Description
This is a SpikeInExpressionSet object containing the data from the Affymetrix HGU133A Spike-in Experiment.

Usage
\texttt{data(affy)}

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

Boxplots of Fold Changes Calculated by spkBox

Description

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

Usage

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

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 names exprs and a matrix named spikeIn.

Extends

Extends class ExpressionSet.

Creating Objects

```
createSpikeInExpressionSet(exprs, spikeIn, ...)
```

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), exprs = new("matrix"), spikeIn = new("matrix"))
```

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)


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

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

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)

spkBali Quantify Microarray Spike-in Design Imbalance

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

Usage
spkBali(object)

Arguments
object a SpikeInExpressionSet object

Value
The probe and array imbalances.
**spkBox**

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

**References**

**Examples**
```
data(affy)
spkBal(affy)
```

---

### Fold Change Calculations

**Description**
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.

**Usage**
```
spkBox(object, spkSlopeOut, fc = 2, tol = 3, reduce=TRUE)
```

**Arguments**
- **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

**Details**
This function requires the output of spkSlope.

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

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

**Examples**
```
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

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

data(affy)
affySlope <- spkSlope(affy)
spkDensity(affy, affySlope)
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

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

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)
**spkPairNS**

*Pairwise Comparisons for Non-Spike-in Genes*

**Description**

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

**Usage**

```r
spkPairNS(object, output="M")
```

**Arguments**

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

**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**

```r
data(affy)
affyPairNS <- spkPairNS(affy)
```

**spkPair**

*Pairwise Comparisons for Spike-in Genes*

**Description**

Compute log-ratios among spike-in genes.

**Usage**

```r
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.
spkPot

Author(s)
Matthew N. McCall

Examples

```r
data(affy)
affyPair <- spkPair(affy)
```

---

**spkPot**

*Probability of being in the Top*

Description

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

Usage

```r
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

```r
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

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

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

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
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**

<table>
<thead>
<tr>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>avgExp</td>
<td>average expression at each nominal concentration</td>
</tr>
<tr>
<td>slopes</td>
<td>the regression slopes - overall and for each bin</td>
</tr>
<tr>
<td>breaks</td>
<td>which spike-in levels fall in each bin</td>
</tr>
<tr>
<td>brkpts</td>
<td>the expression value of the cut points between bins</td>
</tr>
<tr>
<td>prop</td>
<td>the proportion of non-spike-in probes with expression less than the average expression at each nominal concentration</td>
</tr>
</tbody>
</table>

**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**

- **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

```r
## 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)
```
Summary of Fold Changes Calculated by `spkBox`

**Description**

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

**Usage**

```r
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**

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

*Topic classes
  SpikeInExpressionSet-class, 2
*Topic datasets
  affy, 1
*Topic hplot
  plotSpkBox, 1
  spkAll, 4
  spkDensity, 7
  spkMA, 9
  spkSlope, 12
*Topic manip
  spkAccSD, 3
  spkAll, 4
  spkAnova, 5
  spkBal, 6
  spkBox, 6
  spkGNN, 8
  spkPair, 10
  spkPairNS, 10
  spkPot, 11
  spkQuantile, 12
  spkSlope, 12
  spkVar, 14
  summarySpkBox, 15
*Topic package
  spkTools, 13

affy, 1

class:SpikeInExpressionSet
  (SpikeInExpressionSet-class), 2

initialize,SpikeInExpressionSet-method
  (SpikeInExpressionSet-class), 2

plotSpkBox, 1

spikeIn
  (SpikeInExpressionSet-class), 2
spikeIn,SpikeInExpressionSet-method
  (SpikeInExpressionSet-class), 2