Package ‘subSeq’

January 24, 2026

Type Package

Title Subsampling of high-throughput sequencing count data

Version 1.41.0

Author David Robinson, John D. Storey, with contributions from Andrew J. Bass

Maintainer Andrew J. Bass <ajbass@princeton.edu>, John D. Storey <jstorey@princeton.edu>

biocViews ImmunoOncology, Sequencing, Transcription, RNASeq, GeneExpression, DifferentialExpression

Description Subsampling of high throughput sequencing count data for use in experiment design and analysis.

VignetteBuilder knitr

Imports data.table, dplyr, tidyr, ggplot2, magrittr, qvalue (>= 1.99), digest, Biobase

Suggests limma, edgeR, DESeq2, DEXSeq (>= 1.9.7), testthat, knitr

Depends R (>= 3.2)

URL http://github.com/StoreyLab/subSeq

License MIT + file LICENSE

NeedsCompilation no

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

git_branch devel

git_last_commit 133fac9

git_last_commit_date 2025-10-29

Repository Bioconductor 3.23

Date/Publication 2026-01-23

Contents

`combineSubsamples` *combine multiple subsamples objects*

Description

Given two or more subsamples objects, combine them into one larger object, on which we can perform all the usual analyses and plots.

Usage

```
combineSubsamples(...)
```

Arguments

`...`	Two or more subsamples objects

Details

If there are columns in some subsamples objects that are not in others, the missing values will be filled with NA

Value

subSeq object

Examples

```
# see ?subsample to see how ss is generated
data(ss)

# combine multiple subsampling objects (in this example they happen to be the same object)
ss_new <- combineSubsamples(ss, ss)
```

`generateSubsampledMatrix` *Generate the read matrix corresponding to a particular level*

Description

Generate a subsampled matrix from an original count matrix. This can be used to perform read subsampling analyses, (though generally the `subsample` function is recommended).

It is also useful for reproducing the results of an earlier run (see Details).

Usage

```
generateSubsampledMatrix(counts, proportion, seed, replication = 1)
```

Arguments

`counts`	Original matrix of read counts
`proportion`	The specific proportion to subsample
`seed`	A subsampling seed, which can be extracted from a subsamples or summary.subsamples object. If not given, doesn't set the seed.
`replication`	Replicate number: allows performing multiple deterministic replications at a given subsampling proportion

Details

A subsamples object, or a summary.subsamples object, does not contain the subsampled count matrix at each depth (as it would take too much space and is rarely used). However, as it saves the random seed used to generate the count matrix, the count matrix at any depth can be retrieved. This can be done for a subsamples object `ss` by retrieving the seed with `getSeed(ss)`. When given along with the original counts, the proportion, and the replication number (if more than one subsampling was done at each proportion) this produces the same matrix as was used in the analysis.

The seed is calculated deterministically using an md5 hash of three combined values: the global seed used for the subsampling object, the subsampling proportion, and the replication # for that proportion.

Value

subsamples matrix at specified subsampling proportion

Examples

```
data(hammer)

hammer.counts = Biobase::exprs(hammer)[, 1:4]
hammer.design = Biobase::pData(hammer)[1:4, ]
hammer.counts = hammer.counts[rowSums(hammer.counts) >= 5, ]
```

```
ss = subsample(hammer.counts, c(.01, .1, 1), treatment=hammer.design$protocol,
               method=c("edgeR", "DESeq2", "voomLimma"))

seed = getSeed(ss)

# generate the matrices used at each subsample
subm.01 = generateSubsampledMatrix(hammer.counts, .01, seed)
subm.1 = generateSubsampledMatrix(hammer.counts, .1, seed)
```

getSeed *Extract the global random seed from a subsamples object*

Description

A subsamples object, or a summary.subsamples object, does not contain the subsampled count matrix at each depth (as it would take too much space and is rarely used). However, as it saves the random seed used to generate the count matrix, the count matrix at any depth can be retrieved. This can be done for a subsamples object ss by retrieving the seed with getSeed(ss). If this seed is provided to the subsample function, then the same matrices will be generated when the proportion is the same.

This is useful for adding additional methods or subsampling depths to an existing subsamples object (after which they can be combined with combineSubsamples).

Usage

```
getSeed(ss)
```

Arguments

ss	A subsamples object, returned from the subsample function, or a summary of that object

Value

get seed of subSeq object

Examples

```
data(hammer)

hammer.counts = Biobase::exprs(hammer)[, 1:4]
hammer.design = Biobase::pData(hammer)[1:4, ]
hammer.counts = hammer.counts[rowSums(hammer.counts) >= 5, ]

ss = subsample(hammer.counts, c(.01, .1, 1), treatment=hammer.design$protocol,
               method=c("edgeR", "DESeq2", "voomLimma"))

seed = getSeed(ss)
```

`hammer` *ExpressionSet results from Hammer et al 2010*

Description

An ExpressionSet containing the results of the Hammer et al 2010 RNA-Seq study on the nervous system of rats (Hammer et al 2010). This dataset is used in the examples and vignette for the subSeq package.

This was downloaded from the ReCount database of analysis-ready RNA-Seq datasets (Frazee et al 2011).

Hammer, P., Banck, M. S., Amberg, R., Wang, C., Petznick, G., Luo, S., Khrebtukova, I., Schroth, G. P., Beyerlein, P., and Beutler, A. S. (2010). mRNA-seq with agnostic splice site discovery for nervous system transcriptomics tested in chronic pain. Genome research, 20(6), 847-860. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2877581/

Frazee, A. C., Langmead, B., and Leek, J. T. (2011). ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. BMC Bioinformatics, 12, 449. http://bowtie-bio.sourceforge.net/recount/

`plot.subsamples` *plot metrics as a function of subsampled read depth*

Description

Plot the number of genes found significant, the Spearman correlation of the effect size estimates with the full experiment, and the empirical false discovery rate as a function of the subsampled read depth. This determines whether these metrics saturate, which indicates that the experiment has an appropriate sequencing depth.

Usage

```
## S3 method for class 'subsamples'
plot(x, ...)
```

Arguments

`x`	a `subsamples` object
`...`	further arguments passed to or from other methods.

Details

This is an alias for the plot.summary.subsamples function, so that plotting can be done directly on the subsamples object. We recommend using `summary(ss)` first, so that the summary operation does not have to be performed each time the figure is plotted, and so the summary object can be examined on its own.

Value

plot a subSeq object

Examples

```
if (interactive()) {
# import the subsampling object (see ?subsample to see how ss is created)
data(ss)

# plot subsample object
plot(ss)
}
```

---

plot.summary.subsamples

plot metrics as a function of subsampled read depth

Description

Plot the number of genes found significant, the Spearman correlation of the effect size estimates with the full experiment, and the empirical false discovery rate as a function of the subsampled read depth. This determines whether these metrics saturate, which indicates that the experiment has an appropriate sequencing depth.

Usage

```
## S3 method for class 'summary.subsamples'
plot(x, ...)
```

Arguments

x	a summary.subsamples object
...	further arguments passed to or from other methods.

Value

see description

Examples

```
if (interactive()) {
  # import the subsampling object (see ?subsample to see how ss is created)
  data(ss)

  # summarise object
  ss <- summary(ss)

  # plot
  plot(ss)
}
```

ss *Subsampling results using the hammer dataset*

Description

The subsample object ss is the result from applying the subsample function to the hammer data set. The hypothesis test was a simple two-sample comparison (control vs. L5 SNL). Voom, DESeq2 and edgeR were used to test for differential expression at three different subsampling proportions: 0.01, 0.1 and 1. Genes with less than 5 counts across all replicates were filtered. For more details on how the object was generated, please see the subsample function.

The subsample object can then be used to determine whether an experiment has adequate read depth (see plot and summary functions).

Hammer, P., Banck, M. S., Amberg, R., Wang, C., Petznick, G., Luo, S., Khrebtukova, I., Schroth, G. P., Beyerlein, P., and Beutler, A. S. (2010). mRNA-seq with agnostic splice site discovery for nervous system transcriptomics tested in chronic pain. Genome research, 20(6), 847-860. http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2877581/

Frazee, A. C., Langmead, B., and Leek, J. T. (2011). ReCount: a multi-experiment resource of analysis-ready RNA-seq gene count datasets. BMC Bioinformatics, 12, 449. http://bowtie-bio.sourceforge.net/recount/

Examples

```
# import the subsampling object (see ?subsample to see how ss is created)
data(ss)

# summarise object
sum_ss <- summary(ss)
#plot
if (interactive()) {
  plot(ss)
}
```

subsample *Subsample reads and perform statistical testing on each sample*

Description

Perform subsampling at multiple proportions on a matrix of count data representing mapped reads across multiple samples in many genes. For each sample, perform some statistical operations.

Usage

```
subsample(counts, proportions, method = "edgeR", replications = 1,
  seed = NULL, qvalues = TRUE, env = parent.frame(), ...)
```

Arguments

`counts` Matrix of unnormalized counts

`proportions` Vector of subsampling proportions in (0, 1]

`method` One or more methods to be performed at each subsample, such as edgeR or DESeq (see Details)

`replications` Number of replications to perform at each depth

`seed` An initial seed, which will be stored in the output so that any individual simulation can be reproduced.

`qvalues` Whether q-values should be calculated for multiple hypothesis test correction at each subsample.

`env` Environment in which to find evaluate additional hander functions that are given by name

`...` Other arguments given to the handler, such as `treatment`

Details

Method represents the name of a handler function, which can be custom-written by the user.

If a gene has a count of 0 at a particular depth, we set the p-value to 1 and the coefficient to 0 to stay consistent between programs. If the gene has a count that is not 0 but the p-value is NA, we set the p-value to 1 but keep the estimated coefficient.

Value

A subsample S3 object, which is a data.table containing

`pvalue` A p-value calculated for each gene by the handler

`coefficient` An effect size (usually log fold change) calculated for each gene by the handler

`ID` gene ID

`count` the number of reads to this specific gene in this subsample

`depth` the overall sequencing depth of this subsample

`method` the method used (the name of the handler)

Examples

```
data(hammer)

hammer.counts = Biobase::exprs(hammer)[, 1:4]
hammer.design = Biobase::pData(hammer)[1:4, ]
hammer.counts = hammer.counts[rowSums(hammer.counts) >= 5, ]

ss = subsample(hammer.counts, c(.01, .1, 1), treatment=hammer.design$protocol,
                method=c("edgeR", "DESeq2", "voomLimma"))
```

summary.subsamples — *calculate summary statistics for each subsampled depth in a subsamples object*

Description

Given a subsamples object, calculate a metric for each depth that summarizes the power, the specificity, and the accuracy of the effect size estimates at that depth.

Usage

```
## S3 method for class 'subsamples'
summary(object, oracle = NULL, FDR.level = 0.05,
  average = FALSE, p.adjust.method = "qvalue", ...)
```

Arguments

`object`	a subsamples object
`oracle`	a subsamples object of one depth showing what each depth should be compared to; if NULL, each will be compared to the highest depth
`FDR.level`	A false discovery rate used to calculate the number of genes found significant at each level
`average`	If TRUE, averages over replications at each method+depth combination before returning
`p.adjust.method`	Method to correct p-values in order to determine significance. By default "qvalue", but can also be given any method that can be given to p.adjust.
`...`	further arguments passed to or from other methods.

Details

To perform these calculations, one must compare each depth to an "oracle" depth, which, if not given explicitly, is assumed to be the highest subsampling depth. This thus summarizes how closely each agrees with the full experiment: if very low-depth subsamples still agree, it means that the depth is high enough that the depth does not make a strong qualitative difference.

The concordance correlation coefficient is described in Lin 1989. Its advantage over the Pearson is that it takes into account not only whether the coefficients compared to the oracle close to a straight line, but whether that line is close to the x = y line.

Note that selecting average=TRUE averages the depths of the replicates (as two subsamplings with identical proportions may have different depths by chance). This may lead to depths that are not integers.

Value

A summary object, which is a `data.table` with one row for each subsampling depth, containing the metrics

`significant` number of genes found significant at the given FDR

`pearson` Pearson correlation of the coefficient estimates with the oracle

`spearman` Spearman correlation of the coefficient estimates with the oracle

`concordance` Concordance correlation of the coefficient estimates with the oracle

`MSE` mean squared error between the coefficient estimates and the oracle

`estFDP` estimated FDP: the estimated false discovery proportion, as calculated from the average oracle local FDR within genes found significant at this depth

`rFDP` relative FDP: the proportion of genes found significant at this depth that were not found significant in the oracle

`percent` the percentage of genes found significant in the oracle that were found significant at this depth

References

Lawrence I-Kuei Lin (March 1989). "A concordance correlation coefficient to evaluate reproducibility". Biometrics (International Biometric Society) 45 (1): 255-268.

Examples

```
# see subsample function to see how ss is generated
data(ss)
# summarise subsample object
ss.summary = summary(ss)
```

Index