Package ‘mpra’

May 4, 2024

Version 1.26.0
Title Analyze massively parallel reporter assays
Description Tools for data management, count preprocessing, and differential analysis in massively parallel report assays (MPRA).
Depends R (>= 3.4.0), methods, BiocGenerics, SummarizedExperiment, limma
Suggests BiocStyle, knitr, markdown, RUnit
Imports S4Vectors, scales, stats, graphics, statmod
VignetteBuilder knitr
License Artistic-2.0
URL https://github.com/hansenlab/mpra
BugReports https://github.com/hansenlab/mpra/issues
biocViews Software, GeneRegulation, Sequencing, FunctionalGenomics
git_url https://git.bioconductor.org/packages/mpra
git_branch RELEASE_3_19
git_last_commit e361e46
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-03
Author Leslie Myint [cre, aut], Kasper D. Hansen [aut]
Maintainer Leslie Myint <leslie.myint@gmail.com>

Contents

mpra-package .................................................. 2
compute_logratio ............................................. 3
get_precision_weights ...................................... 3
Description

Tools for data management, count preprocessing, and differential analysis in massively parallel report assays (MPRA).

Details

This package provides tools for the analysis of MPRA data. The primary purpose is to enable powerful differential analysis of activity measures, but the package can also be used to generate precision weights useful in regression analyses of activity scores on sequence features. The main workhorse is the `mpralm` function which draws on the previously proposed voom framework for RNA-seq analysis in the `limma` package.

Author(s)

Leslie Myint [cre, aut], Kasper D. Hansen [aut]
Maintainer: Leslie Myint <leslie.myint@gmail.com>

References


Examples

data(mpraSetAggExample)
design <- data.frame(intcpt = 1,
            episomal = grepl("MT", colnames(mpraSetAggExample)))
mpralm_fit <- mpralm(object = mpraSetAggExample, design = design,
                  aggregate = "none", normalize = TRUE,
                  model_type = "indep_groups", plot = FALSE)
```r
toptab <- topTable(mpralm_fit, coef = 2, number = Inf)
head(toptab)
```

**compute_logratio**  
*Compute activity measure (log-ratio) for each element.*

**Description**

Compute the log ratio of RNA counts to DNA counts using different methods. For "mean", uses the average of barcode-specific log ratios. For "sum", sums RNA and DNA counts over barcodes before forming the log ratio.

**Usage**

```r
compute_logratio(object, aggregate = c("mean", "sum", "none"))
```

**Arguments**

- **object**: An object of class `MPRASet`.
- **aggregate**: Aggregation method over barcodes: "mean" to use the average of barcode-specific log ratios, "sum" to use the log ratio of summed RNA and DNA counts, "none" to perform no aggregation (counts have already been summarized over barcodes).

**Value**

A matrix with the same dimension as `object`, containing element- and sample-specific log ratios.

**Examples**

```r
data(mpraSetAggExample)
logr <- compute_logratio(mpraSetAggExample, aggregate = "sum")
```

---

**get_precision_weights**  
*Get precision weights from the copy number-variance relationship.*

**Description**

Estimates the variability of the supplied log-ratios across samples as a function of copy number (DNA count levels).

**Usage**

```r
get_precision_weights(logr, design, log_dna, span = 0.4, plot = TRUE, ...)
```
Arguments

- **logr**: Matrix of outcome measures: log2 ratio of RNA counts to DNA counts.
- **design**: Design matrix specifying comparisons of interest.
- **log_dna**: Matrix of log2 aggregated DNA counts of the same dimension as `logr`.
- **span**: The smoothing span for `lowess` in estimating the copy number-variance relationship. Default: 0.4.
- **plot**: If TRUE, plot the copy number-variance relationship.
- **...**: Further arguments to be passed to `lmFit` for obtaining residual standard deviations used in estimating the copy number-variance relationship.

Details

Residual standard deviations are computed using the supplied outcomes and design matrix. The square root of the residual standard deviations are modeled as a function of the average log2 aggregated DNA counts to estimate the copy number-variance relationship.

Value

A matrix of precision weights of the same dimension as `logr` and `log_dna`.

References


Examples

```r
data(mpraSetAggExample)
design <- data.frame(intcpt = 1,
                     episomal = grepl("MT", colnames(mpraSetAggExample)))
logr <- compute_logratio(mpraSetAggExample, aggregate = "none")
log_dna <- log2(getDNA(mpraSetAggExample, aggregate = FALSE) + 1)
w <- get_precision_weights(logr = logr, design = design,
                          log_dna = log_dna, plot = FALSE)
```

**Description**

Fits weighted linear models to test for differential activity in MPRA data.

**Usage**

```r
mpralm(object, design, aggregate = c("mean", "sum", "none"), normalize = TRUE,
       block = NULL, model_type = c("indep_groups", "corr_groups"),
       plot = TRUE, ...)
```
**Arguments**

- **object**: An object of class `MPRASet`.
- **design**: Design matrix specifying comparisons of interest. The number of rows of this matrix should equal the number of columns in `object`. The number of columns in this design matrix has no constraints and should correspond to the experimental design.
- **aggregate**: Aggregation method over barcodes: "mean" to use the average of barcode-specific log ratios, "sum" to use the log ratio of summed RNA and DNA counts, "none" to perform no aggregation (counts have already been summarized over barcodes).
- **normalize**: If TRUE, perform total count normalization before model fitting.
- **block**: A vector giving the sample designations of the columns of `object`. The default, `NULL`, indicates that all columns are separate samples.
- **model_type**: Indicates whether an unpaired model fit ("indep_groups") or a paired mixed-model fit ("corr_groups") should be used.
- **plot**: If TRUE, plot the mean-variance relationship.
- **...**: Further arguments to be passed to `lmFit` for obtaining residual standard deviations used in estimating the mean-variance relationship.

**Details**

Using `method_type = "corr_groups"` use the `duplicateCorrelation` function from the `limma` package to estimate the intra-replicate correlation of log-ratio values.

**Value**

An object of class `MArrayLM` resulting from the `eBayes` function.

**References**


**Examples**

```r
data(mpraSetAggExample)
design <- data.frame(intcpt = 1,
                     episomal = grepl("MT", colnames(mpraSetAggExample)))
mpralm_fit <- mpralm(object = mpraSetAggExample, design = design,
```
aggregate = "none", normalize = TRUE,
model_type = "indep_groups", plot = FALSE)
toptab <- topTable(mpralm_fit, coef = 2, number = Inf)
head(toptab)

---

### Description

A container for data from massively parallel reporter assays (MPRA). Builds on the SummarizedExperiment class.

### Usage

`## Constructor
MPRASet(DNA = new("matrix"), RNA = new("matrix"),
        barcode = new("character"), eid = new("character"),
        eseq = new("character"), ...)

## Accessors
getRNA(object, aggregate = FALSE)
getDNA(object, aggregate = FALSE)
getBarcode(object)
getEid(object)
getEseq(object)

### Arguments

- `object`: A MPRASet object.
- `aggregate`: A logical indicating if data should be aggregated to the element level (by summing across barcodes).
- `DNA`: A matrix of DNA counts where rows correspond to elements or individual barcodes and columns to samples of conditions being compared.
- `RNA`: A matrix of RNA counts where rows correspond to elements or individual barcodes and columns to samples of conditions being compared.
- `barcode`: If barcodes are supplied, a character vector of length equal to the number of rows in DNA and RNA containing the barcode sequences or identifiers. NULL otherwise.
- `eid`: A character vector of length equal to the number of rows in DNA and RNA containing the enhancer identifiers corresponding to each row.
- `eseq`: If supplied, a character vector of length equal to the number of rows in DNA and RNA containing the enhancer sequences corresponding to the regulatory elements in each row. NULL otherwise.
- `...`: Further arguments to be passed to SummarizedExperiment.
mpraSetExample

Value

The constructor function MPRASet returns an object of class "MPRASet".

Slots

Slots are as described in a SummarizedExperiment. Of particular interest are colData which describes the phenotype data. The assay slot holds the assay data, with specific assay names RNA and DNA (accessed by getRNA and getDNA). Element and barcode data are accessible in the rowData slot. We have chosen to store barcode and element as character to be flexible, although they are often DNA sequences (and could therefore be considered DNAStringSet (from package Biostrings)).

Extends

Class "SummarizedExperiment", directly.

Accessors

getDNA: Gets the DNA channel data.
getRNA: Gets the RNA channel data.
getBarcode: Gets the barcode, if present.
getEid: Gets the element ID
getEseq: Gets the element sequence, if present.

See Also

SummarizedExperiment for the basic class that is used as a building block.

Examples

showClass("MPRASet")

Description

Example data for the MPRA package. mpraSetExample and mpraSetAggExample come from a study by Inoue et al that compares episomal and lentiviral MPRA. The former contains data at the barcode level and the latter contains data aggregated over barcodes. mpraSetAllelicExample come from a study by Tewhey et al that looks at regulatory activity of allelic versions of thousands of SNPs to follow up on prior eQTL results.

Usage

data("mpraSetExample")
data("mpraSetAggExample")
data("mpraSetAllelicExample")
**normalize_counts**

**Format**

An MPRASet.

**Details**

`mpraSetExample` contains barcode level information for the study by Inoue et al. `mpraSetAggExample` contains count information from `mpraSetExample` where the counts have been summed over barcodes for each element. `mpraSetAllelicExample` contains count information for the Tewhey et al study. The counts have been summed over barcodes for each element.

**Source**

A script for creating the three datasets is supplied in the `scripts` folder of the package. The data are taken from the GEO submission associated with the paper (see references), specifically GSE83894 and GSE75661.

**References**


**Examples**

```r
data(mpraSetAggExample)
```

<table>
<thead>
<tr>
<th>normalize_counts</th>
<th>Total count normalization of DNA and RNA counts</th>
</tr>
</thead>
</table>

**Description**

Total count normalization of DNA and RNA counts.

**Usage**

```r
normalize_counts(object, block = NULL)
```

**Arguments**

- `object`  
  An object of class MPRASet.
- `block`  
  A vector giving the sample designations of the columns of `object`. The default, `NULL`, indicates that all columns are separate samples.
Details

block is a vector that is used when the columns of the MPRAset object are paired. This often is the case when comparing allelic versions of an element. In this case, the first \( s \) columns of \texttt{object} give the counts for the reference allele in \( s \) samples. The second \( s \) columns give the counts for the alternative allele measured in the same \( s \) samples. With 3 samples, \texttt{block} would be \( c(1,2,3,1,2,3) \). All columns are scaled to have 10 million counts.

Value

An object of class \texttt{MPRASet} with the total count-normalized DNA and RNA counts.

Examples

data(mpraSetAggExample)
mpraSetAggExample <- normalize_counts(mpraSetAggExample)
Index

* classes
  MPRASet-class, 6
* datasets
  mpraSetExample, 7
* package
  mpra-package, 2

compute_logratio, 3
get_precision_weights, 3
getBarcode (MPRASet-class), 6
getDNA (MPRASet-class), 6
getEid (MPRASet-class), 6
getEseq (MPRASet-class), 6
getRNA (MPRASet-class), 6

mpra (mpra-package), 2
mpra-package, 2
mpralm, 4
MPRASet (MPRASet-class), 6
MPRASet-class, 6
mpraSetAggExample (mpraSetExample), 7
mpraSetAllelicExample (mpraSetExample), 7
mpraSetExample, 7

normalize_counts, 8

show,MPRASet-method (MPRASet-class), 6
SummarizedExperiment, 7