Package ‘spkTools’

May 30, 2024

Version 1.60.0
Date 2016-02-10
Title Methods for Spike-in Arrays
Author Matthew N McCall <mccallm@gmail.com>, Rafael A Irizarry
        <rafa@jhu.edu>
Maintainer Matthew N McCall <mccallm@gmail.com>
Depends R (>= 2.7.0), Biobase (>= 2.5.5)
Suggests xtable
Imports Biobase (>= 2.5.5), graphics, grDevices, gtools, methods,
        RColorBrewer, stats, utils
Description The package contains functions that can be used to compare
        expression measures on different array platforms.
License GPL (>= 2)

URL http://bioconductor.org
biocViews Software, Technology, Microarray
git_url https://git.bioconductor.org/packages/spkTools
git_branch RELEASE_3_19
git_last_commit 481a493
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-29

Contents

affy ................................................................. 2
plotSpkBox ......................................................... 3
SpikeInExpressionSet-class ................................... 3
spkAccSD ............................................................. 5
spkAll ................................................................. 6
This is a SpikeInExpressionSet object containing the data from the Affymetrix HGU133A Spike-in Experiment.

Usage

data(affy)

Format

It contains a matrix of expression values and a matrix of nominal concentrations.

Source

plotSpkBox

__Boxplots of Fold Changes Calculated by spkBox__

**Description**

Plots boxplots of the data resulting from a call to spkBox.

**Usage**

```r
plotSpkBox(boxs, fc=2, box.names=NULL, ...)
```

**Arguments**

- `boxs`: the output of a call to spkBox
- `fc`: expected fold change
- `box.names`: names to be printed below each boxplot
- `...`: parameters passed to boxplot

**Value**

Boxplots for spike-in and non-spike-in comparisons stratified by ALE strata are produced.

**Author(s)**

Matthew N. McCall

**Examples**

```r
data(affy)
affySlope <- spkSlope(affy)
affyBox <- spkBox(affy, affySlope)
plotSpkBox(affyBox)
```

---

SpikeInExpressionSet-class

__Class to Contain and Describe High-Throughput Expression Level Assays with Spike-in Data__

**Description**

This is a class representation for spike-in expression data. SpikeInExpressionSet class is derived from ExpressionSet, and requires a matrix named `exprs` and a matrix named `spikeIn`.

**Extends**

Extends class ExpressionSet.
Creating Objects

createSpikeInExpressionSet(exprs, spikeIn, ...)
new("SpikeInExpressionSet", phenoData = new("AnnotatedDataFrame"), featureData = new("AnnotatedDataFrame"), experimentData = new("MIAME"), annotation = character(0), exprs = new("matrix"), spikeIn = new("matrix"))

This creates a SpikeInExpressionSet with assayData implicitly created to contain exprs and spikeIn. Additional named matrix arguments with the same dimensions as exprs are added to assayData; the row and column names of these additional matrices should match those of exprs and spikeIn.

new("SpikeInExpressionSet", assayData = assayDataNew(exprs=new("matrix"),spikeIn=new("matrix")), phenoData = new("AnnotatedDataFrame"), featureData = new("AnnotatedDataFrame"), experimentData = new("MIAME"), annotation = character(0),

This creates a SpikeInExpressionSet with assayData provided explicitly. In this form, the only required named argument is assayData.

Slots

Inherited from ExpressionSet:

assayData: Contains matrices with equal dimensions, and with column number equal to nrow(phenoData).
assayData must contain a matrix exprs and a matrix spikeIn with rows representing features and columns representing samples.

phenoData: See eSet
annotation See eSet
featureData See eSet
experimentData: See eSet

Methods

Class-specific methods:

spikeIn(SpikeInExpressionSet), spikeIn(SpikeInExpressionSet)<- Access and set elements named spikeIn in the AssayData-class slot.
spkSplit(SpikeInExpressionSet) creates two SpikeInExpressionSet objects – one with the spike-in probes and one with the non-spike-in probes.

For derived methods (see ExpressionSet).

See Also

eSet-class, ExpressionSet-class.
Examples

# create an instance of SpikeInExpressionSet
new("SpikeInExpressionSet")

new("SpikeInExpressionSet", exprs=matrix(runif(1000), nrow=100), spikeIn=matrix(rep(1:10,100), nrow=100))

# class specific methods
data(affy)
affySpikes <- spikeIn(affy)
affySplit <- spkSplit(affy)

---

spkAccSD

<table>
<thead>
<tr>
<th>spkAccSD</th>
<th>Accuracy Standard Deviation</th>
</tr>
</thead>
</table>

Description

Estimates the standard deviation for spike-ins at the lowest possible fold change in each bin.

Usage

spkAccSD(object, spkSlopeOut, tol=3)

Arguments

- object: a SpikeInExpressionSet object
- spkSlopeOut: the output from the spkSlope function
- tol: number of digits after decimal point

Value

returns the median absolute deviation (MAD) for each bin.

Author(s)

Matthew N. McCall

Examples

data(affy)
affySlope <- spkSlope(affy)
spkAccSD <- spkAccSD(affy, affySlope)
spkAll  

Spike-in Functions Wrapper

Description

A wrapper for the functions contained in the spkTools package, which calls each function.

Usage

```r
spkAll(object, label, model=expr~spike+probe+array, fc=NULL, tol=3, xrngs=NULL, yrngs=NULL, cuts=c(.6,.99), potQuantile=.995, gnn=c(25,100,10000), pch=".", output="eps")
```

Arguments

- **object**: a `SpikeInExpressionSet` object
- **label**: a character string to insert into the graphs and tables produced
- **model**: model to be passed to `spkAnova`
- **fc**: the fold change for which fold change plots will be produced
- **tol**: the number of digits after the decimal point in fc
- **xrngs**: ranges for the x-axis of each plot. d=density, s=slope, v=box, m=M vs A
- **yrngs**: ranges for the y-axis of each plot. d=density, s=slope, v=box, m=M vs A
- **cuts**: quantiles used to make the low, medium, and high bins
- **potQuantile**: the desired quantile to compute the probability of being above
- **gnn**: a vector of 3 numbers passed to `spkGNN`: the desired number of true positives, the number of truly expressed genes, and the number of truly unexpressed genes
- **pch**: plotting point to be used in `spkSlope`
- **output**: the format in which to save the plots produced. Options are "pdf" and "eps"

Value

The full complement of plots and tables described in the vignette are created and saved in the current working directory.

Author(s)

Matthew N. McCall

Examples

```r
data(affy)
spkAll(affy, label="affy", fc=2)
```
spkAnova

Anova Model for Microarray Spike-in Data

Description
Computes the mean squared errors of a microarray spike-in design due to concentration, probe, array, and error.

Usage
spkAnova(object, model=expr~spike+probe+array)

Arguments
- object: a SpikeInExpressionSet object
- model: the anova model

Value
A vector of the mean squared errors from the anova model.

Author(s)
Matthew N. McCall

Examples
data(affy)
spkAnova(affy)

spkBal
Quantify Microarray Spike-in Design Imbalance

Description
Computes the imbalance of a microarray spike-in design due to probes and arrays.

Usage
spkBal(object)

Arguments
- object: a SpikeInExpressionSet object
The probe and array imbalances.

Matthew N. McCall


data(affy)
spkBal(affy)

spkBox

A function to calculate the log-ratios stratified by which ALE groups yield the comparison. They are stratified by which bins are being compared to produce the given fold change.

spkBBox(object, spkSlopeOut, fc = 2, tol = 3, reduce=TRUE)

object a SpikeInExpressionSet object

spkSlopeOut the output of the spkSlope function

fc the fold change of interest

tol the precision (number of digits after decimal point) in fc

reduce if TRUE the number of points plotted in the null bins is reduced

This function requires the output of spkSlope.

A list with the log-ratios separated by ALE strata comparison.
**spkDensity**

**Author(s)**
Matthew N. McCall

**Examples**

```r
data(affy)
affySlope <- spkSlope(affy)
spkBox(affy, affySlope)
```

---

**spkDensity**  
*Spike-in Density Plot*

---

**Description**

A density plot of the non-spike-in expression with a rug of the average expression at each spike-in level.

**Usage**

```r
spkDensity(object, spkSlopeOut, cuts=TRUE, label = NULL, ...)
```

**Arguments**

- `object`: a SpikeInExpressionSet object
- `spkSlopeOut`: the output from the spkSlope function
- `cuts`: if TRUE vertical lines are drawn at the expression values separating low vs medium and medium vs high ALE strata
- `label`: a character string to insert into the plot title
- `...`: arguments passed to the plot function

**Details**

This function requires the output of spkSlope.

**Value**

Density plot is produced.

**Author(s)**
Matthew N. McCall

**Examples**

```r
data(affy)
affySlope <- spkSlope(affy)
spkDensity(affy, affySlope)
```
spkGNN  

**Genes Needed to Detect N True Positives**

**Description**

Computes the number of genes one would need to consider to obtain a given number of truly positive genes if one considered genes in order of decreasing observed fold change.

**Usage**

```r
spkGNN(n, n.expr, n.unexpr, AccuracySlope, AccuracySD, nullfc)
```

**Arguments**

- `n` the desired number of true positives
- `n.expr` the actual number of truly expressed genes
- `n.unexpr` the actual number of truly unexpressed genes
- `AccuracySlope` the signal detect slope from the spkSlope function
- `AccuracySD` the standard deviation of the signal detect slope from the spkAccSD function
- `nullfc` a vector of null fold changes from the spkBox function

**Value**

This function returns the expected number of genes one would have to consider to obtain N true positives under the given conditions.

**Author(s)**

Matthew N. McCall

**Examples**

```r
data(affy)
spkSlopeOut <- spkSlope(affy)
spkBoxOut <- spkBox(affy, spkSlopeOut, fc=2)
AccuracySlope <- round(spkSlopeOut$slope[-1], digits=2)
AccuracySD <- round(spkAccSD(affy, spkSlopeOut), digits=2)
spkGNN(n=25, n.expr=100, n.unexpr=10000, AccuracySlope[2], AccuracySD[2], spkBoxOut[[2]])
```
spkMA

MA Plots

Description

Plots log-ratios (M) vs. average log expression (A) for a SpikeInExpressionSet object.

Usage

spkMA(object, spkSlopeOut, fc=2, tol=3, label=NULL, ylim=NULL, outlier=1, reduce=TRUE, plot.legend=TRUE)

Arguments

object a SpikeInExpressionSet object
spkSlopeOut the output from the spkSlope function
fc the fold change of interest
tol the precision (number of digits after decimal point) in fc
label a character string to insert into the plot title
ylim limits of y-axis
outlier log fold change cut-off for outliers
reduce if TRUE some points are removed from the background to speed plotting
plot.legend if TRUE a legend is plotted

Value

The MA plot is produced.

Author(s)

Matthew N. McCall

Examples

data(affy)
affySlope <- spkSlope(affy)
spkMA(affy, affySlope)
Pairwise Comparisons for Spike-in Genes

Description
Compute log-ratios among spike-in genes.

Usage
spkPair(object)

Arguments
object      a SpikeInExpressionSet object

Value
An array containing either log-ratios (M), average log expression (A), and nominal concentrations (N1 & N2). Dimension one is genes, dimension two is array pairings, dimension three is M, A, N1, and N2.

Author(s)
Matthew N. McCall

Examples
data(affy)
affyPair <- spkPair(affy)

Pairwise Comparisons for Non-Spike-in Genes

Description
Compute log-ratios among non-spike-in genes.

Usage
spkPairNS(object, output="M")

Arguments
object      a SpikeInExpressionSet object
output      what to return; either "M" for log-ratios or "A" for average log expression.
**spkPot**

**Value**
A matrix containing either log-ratios (M) or average log expression (A). Rows are genes and columns are array pairings.

**Author(s)**
Matthew N. McCall

**Examples**
```
data(affy)
affyPairNS <- spkPairNS(affy)
```

---

<table>
<thead>
<tr>
<th>spkPot</th>
<th>Probability of being in the Top</th>
</tr>
</thead>
</table>

**Description**
Compute the probability that a spike-in with a nominal fold change of 2 appears in the the top 0.5% (default) of log-ratios.

**Usage**
```
spkPot(object, spkSlopeOut, sig, SD, precisionQuantile)
```

**Arguments**
- `object`: a `SpikeInExpressionSet` object
- `spkSlopeOut`: the output from the `spkSlope` function
- `sig`: the signal detect slopes from a call to `spkSlope`
- `SD`: the standard deviation from `spkAccSD`
- `precisionQuantile`: the desired quantile to compute the probability of being above

**Value**
A vector of probabilities for each ALE strata.

**Author(s)**
Matthew N. McCall

**Examples**
```
data(affy)
affySlope <- spkSlope(affy)
affyAccSD <- spkAccSD(affy, affySlope)
spkPot(affy, affySlope, affySlope$slopes, affyAccSD, .995)
```
spkQuantile

**Empirical Quantiles**

**Description**

An internal function called by spkSlope.

**Usage**

```r
spkQuantile(amt, avgE, ens, p)
```

**Arguments**

- `amt`: a vector of nominal concentrations
- `avgE`: the observed average expression corresponding to each nominal concentration
- `ens`: the average expression across arrays of unexpressed genes
- `p`: the quantiles to make the bins

**Author(s)**

Matthew N. McCall

**Examples**

```r
data(affy)
affySlope <- spkSlope(affy)
```

spkSlope

**Signal Detect Slope Plot**

**Description**

Plots observed expression vs. nominal concentration. The overall regression slope, as well as, regression slopes for low, medium, and high bins are computed and the regression lines plotted.

**Usage**

```r
spkSlope(object, label = NULL, cuts=c(.6,.99), ...)
```

**Arguments**

- `object`: a SpikeInExpressionSet object
- `label`: a character string to insert into the plot title
- `cuts`: quantiles used to make the low, medium, and high bins
- `...`: arguments passed to the plot function
Details

The bins are created by computing the proportion of non-spike-in genes with expression values less than or equal to the average expression value at each nominal concentration. Using the default value of cuts, the high bin contains nominal concentrations with 99 percent or more of the non-spike-in expression values lower than it. The medium bin contains nominal concentrations with between 60 and 99 percent of the non-spike-in expression values lower than it. The low bin contains nominal concentrations with less than 60 percent of the non-spike-in expression values lower than it.

Value

- `avgExp`: average expression at each nominal concentration
- `slopes`: the regression slopes - overall and for each bin
- `breaks`: which spike-in levels fall in each bin
- `brkpts`: the expression value of the cut points between bins
- `prop`: the proportion of non-spike-in probes with expression less than the average expression at each nominal concentration

Author(s)

Matthew N. McCall

Examples

```r
data(affy)
spkSlope(affy)
```

Description

A collection of functions to examine microarray datasets that include spike-ins. In particular, it allows one to explore the distribution of spike-ins within the range of possible expression values, the relationship between nominal concentration and expression, and the relationship between expected and observed fold change for different levels of comparison.

Details

```r
Package: spkTools
Type: Package
Version: 0.0.1
Date: 2007-10-9
License: GPL version 2 or newer
```
Author(s)
Matthew N. McCall
Maintainer: Matthew N. McCall <mmccall@jhsph.edu>

Examples
## The Three Plots
data(affy)
par(mfrow=c(2,2))
affySlope <- spkSlope(affy)
spkDensity(affy, affySlope)
spkBox(affy, affySlope)

## The Full Wrapper
data(affy)
spkAll(affy, label="Affymetrix", fc=2)

---

### spkVar

**Spike-in Variance**

**Description**
Compute an estimate of the standard deviation in expression at each nominal concentration.

**Usage**

```r
spkVar(object)
```

**Arguments**

- `object` a SpikeInExpressionSet object

**Value**

a matrix containing spike-in levels and corresponding MADs.

**Author(s)**
Matthew N. McCall

**Examples**

```r
data(affy)
spkVar(affy)
```
summarySpkBox

Summary of Fold Changes Calculated by spkBox

Description

Prints a summary table of the data resulting from a call to spkBox.

Usage

summarySpkBox(boxs)

Arguments

boxs the output of a call to spkBox

Value

A dataframe with 2 columns: the mean fold change and the median average distance of the fold changes.

Author(s)

Matthew N. McCall

Examples

data(affy)
affySlope <- spkSlope(affy)
affyBox <- spkBox(affy, affySlope)
plotSpkBox(affyBox)
Index

* **classes**
  - SpikeInExpressionSet-class, 3

* **datasets**
  - affy, 2

* **hplot**
  - plotSpkBox, 3
  - spkAll, 6
  - spkDensity, 9
  - spkMA, 11
  - spkSlope, 14

* **manip**
  - spkAccSD, 5
  - spkAll, 6
  - spkAnova, 7
  - spkBal, 7
  - spkBox, 8
  - spkGNN, 10
  - spkPair, 12
  - spkPairNS, 12
  - spkPot, 13
  - spkQuantile, 14
  - spkSlope, 14
  - spkVar, 16
  - summarySpkBox, 17

* **package**
  - spkTools, 15

affy, 2

class:SpikeInExpressionSet
  - (SpikeInExpressionSet-class), 3

initialize,SpikeInExpressionSet-method
  - (SpikeInExpressionSet-class), 3

plotSpkBox, 3

spikeIn (SpikeInExpressionSet-class), 3
spikeIn, SpikeInExpressionSet-method
  - (SpikeInExpressionSet-class), 3

spikeIn<-,SpikeInExpressionSet,matrix-method
  - (SpikeInExpressionSet-class), 3

SpikeInExpressionSet
  - (SpikeInExpressionSet-class), 3

SpikeInExpressionSet-class, 3
SpikeInExpressionSet-methods
  - (SpikeInExpressionSet-class), 3

spkAccSD, 5
spkAll, 6
spkAnova, 7
spkBal, 7
spkBox, 8
spkDensity, 9
spkGNN, 10
spkMA, 11
spkPair, 12
spkPairNS, 12
spkPot, 13
spkQuantile, 14
spkSlope, 14
spkSplit (SpikeInExpressionSet-class), 3
spkSplit,SpikeInExpressionSet-method
  - (SpikeInExpressionSet-class), 3

spkTools, 15
spkTools-package (spkTools), 15
spkVar, 16
summarySpkBox, 17