform 2 corresponding to the second group. Replicates in columns 2-5 should be separated by commas, e.g. 1623,432,6 for three replicates. Column 6 contains the effective length of isoform 1. Column 7 contains the effective length of isoform 2.

Value

The function clean.data returns a list containing the following entries:

`I1`	Group 1 isoform 1 counts for each replicate.
`S1`	Group 1 isoform 2 counts for each replicate.
`I2`	Group 2 isoform 1 counts for each replicate.
`S2`	Group 2 isoform 2 counts for each replicate.
`length_I`	Effective lengths of isoform 1.

length_S	Effective lengths of isoform 2.
exonList	IDs of the exons/events.
nExon	Number of exons/events.
M	Vector containing the number of replicates per exon/event.

---

counts *PDseDataSet counts*

Description

PDseDataSet counts

Usage

```
counts(object)
```

Arguments

object	A PDseDataSet object

Value

A counts matrix

---

load.data *load.data*

Description

Loads the matched pair data to be analyzed by PAIRADISE.

Usage

```
load.data(my.data)
```

Arguments

my.data	Data frame containing grouped data to be analyzed.

Details

The data frame has 7 columns, arranged as follows: Column 1 contains the ID of the exons/events. Column 2 contains counts of isoform 1 corresponding to the first group. Column 3 contains counts of isoform 2 corresponding to the first group. Column 4 contains counts of isoform 1 corresponding to the second group. Column 5 contains counts of isoform 2 corresponding to the second group. Replicates in columns 2-5 should be separated by commas, e.g. 1623,432,6 for three replicates. Column 6 contains the effective length of isoform 1. Column 7 contains the effective length of isoform 2.

Value

The function load.data returns a list containing the following entries:

I1	Group 1 isoform 1 counts for each replicate.
S1	Group 1 isoform 2 counts for each replicate.
I2	Group 2 isoform 1 counts for each replicate.
S2	Group 2 isoform 2 counts for each replicate.
length_I	Effective lengths of isoform 1.
length_S	Effective lengths of isoform 2.
exonList	IDs of the exons/events.
nExon	Number of exons/events.
M	Vector containing the number of replicates per exon/event.

---

logit *logit*

Description

Takes in a vector and applies the logit function elementwise to that vector

Usage

```
logit(x)
```

Arguments

x	: numeric vector, whose entries should be strictly between 0 and 1

Value

logit(x)

loglikelihood *loglikelihood*

Description

Used internally in PAIRADISE to compute the log-likelihood function

Usage

```
loglikelihood(
  M,
  I1,
  S1,
  I2,
  S2,
  l.iI,
  l.iS,
  logit.psi1,
  logit.psi2,
  alpha,
  s1,
  s2,
  s,
  mu,
  delta
)
```

Arguments

M	Number of replicates for the current exon. Positive integer.
I1	Exon inclusion counts for group 1. Positive integers.
S1	Exon skipping counts for group 1. Positive integers.
I2	Exon inclusion counts for group 2. Positive integers.
S2	Exon skipping counts for group 2. Positive integers.
l.iI	Effective length of inclusion isoform. Positive integer.
l.iS	Effective length of skipping isoform. Positive integer.
logit.psi1	Numeric vector with values of logit psi1.
logit.psi2	Numeric vector with values of logit psi2.
alpha	Numeric vector with values of alpha.
s1	Group 1 standard deviation. Positive number.
s2	Group 2 standard deviation. Positive number.
s	Overall standard deviation. Positive number.
mu	Parameter mu.
delta	Parameter delta.

Value

log likelihood value at input.

`optimize1` *optimize1*

Description

Used internally in PAIRADISE to compute the MLEs of delta, mu, sigma1, sigma2, sigma

Usage

```
optimize1(
  x,
  M,
  I1,
  S1,
  I2,
  S2,
  l.iI,
  l.iS,
  logit.psi1,
  logit.psi2,
  alpha,
  equal.variance = FALSE
)
```

Arguments

`x`	Numeric vector such that x = (sigma1, sigma2, sigma, mu, delta) if equal.variance = FALSE, and x = (sigma1, sigma, mu, delta) if equal.variance = TRUE. sigma1, sigma2, sigma must be positive
`M`	Number of replicates for the current exon.
`I1`	Exon inclusion counts for group 1. Positive integers.
`S1`	Exon skipping counts for group 1. Positive integers.
`I2`	Exon inclusion counts for group 2. Positive integers.
`S2`	Exon skipping counts for group 2. Positive integers.
`l.iI`	Effective length of inclusion isoform. Positive integer.
`l.iS`	Effective length of skipping isoform. Positive integer.
`logit.psi1`	Numeric vector with values of logit psi1.
`logit.psi2`	Numeric vector with values of logit psi2.
`alpha`	Numeric vector with values of alpha.
`equal.variance`	Are the group variances assumed equal? Default value is FALSE.

Value

The MLEs.

`optimize2` *optimize2*

Description

Used internally in PAIRADISE to compute the MLEs of logit(psi1), logit(psi2), alpha

Usage

```
optimize2(x, k, I1, S1, I2, S2, l.iI, l.iS, delta, mu, s1, s2, s)
```

Arguments

`x`	Numeric vector such that x = (logit(psi1), logit(psi2), alpha)
`k`	Index representing current replicate number.
`I1`	Exon inclusion counts for group 1. Positive integers.
`S1`	Exon skipping counts for group 1. Positive integers.
`I2`	Exon inclusion counts for group 2. Positive integers.
`S2`	Exon skipping counts for group 2. Positive integers.
`l.iI`	Effective length of inclusion isoform. Positive integer.
`l.iS`	Effective length of skipping isoform. Positive integer.
`delta`	Parameter delta.
`mu`	Parameter mu.
`s1`	Group 1 standard deviation. Positive number.
`s2`	Group 2 standard deviation. Positive number.
`s`	Overall standard deviation. Positive number.

Value

The MLEs.

PAIRADISE *PAIRADISE Detecting allele-specific alternative splicing from population-scale RNA-seq data*

Description

We introduce PAIRADISE (PAIred Replicate analysis of Allelic DIfferential Splicing Events), a method for detecting allele-specific alternative splicing (ASAS) from RNA-seq data. PAIRADISE uses a statistical model that aggregates ASAS signals across multiple individuals in a population. It formulates ASAS detection as a statistical problem for identifying differential alternative splicing from RNA-seq data with paired replicates. The PAIRADISE statistical model is applicable to many forms of allele-specific isoform variation (e.g. RNA editing), and can be used as a generic statistical model for RNA-seq studies involving paired replicates.

See Also

pairadise

pairadise *pairadise*

Description

Primary function of the PAIRADISE package. Analyzes matched pairs for differences in isoform expression. Uses parallel processing to speed up computation.

Usage

```
pairadise(
  pdat,
  nIter = 100,
  tol = 10^(-2),
  pseudocount = 0,
  seed = 12321,
  equal.variance = FALSE,
  numCluster = 2,
  BPPARAM = MulticoreParam(numCluster)
)
```

Arguments

pdat	A PDseDataSet object
nIter	Positive integer. Specifies the maximum number of iterations of the optimization algorithm allowed. Default is nIter = 100

tol	Positive number. Specifies the tolerance level for terminating the optimization algorithm, defined as the difference in log-likelihood ratios between iterations. Default is tol = 10^(-2)
pseudocount	Positive number. Specifies a value for a pseudocount added to each count at the beginning of the analysis. Default is pseudocount = 0
seed	An integer to set seed.
equal.variance	Are the group variances assumed equal? Default value is FALSE.
numCluster	Number of clusters to use for parallel computing.
BPPARAM	parallel parameters from package BiocParallel.

Details

This is the primary function of the PAIRADISE package that implements the PAIRADISE algorithm.

Value

A PDseDataSet object contains outputs from PAIRADISE algorithm.

Examples

```
############################
## Example: Simulated data ##
############################

set.seed(12345)
data("sample_dataset")
pdat <- PDseDataSetFromMat(sample_dataset)
pdat <- pairadise(pdat, numCluster =4)
results(pdat)
```

PDseDataSet-class *PDseDataSet object and constuctor*

Description

'PDseDataSet' is a subclass of 'SummarizedExperiment'. It can used to store inclusion and skipping splicing counts for pair designed samples.

Usage

```
PDseDataSet(counts, design, lengths)
```

Arguments

counts	The counts of splicing events, including inclusion and skipping counts in 3 dimensions for each sample.
design	The paired design data.frame, including sample column for sample ids and group column for design factors.
lengths	Two columns iLen and sLen for the effective lengths of inclusion and skipping isoforms.

Value

A PDseDataSet object

Examples

```
icount <- matrix(1:4, 1)
scount <- matrix(5:8, 1)
acount <- abind::abind(icount, scount, along = 3)
design <- data.frame(sample = rep(c("s1", "s2"), 2),
group = rep(c("T", "N"), each = 2))
lens <- data.frame(sLen=1L, iLen=2L)
PDseDataSet(acount, design, lens)
```

PDseDataSetFromMat *PDseDataSet from rMATs/PAIRADISE Mat format*

Description

The Mat format should have 7 columns, arranged as follows: Column 1 contains the ID of the alternative splicing events. Column 2 contains counts of isoform 1 corresponding to the first group. Column 3 contains counts of isoform 2 corresponding to the first group. Column 4 contains counts of isoform 1 corresponding to the second group. Column 5 contains counts of isoform 2 corresponding to the second group. Column 6 contains the effective length of isoform 1. Column 7 contains the effective length of isoform 2. Replicates in columns 2-5 should be separated by commas, e.g. "1623,432,6" for three replicates and the replicate order should be consistent for each column to ensure pairs are matched correctly.

Usage

```
PDseDataSetFromMat(dat)
```

Arguments

dat	The Mat format dataframe.

Value

A PDseDataSet object

Examples

```
data("sample_dataset")
pdat <- PDseDataSetFromMat(sample_dataset)
```

`results` *Extract results for pairadise analysis*

Description

Extract results for pairadise analysis

Usage

```
results(pdat, p.adj = "BH", sig.level = 0.01, details = FALSE)
```

Arguments

`pdat`	A PDseDataSet object from pairadise analysis
`p.adj`	The p ajustment method.
`sig.level`	The cutoff of significant results
`details`	Whether to list detailed results.

Value

The function return a results DataFrame.

`testStats`	Vector of test statistics for paired analysis.
`p.value`	Vector of pvalues for each exon/event.
`p.adj`	The adjusted p values

If details is TRUE, more detailed parameter estimates for constrained and unconstrained model will return.

Examples

```
data("sample_dataset")
pdat <- PDseDataSetFromMat(sample_dataset)
pdat <- pairadise(pdat)
results(pdat)
```

sample_dataset *sample_dataset*

Description

The CEU dataset was generated by analyzing the allele-specific alternative splicing events in the GEUVADIS CEU data. Allele-specific reads were mapped onto alternative splicing events using rPGA (version 2.0.0). Then the allele-specific bam files mapped onto the two haplotypes are merged together to detect alternative splicing events using rMATS (version 3.2.5)16.

The LUSC dataset was generated by analyzing the tumor versus adjacent control samples from TCGA LUSC RNA-seq data.

Usage

```
data(sample_dataset)

data(sample_dataset_CEU)

data(sample_dataset_LUSC)
```

Format

The dataset has 7 columns, arranged as follows:

ExonID Column 1 contains the ID of the alternative splicing events.

I1 Column 2 contains counts of isoform 1 corresponding to the first group.

S1 Column 3 contains counts of isoform 2 corresponding to the first group.

I2 Column 4 contains counts of isoform 1 corresponding to the second group.

S2 Column 5 contains counts of isoform 2 corresponding to the second group.

I_len Column 6 contains the effective length of isoform 1.

S_len Column 7 contains the effective length of isoform 2.

The dataset has 7 columns, arranged as follows:

ExonID Column 1 contains the ID of the alternative splicing events.

I1 Column 2 contains counts of isoform 1 corresponding to the first group.

S1 Column 3 contains counts of isoform 2 corresponding to the first group.

I2 Column 4 contains counts of isoform 1 corresponding to the second group.

S2 Column 5 contains counts of isoform 2 corresponding to the second group.

I_len Column 6 contains the effective length of isoform 1.

S_len Column 7 contains the effective length of isoform 2.

The dataset has 7 columns, arranged as follows:

ExonID Column 1 contains the ID of the alternative splicing events.

I1 Column 2 contains counts of isoform 1 corresponding to the first group.

S1 Column 3 contains counts of isoform 2 corresponding to the first group.

I2 Column 4 contains counts of isoform 1 corresponding to the second group.

S2 Column 5 contains counts of isoform 2 corresponding to the second group.

I_len Column 6 contains the effective length of isoform 1.

S_len Column 7 contains the effective length of isoform 2.

`sigmoid` *sigmoid*

Description

Takes in a vector and applies the sigmoid function elementwise to that vector

Usage

```
sigmoid(x)
```

Arguments

`x` : numeric vector

Value

sigmoid(x)

Index