Package ‘cummeRbund’

April 9, 2015

Title Analysis, exploration, manipulation, and visualization of Cufflinks high-throughput sequencing data.

Version 2.8.2

Date 2013-04-22

Author L. Goff, C. Trapnell, D. Kelley

Description Allows for persistent storage, access, exploration, and manipulation of Cufflinks high-throughput sequencing data. In addition, provides numerous plotting functions for commonly used visualizations.

Imports methods, plyr, BiocGenerics, Biobase

Depends R (>= 2.7.0), BiocGenerics (>= 0.3.2), RSQLite (>= 1.0.0), ggplot2, reshape2, fastcluster, rtracklayer, Gviz

Suggests cluster, plyr, NMFN, stringr, GenomcFeatures, GenomicRanges, rjson

Maintainer Loyal A. Goff <lgoff@csail.mit.edu>

License Artistic-2.0


LazyLoad yes

biocViews Sequencing, RNASeq, GeneExpression, DifferentialExpression, Infrastructure, DataImport, DataRepresentation, Visualization, Clustering, MultipleComparison, QualityControl

R topics documented:

cummeRbund-package ......................................................... 3
addFeatures ............................................................... 5
count-methods ............................................................. 6
countMatrix ................................................................. 7
csBoxplot ................................................................. 8
csCluster ................................................................. 9
topics documented:

csClusterPlot .......................................................... 10
csDendro ................................................................. 11
csDensity ................................................................. 12
csDistHeat .............................................................. 14
csHeatmap ............................................................... 15
csScatter ................................................................. 17
csSpecificity ............................................................. 18
csVolcano ................................................................. 19
CuffData-class .......................................................... 20
CuffDist-class .......................................................... 21
CuffFeature-class ..................................................... 22
CuffFeatureSet-class ................................................ 24
CuffGene-class ........................................................ 25
CuffGeneSet-class ...................................................... 27
CuffSet-class ........................................................... 28
diffData ................................................................. 30
Dimensionality Reduction ........................................... 31
dispersionPlot .......................................................... 32
distValues ............................................................... 33
Exploratory Analysis .................................................. 34
expressionBarplot ..................................................... 35
expressionPlot ........................................................ 37
featureNames ........................................................... 38
features ................................................................. 39
findGene ................................................................. 40
findSimilar ............................................................. 41
fpkm-methods ........................................................... 42
fpkmMatrix .............................................................. 43
getFeatures ............................................................. 44
getGene ................................................................. 45
getGeneId ............................................................... 46
getGenes ............................................................... 47
getLevels ............................................................... 48
getRepLevels .......................................................... 49
getSig ................................................................. 50
getSigTable ............................................................ 51
JSDist ................................................................. 52
JSDistFromP ............................................................ 53
JSDistVec .............................................................. 54
makeprobs .............................................................. 55
makeprobsvec ........................................................ 56
MAplot ............................................................... 57
PINK1 ................................................................. 58
QCplots ............................................................... 59
readCufflinks .......................................................... 60
repFpkm-methods ..................................................... 62
replicates ............................................................. 63
runInfo ................................................................. 64
Description

Allows for persistent storage, access, and manipulation of Cufflinks high-throughput sequencing data. In addition, provides numerous plotting functions for commonly used visualizations. ~ A concise (1-5 lines) description of the package ~

Details

Package: cummeRbund
Version: 0.1.3
Suggests:
Depends: R (>= 2.7.0), RSQLite, reshape2, ggplot2, methods
License: MIT License
LazyLoad: yes
biocViews: HighThroughputSequencing, HighThroughputSequencingData, RNAseq, RNAseqData, GeneExpression, DifferentiallyExpressed, DataImport, DataRepresentation, Visualization, Bioinformatics, Clustering, MultipleComparisons, QualityControl
Packaged: 2011-08-05 18:03:50 UTC; lgoff
Built: R 2.12.1; ; 2011-08-05 18:03:57 UTC; unix

Index:

CuffData-class Class "CuffData"
CuffDist-class Class "CuffDist"
CuffFeature-class Class "CuffFeature"
CuffFeatureSet-class Class "CuffFeatureSet"
CuffGene-class Class "CuffGene"
CuffGeneSet-class Class "CuffGeneSet"
CuffSet-class Class "CuffSet"
JSdist Jensen-Shannon distance on columns
addFeatures addFeatures
addFeatures-methods ~ Methods for Function addFeatures in Package cummeRbund ~
createDB createDB
cummeRbund-package

cummeRbund-package

cummeRbund: The finishing touch on your Tuxedo workflow. Analysis, manipulation, and visualization of Cufflinks HTS data.

diffData

differential comparison data

diffData-methods

~ Methods for Function diffData in Package cummeRbund ~

dim-methods

~ Methods for Function dim in Package base ~

expressionBarplot

Barplot

expressionBarplot-methods

~ Methods for Function expressionBarplot in Package cummeRbund ~

expressionPlot

Expression Plot

expressionPlot-methods

~ Methods for Function expressionPlot in Package cummeRbund ~

featureNames

Feature names

featureNames-methods

~ Methods for Function featureNames in Package cummeRbund ~

features

Features

features-methods

~ Methods for Function features in Package cummeRbund ~

fpkm

Retrieve FPKM values

fpkm-methods

~ Methods for Function fpkm in Package cummeRbund ~

fpkmMatrix

Retrieve FPKM values as matrix

fpkmMatrix-methods

~ Methods for Function fpkmMatrix in Package cummeRbund ~

getGene

gene

getGene-methods

~ Methods for Function getGene in Package cummeRbund ~

getGenes

gene

getGenes-methods

~ Methods for Function getGenes in Package cummeRbund ~
addFeatures

### Description

Adds a data.frame of features to a the SQLite backend database.

### Usage

```r
## S4 method for signature CuffSet
addFeatures(object, features, level="genes", ...)
```
Arguments

- **object**: An object of class (`'CuffSet'` or `'CuffData'`)
- **features**: A data.frame of features to add. 1st column MUST contain ids (i.e., `gene_id` for `'gene'` features, `isoform_id` for `'isoform'` features, etc)
- **level**: One of c(`'genes','isoforms','TSS','CDS'`) to indicate which type of features you are being added, and to what data-level.
- ... Additional arguments.

Details

None

Value

None

Note

None

Author(s)

Loyal A. Goff

References

None

Examples

# None yet.

---

### count-methods

*Retrieve count values (raw and normalized)*

### Description

Returns a data.frame from @count slot

### Details

Returns a data.frame of count values.

### Value

A data.frame of count-level values for a set of features.
Methods

signature(object = "CuffData")
signature(object = "CuffFeature")
signature(object = "CuffFeatureSet")

Note

None

Author(s)

Loyal A. Goff

References

None

Examples

data(sampleData)
count(PINK1)

Description

Retrieve count values as gene by condition matrix

Usage

## S4 method for signature CuffData
countMatrix(object, fullnames=FALSE, sampleIdList)
## S4 method for signature CuffData
repCountMatrix(object, fullnames=FALSE, repIdList)

Arguments

object An object of class (`CuffData`, `CuffFeatureSet`, `CuffGeneSet`, `CuffGene`, or `CuffFeature`)
fullnames A logical value whether or not to concatenate gene_short_name and tracking_id values (easier to read labels)
sampleIdList A vector of sample names to subset the resulting matrix.
repIdList A vector of replicate names to subset the resulting replicate matrix.
csBoxplot

Details
None.

Value
A feature x condition matrix of count values.

Note
None

Author(s)
Loyal A. Goff

References
None.

Examples

```r
data(sampleData)
countMatrix(sampleGeneSet)
repCountMatrix(sampleGeneSet)
```

Description
Creates a ggplot2 plot object with a geom_box layer displaying summary statistics for FPKM values across samples (x).

Usage

```r
## S4 method for signature CuffData
csBoxplot(object, logMode=TRUE, pseudocount=0.0001, replicates=FALSE,...)
```

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>object</td>
<td>An object of class CuffData.</td>
</tr>
<tr>
<td>logMode</td>
<td>A logical argument to log10-transform FPKM values.</td>
</tr>
<tr>
<td>pseudocount</td>
<td>Value added to FPKM to avoid log-transform issues.</td>
</tr>
<tr>
<td>replicates</td>
<td>A logical value whether or not to plot individual replicates or aggregate condition values.</td>
</tr>
<tr>
<td>...</td>
<td>Additional arguments to csBoxplot</td>
</tr>
</tbody>
</table>
### csCluster

**Details**

None

**Value**

A ggplot2 plot object with a geom_box layer.

**Note**

None

**Author(s)**

Loyal A. Goff

**References**

None

**Examples**

```r
a<-readCufflinks(system.file("extdata", package="cummeRbund")) # Read cufflinks data and create CuffSet object
genes<-a@genes # CuffData object for all genes
csBoxplot(genes)
```

------

### Description

Returns a ggplot2 plot object with geom_line layer plotting FPKM values over conditions faceted by k-means clustering clusters. (Euclidean). This is very crude at this point. This does not return any of the clustering information directly, but if you want it, you can retrieve it from the ggplot object returned.

**Usage**

```r
# S4 method for signature CuffFeatureSet
csCluster(object,k,logMode=T,method = "none",pseudocount=1,...)
```

**Arguments**

- `object` An object of class CuffFeatureSet.
- `k` Number of pre-defined clusters to attempt to find.
- `logMode` A logical value whether or not to log-transform the FPKM values prior to clustering.
method Distance function to use when computing cluster solution. Default "none" will use the Jensen-Shannon distance (JSdist). Provide a function that returns a dist object on rows.
pseudocount Value added to FPKM to avoid log-transform issues.
... Additional arguments to pam.

Details
Uses 'kmeans' function.

Author(s)
Loyal A. Goff

Source
None

References
None.

Examples
data(sampleData)
csCluster(sampleGeneSet,4)

csClusterPlot clus
csClusterPlot clus

description
Replaces the default plotting behavior of the old csCluster. Takes as an argument the output of csCluster and plots expression profiles of features facet by cluster.

Usage
csClusterPlot(clustering, pseudocount=1.0,logMode=FALSE,drawSummary=TRUE,sumFun=mean_cl_boot)

Arguments
clustering The output of csCluster. (Must be the output of csCluster. Only this data format contains the necessary information for csClusterPlot.)
pseudocount Value added to FPKM to avoid log transformation issues.
logMode Logical argument whether to plot FPKM with log axis (Y-axis).
drawSummary Logical value whether or not to draw a summary line for each cluster (by default this is the cluster mean)
sumFun Summary function used to by drawSummary (default: mean_cl_boot)
Details

This replaces the default plotting behavior of the old csCluster() method. This was necessary so as to preserve the cluster information obtained by csCluster in a stable format. The output of csClusterPlot is a ggplot2 object of expressionProfiles faceted by cluster ID.

Value

A ggplot2 object of expressionProfiles faceted by cluster ID.

Note

None.

Author(s)

Loyal A. Goff

References

None.

Examples

data(sampleData)
myClustering<-csCluster(sampleGeneSet,k=4)
csClusterPlot(myClustering)

csDendro

Description

Creates a grid graphics plot of a dendrogram of Jensen-Shannon distances between conditions of a CuffFeatureSet or CuffGeneSet object.

Usage

## S4 method for signature CuffFeatureSet
csDendro(object,logMode=T,pseudocount=1,replicates=FALSE)
## S4 method for signature CuffData
csDendro(object,logMode=T,pseudocount=1,replicates=FALSE,...)
Arguments

object An object of class 'CuffFeatureSet' or 'CuffGeneSet'
logMode A logical argument to log10-transform FPKM values prior to plotting.
pseudocount Value to be added to FPKM for appropriate log transformation and clustering. (Avoids zero-based errors)
replicates A logical value whether or not to plot individual replicates or aggregate condition values.
... Additional arguments to csHeatmap

Details

None

Value

Returns a dendrogram object and plots that object by default.

Note

None

Author(s)

Loyal A. Goff and Cole Trapnell

References

None.

Examples

data(sampleData)
csDendro(sampleGeneSet)

csDensity

Density plot of CuffData

Description

Creates a smoothed density plot, by sample, for log10 FPKM values from a cuffdiff run.

Usage

## S4 method for signature CuffData
csDensity(object, logMode=TRUE, pseudocount=0, labels, features=FALSE, replicates=FALSE,...)
## S4 method for signature CuffFeatureSet
csDensity(object, logMode=TRUE, pseudocount=0, labels, features=FALSE, replicates=FALSE,...)
Arguments

object An object of class CuffData.

logMode A logical value of whether or not to log10-transform FPKM values. By default this is TRUE.

pseudocount Pseudocount value added to FPKM to avoid errors in log-transformation of true zero values.

labels A list of tracking_id values or gene_short_name values used for 'callout' points on the density plot for reference. (Not implemented yet).

features Will include all fields from 'features' slot in returned ggplot object. Useful for further manipulations of plot object using feature-level attributes (e.g. gene_type, class_code, etc)

replicates A logical value whether or not to plot individual replicates or aggregate condition values.

... Additional arguments

Details

Creates a density plot, by sample, for log10-transformed FPKM values from a cuffdiff run.

Value

A ggplot2 plot object

Note

None

Author(s)

Loyal A. Goff

References

None

Examples

a<-readCufflinks(system.file("extdata", package="cummeRbund")) #Create CuffSet object from sample data
genes<-a@genes #Create CuffData object for all genes
d<-csDensity(genes) #Create csDensity plot
d #Render plot
Description

Creates a ggplot plot object with a geom_tile layer of JS Distance values between samples or genes.

Usage

```r
## S4 method for signature CuffFeatureSet
csDistHeat(object, replicates=F, samples.not.genes=T,
logMode=T, pseudocount=1.0/
heatscale=c(low=lightyellow,mid=orange,high=darkred),
heatMidpoint=NULL, ...)
```

Arguments

- `object`: An object of class 'CuffFeatureSet' or 'CuffGeneSet'
- `replicates`: A logical argument whether or not to use individual replicate FPKM values as opposed to condition FPKM estimates. (default: FALSE)
- `samples.not.genes`: Compute distances between samples rather than genes. If False, compute distances between genes.
- `logMode`: A logical argument to log10-transform FPKM values prior to plotting.
- `pseudocount`: Value to be added to FPKM for appropriate log transformation and clustering. (Avoids zero-based errors)
- `heatscale`: A list with min length=2, max length=3 that describe the color scale.
- `heatMidpoint`: Value for midpoint of color scale.
- `...`: Additional arguments to csHeatmap

Details

None

Value

A ggplot2 plot object with a geom_tile layer to display distance between samples or genes.

Note

None

Author(s)

Loyal A. Goff, Cole Trapnell, and David Kelley
References

None

Examples

data(sampleData)
csDistHeat(sampleGeneSet)

csHeatmap

csHeatmap

Description

Creates a ggplot plot object with a geom_tile layer of FPKM values per feature and sample.

Usage

## S4 method for signature CuffFeatureSet
csHeatmap(object, rescaling=none, clustering=none, labCol=T, labRow=T, logMode=T, pseudocount=1./zero.noslash,
border=FALSE, heatscale= c(low=lightyellow,mid=orange,high=darkred), heatMidpoint=NULL, fullnames = T,
method=none, ...)
## S4 method for signature CuffFeatureSet
csFoldChangeHeatmap(object, control_condition, replicate_num=NULL, clustering=none, labCol=T, labRow=T,
logMode=F, pseudocount=1./zero.noslash, border=FALSE, heatscale=c(low=steelblue,mid=white,high=tomato), heatMidpoint=0,fullnames=T,replicate=none, ...)

Arguments

object An object of class 'CuffFeatureSet' or 'CuffGeneSet'
control_condition A character argument indicating which condition should be used as the denominator for fold change. (e.g. "Day0", "Control", etc)
replicate_num If replicates == TRUE, you must specify both a control condition and a replicate number to use as the denominator.
rescaling Rescaling can either be 'row' or 'column' OR you can pass rescale a function that operates on a matrix to do your own rescaling. Default is 'none'.
clustering Clustering can either be 'row','column','none', or 'both', in which case the appropriate indices are re-ordered based on the pairwise Jensen-Shannon distance of FPKM values.
labCol A logical argument to display column labels.
labRow A logical argument to display row labels.
logMode A logical argument to log10-transform FPKM values prior to plotting.
pseudocount Value to be added to FPKM for appropriate log transformation and clustering. (Avoids zero-based errors)
border A logical argument to draw border around plot.
 heatscale  A list with min length=2, max length=3 that detail the low,mid,and high colors to build the color scale.
 heatMidpoint  Value for midpoint of color scale.
 fullnames  A logical value whether to use 'fullnames' (concatenated gene_short_name and gene_id) for rows in heatmap. Default [ TRUE ].
 replicates  A logical value whether or not to plot individual replicates or aggregate condition values.
 method  Function to be used for clustering. Default is JS-distance. You can pass your own function to this argument as long as the output is an instance of the 'dist' class and is applied to the rows of the input matrix.
 heatRange  Numerical argument for upper bound on log fold change to be visualized.
 ...  Additional arguments to csHeatmap

Details
None

Value
A ggplot2 plot object with a geom_tile layer to display FPKM values by sample (x) and feature (y)

Note
None

Author(s)
Loyal A. Goff and Cole Trapnell

References
None.

Examples

data(sampleData)
csHeatmap(sampleGeneSet)
Description

A scatter plot comparing the FPKM values from two samples in a cuffdiff run.

Usage

```r
## S4 method for signature CuffData
csScatter(object, x, y, logMode=TRUE, pseudocount=1.0, labels, smooth=FALSE, colorByStatus = FALSE, drawRug=TRUE, ...) # S4 method for signature CuffData
csScatterMatrix(object, replicates=FALSE, logMode=TRUE, pseudocount=1.0, hexbin=FALSE, useCounts=FALSE, ...)
```

Arguments

- `object`: An object of class (`'CuffData','CuffFeatureSet'`)
- `x`: Sample name for x axis
- `y`: Sample name for y axis
- `logMode`: Logical argument to render axes on log10 scale (default: `T`)
- `replicates`: Logical argument whether or not to draw individual replicate values instead of condition values. (default: `T`)
- `pseudocount`: Value to add to zero FPKM values for log transformation (default: `0.0001`)
- `smooth`: Logical argument to add a smooth-fit regression line
- `labels`: A list of tracking_ids or gene_short_names that will be 'callout' points in the plot for reference. Useful for finding genes of interest in the field. Not implemented yet.
- `colorByStatus`: A logical argument whether or not to color the points by 'significant' Y or N. (Default = `FALSE`)
- `drawRug`: A logical argument whether or not to draw the rug for x and y axes (Default = `TRUE`)
- `hexbin`: Logical value whether or not to visualize overplotting with hexbin.
- `useCounts`: Logical value whether or not to visualize overplotting with hexbin.
- `...`: Additional arguments to csScatter

Details

None

Value

A ggplot object with geom_point and geom_rug layers.
Note
None

Author(s)
Loyal A. Goff

References
None

Examples
a<-readCufflinks(system.file("extdata", package="cummeRbund")) #Create CuffSet object from sample data
genes<-a@genes #Create CuffData object for all genes
s<-csScatter(genes,hESC,Fibroblasts,smooth=TRUE) #Create plot object
s #render plot object

Description
Returns a matrix of 'Specificity scores' (S) defined as 1-JSD(p_g,q_i) where p_g is the Log10+1 expression profile of a gene g across all conditions j, collapsed into a probability distribution, and q_i is the unit vector of 'perfect expression' in a given condition i.

Usage
## S4 method for signature CuffFeatureSet
csSpecificity(object,logMode=T,pseudocount=1,relative=FALSE,...)
## S4 method for signature CuffData
csSpecificity(object,logMode=T,pseudocount=1,relative=FALSE,...)

Arguments
object An object of class CuffFeatureSet, CuffGeneSet, or CuffData.
logMode A logical argument to log10-transform FPKM values prior to plotting.
pseudocount Value to be added to FPKM for appropriate log transformation and clustering.
(Avoids zero-based errors)
relative A logical argument that when TRUE, will scale the S values from 0-1 by dividing by max(S)
... Additional arguments to fpkmMatrix.

Details
None
**csVolcano**

**Author(s)**
Loyal A. Goff

**Source**
None

**References**
None.

**Examples**

```r
data(sampleData)
csSpecificity(sampleGeneSet)
```

---

**Description**

Creates a volcano plot of log fold change in expression vs -log(pval) for a pair of samples (x,y)

**Usage**

```r
## S4 method for signature CuffData
csVolcano(object, x, y, alpha=/zero.noslash./zero.noslash5, showSignificant=TRUE, features=FALSE, xlimits = c(-2/zero.noslash, 2/zero.noslash), ...)
## S4 method for signature CuffData
csVolcanoMatrix(object, alpha=/zero.noslash./zero.noslash5,xlimits=c(-2/zero.noslash,2/zero.noslash),mapping=aes(),...)
```

**Arguments**

- **object**: An object of class CuffData, CuffFeatureSet, or CuffGeneSet
- **x**: Sample name from 'samples' table for comparison
- **y**: Sample name from 'samples' table for comparison
- **alpha**: Provide an alpha cutoff for visualizing significant genes
- **showSignificant**: A logical value whether or not to distinguish between significant features or not (by color).
- **features**: Will include all fields from 'features' slot in returned ggplot object. Useful for further manipulations of plot object using feature-level attributes (e.g. gene_type, class_code, etc)
- **xlimits**: Set boundaries for x limits to avoid infinity plotting errors. [Default c(-20,20)]
- **mapping**: Passthrough argument for ggplot aesthetics. Can be ignored completely.
- **...**: Additional arguments
Details
This creates a 'volcano' plot of fold change vs. significance for a pairwise comparison of genes or features across two different samples.

Value
A ggplot2 plot object

Note
None

Author(s)
Loyal A. Goff

References
None.

Examples
a<-readCufflinks(system.file("extdata", package="cummeRbund")) #Create CuffSet object
genes<-a@genes #Create cuffData object for all genes
v<-csVolcano(genes,"hESC","Fibroblasts") # Volcano plot of all genes for conditions x=hESC and y=Fibroblast
v #print plot

CuffData-class

Class "CuffData"

Description
A 'pointer' class for all information (FPKM, annotation, differential expression) for a given feature type (genes, isoforms, TSS, CDS). The methods for this function communicate directly with the SQL backend to present data to the user.

Objects from the Class
Objects can be created by calls of the form new("CuffData", DB, tables, filters, type, idField, ...).

Slots
DB: Object of class "SQLiteConnection" ~~
tables: Object of class "list" ~~
filters: Object of class "list" ~~
type: Object of class "character" ~~
idField: Object of class "character" ~~
Methods

```
dim signature(x = "CuffData"): ...
getFeatures signature(object = "CuffData"): ...
DB signature(object = "CuffData"): Accessor for @DB slot
diffTable signature(object = "CuffData"): Create a Full table (wide format) of differential expression information for all pairwise comparisons
makeRnk signature(object = "CuffData"): Internal method to create .rnk file. Should not be called directly
annotation signature(object="CuffData"): Access annotation data
```

Note

None

Author(s)

Loyal A. Goff

References

None

See Also

None

Examples

```
showClass("CuffData")
```

---

CuffDist-class  
Class "CuffDist"

Description

A 'pointer' class to information relative to the distribution-level tests (promoters, splicing, and relative CDS usage)

Objects from the Class

Objects can be created by calls of the form `new("CuffDist", DB, table, type, idField, ...)`. 
Slots

DB: Object of class "SQLiteConnection" ~
table: Object of class "character" ~
type: Object of class "character" ~
idField: Object of class "character" ~

Methods

\texttt{dim} signature(x = "CuffDist"): ... 
\texttt{samples} signature(x = "CuffDist"): ... 
\texttt{DB} signature(object = "CuffDist"): Accessor for @DB slot

Note

None

Author(s)

Loyal A. Goff

References

None

See Also

None

Examples

\texttt{showClass("CuffDist")}

Description

A 'data' container class for all FPKM, annotation, and differential expression data for a single feature (gene, isoform, TSS, or CDS).

Objects from the Class

Objects can be created by calls of the form \texttt{new("CuffFeature", annotation, fpkm, diff, ...).}
**CuffFeature-class**

**Slots**

- annotation: Object of class "data.frame"
- fpkm: Object of class "data.frame"
- diff: Object of class "data.frame"
- repFpkm: Object of class "data.frame"
- count: Object of class "data.frame"
- genome: Object of class "character"

**Methods**

- fpkmMatrix signature(object="CuffFeature"): ...
- repFpkmMatrix signature(object = "CuffFeature"): ...
- length signature(x = "CuffFeature"): ...

**Accessors**

- annotation signature(object="CuffFeature"): Access @annotation slot
- diffData signature(object="CuffFeature"): Access @diff slot
- samples signature(object="CuffFeature"): Get vector of samples

**Note**

'CuffGene' is a superclass of 'CuffFeature' that links gene information for a given gene with all isoform-, TSS-, and CDS-level data for the given gene.

**Author(s)**

Loyal A. Goff

**References**

None

**See Also**

CuffGene

**Examples**

showClass("CuffFeature")
Class \textit{CuffFeatureSet} \hfill

\begin{itemize}
    \item \textbf{Description}
    \begin{itemize}
        \item A `data` container class for all FPKM, annotation, and differential expression data for a set of features (genes, isoforms, TSS, CDS).
    \end{itemize}

    \item \textbf{Objects from the Class}
    \begin{itemize}
        \item Objects can be created by calls of the form \texttt{new("CuffFeatureSet", annotation, fpkm, diff, \ldots}).
    \end{itemize}

    \item \textbf{Slots}
    \begin{itemize}
        \item \texttt{annotation}: Object of class \texttt{"data.frame"} \\
        \item \texttt{fpkm}: Object of class \texttt{"data.frame"} \\
        \item \texttt{diff}: Object of class \texttt{"data.frame"} \\
        \item \texttt{repFpkm}: Object of class \texttt{"data.frame"} \\
        \item \texttt{count}: Object of class \texttt{"data.frame"} \\
        \item \texttt{genome}: Object of class \texttt{"character"}
    \end{itemize}

    \item \textbf{Methods}
    \begin{itemize}
        \item \texttt{diffData signature(object = "CuffFeatureSet")}: \ldots \\
        \item \texttt{featureNames signature(object = "CuffFeatureSet")}: \ldots \\
        \item \texttt{features signature(object = "CuffFeatureSet")}: \ldots \\
        \item \texttt{fpkmMatrix signature(object = "CuffFeatureSet")}: \ldots \\
        \item \texttt{repFpkmMatrix signature(object = "CuffFeatureSet")}: \ldots \\
        \item \texttt{countMatrix signature(object = "CuffFeatureSet")}: \ldots \\
        \item \texttt{samples signature(object = "CuffFeatureSet")}: \ldots \\
        \item \texttt{length signature(object = "CuffFeatureSet")}: \ldots \\
    \end{itemize}

    \item \textbf{Accessors}
    \begin{itemize}
        \item \texttt{annotation signature(object="CuffFeatureSet")}: Access @annotation slot
    \end{itemize}

    \item \textbf{Note}
    \begin{itemize}
        \item None.
    \end{itemize}

    \item \textbf{Author(s)}
    \begin{itemize}
        \item Loyal A. Goff
    \end{itemize}
\end{itemize}
CuffGene-class

References
None.

See Also
CuffGeneSet

Examples
showClass("CuffFeatureSet")

---

CuffGene-class

Class "CuffGene"

Description
A 'data' container class for all FPKM, annotation, and differential expression Data (as well as for all linked features) for a given gene.

Objects from the Class
Objects can be created by calls of the form new("CuffGene", id, isoforms, TSS, CDS, promoters, splicing, relCDS, annotation, fpkm, diff, ...).

Slots
id: Object of class "character" ~
isoforms: Object of class "CuffFeature" ~
TSS: Object of class "CuffFeature" ~
CDS: Object of class "CuffFeature" ~
promoters: Object of class "CuffFeature" ~
relCDS: Object of class "CuffFeature" ~
splicing: Object of class "CuffFeature" ~
annotation: Object of class "data.frame" ~
genome: Object of class "character" ~
fpkm: Object of class "data.frame" ~
diff: Object of class "data.frame" ~
features: Object of class "data.frame" ~

Extends
Class "CuffFeature", directly.
Methods

- **length** signature(object="CuffFeature"): Part of length validation (internal use only)
- **makeGeneRegionTrack** signature(object="CuffFeature"): Creates a GeneRegionTrack object (see package Gviz) from a CuffGene object.
- **genePlot** signature(object="CuffFeature"): Internal use only.
- **csPie** signature(object="CuffGene"): Allows for visualization of relative isoform proportion as a pie chart by condition (or optionally as stacked bar charts by adding + coord_cartesian())

Accessors

- **genes** signature(object = "CuffGene"): Access @genes slot
- **isoforms** signature(object = "CuffGene"): Access @isoforms slot
- **TSS** signature(object = "CuffGene"): Access @TSS slot
- **CDS** signature(object = "CuffGene"): Access @CDS slot
- **promoters** signature(object = "CuffGene"): Access @CDS slot
- **relCDS** signature(object = "CuffGene"): Access @CDS slot
- **splicing** signature(object = "CuffGene"): Access @CDS slot
- **features** signature(object = "CuffGene"): Access @features slot

Note

None.

Author(s)

Loyal A. Goff

References

None.

See Also

- CuffFeature

Examples

showClass("CuffGene")
CuffGeneSet-class

Class "CuffGeneSet"

Description
A `data` container class for all FPKM, annotation, and differential expression data (an associated features) for a given set of genes.

Objects from the Class
Objects can be created by calls of the form `new("CuffGeneSet", annotation, fpkm, diff, ...)`. 

Slots
- ids: Object of class "character" ~~
- isoforms: Object of class "CuffFeatureSet" ~~
- TSS: Object of class "CuffFeatureSet" ~~
- CDS: Object of class "CuffFeatureSet" ~~
- promoters: Object of class "CuffFeatureSet" ~~
- relCDS: Object of class "CuffFeatureSet" ~~
- splicing: Object of class "CuffFeatureSet" ~~
- annotation: Object of class "data.frame" ~~
- fpkm: Object of class "data.frame" ~~
- diff: Object of class "data.frame" ~~

Extends
Class "CuffFeatureSet", directly.

Methods
No methods defined with class "CuffGeneSet" in the signature.

Accessors
- genes signature(object = "CuffGeneSet"): Access @genes slot
- isoforms signature(object = "CuffGeneSet"): Access @isoforms slot
- TSS signature(object = "CuffGeneSet"): Access @TSS slot
- CDS signature(object = "CuffGeneSet"): Access @CDS slot
- promoters signature(object = "CuffGeneSet"): Access @promoters slot
- relCDS signature(object = "CuffGeneSet"): Access @relCDS slot
- splicing signature(object = "CuffGeneSet"): Access @splicing slot
Note
None.

Author(s)
Loyal A. Goff

References
None.

See Also
CuffFeatureSet

Examples
showClass("CuffGeneSet")

CuffSet-class  Class "CuffSet"

Description
A 'pointer' class to connect to, and retrieve data from the SQLite backend database.

Objects from the Class
Objects can be created by calls of the form new("CuffSet", DB, conditions, genes, isoforms, TSS, CDS, promoters, CuffSet-class.
Available methods are primary accessors to retrieve CuffGeneSet or CuffGene objects for manipu-
lation.

Slots
DB: Object of class "SQLiteConnection" ~~
conditions: Object of class "data.frame" ~~
genres: Object of class "CuffData" ~~
isofoms: Object of class "CuffData" ~~
phenoData: Object of class "data.frame" ~~
TSS: Object of class "CuffData" ~~
CDS: Object of class "CuffData" ~~
promoters: Object of class "CuffDist" ~~
splicing: Object of class "CuffDist" ~~
relCDS: Object of class "CuffDist" ~~
conditions: Object of class "data.frame" ~~
Methods

[ signature(x = "CuffSet"): ...

Accessors

DB signature(object="CuffSet"): Access @DB slot
genes signature(object = "CuffSet"): Access @genes slot
isoforms signature(object = "CuffSet"): Access @isoforms slot
TSS signature(object = "CuffSet"): Access @TSS slot
CDS signature(object = "CuffSet"): Access @CDS slot
promoters signature(object = "CuffSet"): Access @promoters slot
splicing signature(object = "CuffSet"): Access @splicing slot
relCDS signature(object = "CuffSet"): Access @relCDS slot
varModel signature(object = "CuffSet"): Access varModel info

Note

None.

Author(s)

Loyal A. Goff

References

None.

See Also

None.

Examples

showClass("CuffSet")
diffData  

Differential comparison data

Description
An accessor method to retrieve differential expression data from a 'CuffData', 'CuffFeatureSet', or 'CuffFeature' object.

Usage
## S4 method for signature CuffData
diffData(object, x, y, features=FALSE)
## S4 method for signature CuffData
diffTable(object, logCutoffValue=99999)

Arguments
- `object`: An object of class ('CuffData' or 'CuffFeatureSet')
- `x`: Optional, if `x` and `y` are both missing, data for all pairwise differential comparisons are returned, otherwise if `x` and `y` are sample names from the 'samples' table, than only differential data pertaining to those two samples are returned.
- `y`: See 'x'
- `features`: A logical value that returns all feature-level data as part of data.frame when true. object must be of class 'CuffData'.
- `logCutoffValue`: Cutoff value for FC estimates to convert to [-]Inf values. Should never really be needed...
- ... Additional arguments.

Details
None

Value
A data.frame object

Note
None

Author(s)
Loyal A. Goff

References
None
Examples

```r
data(sampleData)
diff <- diffData(sampleGeneSet) # returns a dataframe of differential expression data from sample CuffGeneSet object.
```

---

**Dimensionality Reduction**

*Dimensionality reduction utilities*

---

**Description**

Dimensionality reduction plots for feature selection and extraction for cummeRbund

**Usage**

```r
## S4 method for signature CuffData
MDSplot(object, replicates = FALSE, logMode = TRUE, pseudocount = 1.0)

## S4 method for signature CuffData
PCAplot(object, x = "PC1", y = "PC2", replicates = FALSE, pseudocount = 1.0, scale = TRUE, showPoints = TRUE, ...)```

**Arguments**

- `object`: The output of class CuffData from which to draw expression estimates. (e.g. `genes(cuff)`)
- `x`: For PCAplot, indicates which principal component is to be presented on the x-axis (e.g. "PC1","PC2","PC3", etc)
- `y`: See `x`.
- `pseudocount`: Value added to FPKM to avoid log transformation issues.
- `logMode`: Logical value whether or not to use log-transformed expression estimates (default: TRUE)
- `replicates`: A logical value to indicate whether or not individual replicate expression estimates will be used.
- `scale`: For PCAplot, a logical value passed directly to `prcomp`.
- `showPoints`: For PCAplot, a logical value whether or not to display individual gene values on final PCA plot.
- `...`: Additional passthrough arguments (may not be fully implemented yet).

**Details**

These methods attempt to project a matrix of expression estimates across conditions and/or replicates onto a smaller number of dimensions for feature selection, feature extraction, and can also be useful for outlier detection.

**Value**

A ggplot2 object.
Note
None.

Author(s)
Loyal A. Goff

References
None.

Examples

cuff<-readCufflinks(system.file("extdata", package="cummeRbund")) #Create CuffSet object from sample data
p<-PCAplot(genes(cuff),x="PC2",y="PC3",replicates=TRUE)
m<-MDSplot(genes(cuff),replicates=TRUE)
p #Render PCA plot
m #Render MDS plot

dispersionPlot

Mean count vs dispersion plot

Description
A scatter plot comparing the mean counts against the estimated dispersion for a given level of features from a cuffdiff run.

Usage

## S4 method for signature CuffData
dispersionPlot(object)
## S4 method for signature CuffSet
dispersionPlot(object)

Arguments

object An object of class ('CuffData')

Details
None

Value
ggplot object with geom_point layer
distValues

Note
None

Author(s)
Loyal A. Goff

References
None

Examples
a<-readCufflinks(system.file("extdata", package="cummeRbund")) #Create CuffSet object from sample data
genes<-genes(a) #Create CuffData object for all genes
d<-dispersionPlot(genes) #Create plot object
d #render plot object

Description
Returns a data.frame of distribution-level test values from a CuffDist object (@promoters, @splicing, @relCDS)

Usage
## S4 method for signature CuffDist
distValues(object)

Arguments
  object An object of class 'CuffDist'
  ... Additional arguments to distValues

Details
None

Value
Returns a data.frame of distribution-level test values.

Note
None
Exploratory Analysis

Methods for Exploratory Analysis in cummeRbund

Description

Exploratory analysis methods for cummeRbund RNA-Seq data.

Usage

```r
## S4 method for signature CuffData
csNMF(object,k,logMode=T,pseudocount=1,maxiter=1000,replicates=FALSE,fullnames=FALSE)
## S4 method for signature CuffFeatureSet
csNMF(object,k,logMode=T,pseudocount=1,maxiter=1000,replicates=FALSE,fullnames=FALSE)
```

Arguments

- `object`: The output of class CuffData or CuffFeatureSet from which to draw expression estimates. (e.g. genes(cuff) or custom feature set via getGenes() or getFeatures())

- `k`: rank value for factorization

- `logMode`: Logical value whether or not to use log-transformed FPKM values. [Default: TRUE]

- `pseudocount`: Value added to FPKM to avoid log transformation issues.

- `maxiter`: Maximum number of iterations for factorization [Default: 1000]

- `replicates`: A logical value to indicate whether or not individual replicate expression estimates will be used.

- `fullnames`: Logical passthrough value to fpkmMatrix whether or not to concatenate gene_short_name with tracking_id. [Default: FALSE]
expressionBarplot

Details

csNMF is a convenience method to invoke the nmmf() method from package:NMFN. This performs non-negative matrix factorization on the provided data and can be useful for many downstream applications.

Value

csNMF returns W, H - decomposed matrices of input FPKM values. (See package:NMFN for details)

Note

None.

Author(s)

Loyal A. Goff

References

None.

Examples

data(sampleData)
csNMF(sampleGeneSet,4)

details

expressionBarplot  Barplot

Description

A barplot of FPKM values with confidence intervals for a given gene, set of genes, or features of a gene (e.g. isoforms, TSS, CDS, etc).

Usage

## S4 method for signature CuffFeatureSet
eexpressionBarplot(object, logMode=TRUE, pseudocount=1.0, showErrorbars=TRUE, showStatus=TRUE, replicates=FALSE, ...)
expressionBarplot

Arguments

logMode  A logical value whether or not to draw y-axis on log10 scale. Default = FALSE.
pseudocount  Numerical value added to each FPKM during log-transformation to avoid errors.
showErrorbars  A logical value whether or not to draw error bars. Default = TRUE
showStatus  A logical value whether or not to draw visual queues for quantification status of a given gene:condition. Default = TRUE
replicates  A logical value whether or not to plot individual replicates or aggregate condition values.
...  Additional arguments.

Details

None

Value

A ggplot2 plot object

Note

Need to implement logMode and features for this plotting method.

Author(s)

Loyal A. Goff

References

None

Examples

data(sampleData)
PINK1 # sample Cufffeature object
expressionBarplot(PINK1) # Barplot of PINK1 FPKM values
expressionBarplot(PINK1@isoforms) # Barplot of PINK1 FPKM values faceted by isoforms
expressionPlot

Description
A line plot (optionally with confidence intervals) detailing FPKM expression levels across conditions for a given gene(s) or feature(s).

Usage
```r
## S4 method for signature CuffFeature
dexpressionPlot(object, logMode=FALSE, pseudocount=1.0, drawSummary=FALSE, sumFun=mean_cl_boot, showErrorbars=TRUE, showStatus=TRUE, replicates=FALSE, facet = TRUE,...)
```

Arguments
- `object`: An object of class (‘CuffFeature’ or ’CuffGene’)
- `logMode`: A logical value to draw y-axis (FPKM) on log-10 scale. Default = FALSE.
- `pseudocount`: A numeric value added to FPKM to avoid errors on log-10 transformation.
- `drawSummary`: A logical value. Draws a ‘summary’ line with mean FPKM values for each condition.
- `sumFun`: Function used to determine values for summary line. Default = mean_cl_boot
- `showErrorbars`: A logical value whether or not to draw error bars.
- `showStatus`: A logical value whether or not to draw visual queues for quantification status of a given gene:condition. Default = TRUE
- `replicates`: A logical value whether or not to plot individual replicates or aggregate condition values.
- `facet`: A logical value whether or not to facet the plot by feature id (default=TRUE).
- `...`: Additional arguments

Details
None

Value
A ggplot2 plot object

Note
None

Author(s)
Loyal A. Goff
References

None

Examples

data(sampleData)
PINK1 # sample CuffFeature object
eexpressionPlot(PINK1) #Line plot of PINK1 FPKM values
eexpressionPlot(PINK1@isoforms) #Line plot of PINK1 FPKM values faceted by isoforms

<table>
<thead>
<tr>
<th>featureNames</th>
<th>Feature names</th>
</tr>
</thead>
</table>

Description

Retrive a vector of feature names from a ‘CuffData’ or ‘CuffFeatureSet’ object

Usage

## S4 method for signature CuffData
featureNames(object)

Arguments

object An object of class (‘CuffData’ or ‘CuffFeatureSet’)

Details

None

Value

A list of feature names

Note

None

Author(s)

Loyal A. Goff

References

None

Examples

data(sampleData)
featureNames(sampleGeneSet)
# features

Returns a data frame of features from a CuffGene object

## Usage

```r
## S4 method for signature CuffGene
features(object)
```

## Arguments

- `object`: An object of class (`'CuffGene'`)

## Details

None

## Value

A data.frame of feature-level information

## Note

None

## Author(s)

Loyal A. Goff

## References

None

## Examples

```r
data(sampleData)
features(PINK1)
```
Description

A helper function to retrieve the gene_ids given a 'lookup' value (e.g. gene_short_name, isoform_id, etc). Utility to search for gene_id and gene_short_name given a single 'query' string (e.g. query='pink1' will return all genes with 'pink1' (case-insensitive) in the gene_short_name field.

Usage

## S4 method for signature CuffSet
findGene(object, query)

Arguments

- object: An object of class 'CuffSet' (Primary 'pointer' object for Cufflinks data).
- query: A character string for which you would like to retrieve corresponding gene_id values.

Details

None.

Value

Returns a data.frame of gene_id and gene_short_name values corresponding to genes from which 'query' matches.

Note

Right now, this does not return an error if it cannot find a gene. (this is probably a bad thing...)

Author(s)

Loyal A. Goff

References

None.

Examples

cuff<-readCufflinks(system.file("extdata", package="cummeRbund")) #Read cufflinks data and create master CuffSet object
myQuery<-"pink1"
findGene(cuff,myQuery) # Retrieve gene_id values for any genes matching pink1
findSimilar

Description

Returns a CuffGeneSet containing n genes with the most similar expression profiles to gene/profile x.

Usage

## S4 method for signature CuffSet
findSimilar(object, x, n, distThresh, returnGeneSet=TRUE,...)

Arguments

- **object**: A object of class `CuffSet`
- **x**: A 'gene_id' or 'gene_short_name' from which to look up an expression profile OR a vector of expression values to compare all genes (vector must have same length and order of 'samples')
- **n**: Number of similar genes to return
- **distThresh**: A thresholding value on which to filter results based on JS-distance (e.g. A distThresh of 1.0 will return all genes, 0.0 will return those genes with 'perfect identity' to the gene of interest.)
- **returnGeneSet**: A logical value whether to return a CuffGeneSet object [default] or a distance-ranked data frame of similar genes. The latter is useful if you want to explore the returned list based on distances.
- **...**: Additional arguments to fpkmMatrix call within findSimilar (e.g. fullnames=T)

Details

By default, returns a CuffGeneSet object with n similar genes. This may change in the future.

Value

A CuffGeneSet object of n most similar genes to x.

Note

None

Author(s)

Loyal A. Goff

References

None
Examples

```r
a <- readCufflinks(system.file("extdata", package="cummeRbund"))
mySimilarGenes <- findSimilar(a, "PINK1", 10)
```

---

### fpkm-methods

**Retrieve FPKM values**

**Description**

Returns a data.frame from @FPKM slot

**Details**

Returns a data.frame of FPKM values.

**Value**

A data.frame of FPKM-level values for a set of features.

**Methods**

- `signature(object = "CuffData")`
- `signature(object = "CuffFeature")`
- `signature(object = "CuffFeatureSet")`

**Note**

None

**Author(s)**

Loyal A. Goff

**References**

None

**Examples**

```r
data(sampleData)
fpkm(PINK1)
```
Description

Retrieve FPKM values as gene by condition (fpkmMatrix) or gene by replicate (repFpkmMatrix) matrix.

Usage

```r
## S4 method for signature CuffData
fpkmMatrix(object, fullnames = FALSE, sampleIdList)
## S4 method for signature CuffData
repFpkmMatrix(object, fullnames = FALSE, repIdList)
```

Arguments

- `object`: An object of class (`'CuffData'`, `'CuffFeatureSet'`, `'CuffGeneSet'`, `'CuffGene'`, or `'CuffFeature'`)
- `fullnames`: A logical value whether or not to concatenate `gene_short_name` and `tracking_id`
  values (easier to read labels)
- `sampleIdList`: A vector of sample names to subset the resulting matrix.
- `repIdList`: A vector of sample names to subset the resulting matrix.

Details

None.

Value

A feature x condition matrix of FPKM values.

Note

None.

Author(s)

Loyal A. Goff

References

None.
getFeatures

Examples

data(sampleData)
fpkmMatrix(sampleGeneSet)
repFpkmMatrix(sampleGeneSet)

getFeatures

getFeatures

Description

Primary accessor from a CuffSet object to retrieve all related information for >1 (MANY) given FEATURES, indexed by tracking id.

Usage

```r
## S4 method for signature CuffSet
getFeatures(object, featureIdList, sampleIdList=NULL, level=isoforms)
```

Arguments

- `object`: An object of class 'CuffSet' (Primary 'pointer' object for Cufflinks data).
- `featureIdList`: A vector of 'isoform_id', 'TSS_group_id', or 'CDS_id' to identify which features for which you would like to retrieve all information.
- `sampleIdList`: A vector of sample names used to subset or re-order samples in returned object.
- `level`: Feature level to be queried for significance (must be one of 'isoforms', 'TSS', 'CDS')

Details

None.

Value

Returns a CuffFeatureSet object containing all related information for a given set of tracking_id values.

Note

Right now, this does not return an error if it cannot find a gene. (this is probably a bad thing...)

Author(s)

Loyal A. Goff

References

None.
Examples

cuff<-readCufflinks(system.file("extdata", package="cummeRbund")) # Read cufflinks data and create master CuffSet object
sample.isoform.ids<-sample(featureNames(isoforms(cuff)),10)
myGene<-getFeatures(cuff,sample.isoform.ids) # Retrieve all information for a set of 10 sampled features.

getGene

Description

Primary accessor from a CuffSet object to retrieve all related information for one given gene, indexed by gene_id or gene_short_name.

Usage

## S4 method for signature CuffSet
getGene(object, geneId, sampleIdList=NULL)

Arguments

object An object of class 'CuffSet' (Primary 'pointer' object for Cufflinks data).
geneId A character string to identify which gene for which you would like to retrieve all information.
sampleIdList A vector of sample names used to subset or re-order samples in returned object

Details

None.

Value

Returns a CuffGene object containing all related information for a given gene_id or gene_short_name

Note

Right now, this does not return an error if it cannot find a gene. (this is probably a bad thing...)

Author(s)

Loyal A. Goff

References

None.

Examples

a<-readCufflinks(system.file("extdata", package="cummeRbund")) # Read cufflinks data and create master CuffSet object
myGene<-getGene(a,"PINK1") # Retrieve all information for gene "PINK1"
getGeneId

Description

A helper function to retrieve the gene_ids for a given list of feature ids (e.g. isoform_ids, tss_group_ids, or CDS_ids). This should not be called directly by the user.

Usage

## S4 method for signature CuffSet
getGeneId(object, idList)

Arguments

object
An object of class 'CuffSet' (Primary 'pointer' object for Cufflinks data).
idList
A character string to identify the identifiers for which you would like to retrieve corresponding gene_id values.

Details

None.

Value

Returns a vector of gene_id values corresponding to genes from which idList are sub-features.

Note

Right now, this does not return an error if it cannot find a gene. (this is probably a bad thing...)

Author(s)

Loyal A. Goff

References

None.

Examples

cuff<-readCufflinks(system.file("extdata", package="cummeRbund")) #Read cufflinks data and create master CuffSet object
sampleFeatureIds<-sample(featureNames(isoforms(cuff)),10)
correspondingGeneIds<-getGeneId(cuff,sampleFeatureIds) # Retrieve gene_id values for parent genes of sampleFeatureIds.
Description

Primary accessor from a CuffSet object to retrieve all related information for >1 (MANY) given genes, indexed by gene_id or gene_short_name.

Usage

```r
## S4 method for signature CuffSet
genes(object, geneIdList, sampleIdList=NULL)
```

Arguments

- **object**: An object of class 'CuffSet' (Primary 'pointer' object for Cufflinks data).
- **geneIdList**: A vector of gene_ids or gene_short_name to identify which genes for which you would like to retrieve all information.
- **sampleIdList**: A vector of sample names used to subset or re-order samples in returned object.

Details

None.

Value

Returns a CuffGeneSet object containing all related information for a given set of gene_id or gene_short_name values.

Note

Right now, this does not return an error if it cannot find a gene. (this is probably a bad thing...)

Author(s)

Loyal A. Goff

References

None.

Examples

```r
a<-readCufflinks(system.file("extdata", package="cummeRbund")) # Read cufflinks data and create master CuffSet object
data(sampleData)
sampleIDs
myGene<-getGenes(a,sampleIDs) # Retrieve all information for a set of 20 sample genes.
```
getLevels

Description

Returns a list of samples as levels. This should not be called directly by user.

Usage

```r
## S4 method for signature CuffData
getLevels(object)
```

Arguments

- `object` An object of class 'CuffData' or 'CuffFeatureSet' or 'CuffFeature'

Details

For internal usage only.

Value

A vector of sample names as factors.

Note

None.

Author(s)

Loyal A. Goff

References

None.
**Description**

Returns a list of replicate samples as levels. This should not be called directly by user.

**Usage**

```r
## S4 method for signature CuffData
getRepLevels(object)
```

**Arguments**

- `object` An object of class 'CuffSet' or 'CuffData'

**Details**

For internal usage only.

**Value**

A vector of replicate names as factors.

**Note**

None.

**Author(s)**

Loyal A. Goff

**References**

None.
Description

Returns the identifiers of significant genes in a vector format.

Usage

```r
## S4 method for signature CuffSet
getSig(object, x, y, alpha=/zero.noslash./zero.noslash5, level=genes, method="BH", useCuffMTC=FALSE)
```

Arguments

- **object**: A CuffSet object (e.g. cuff)
- **x**: Optional argument to restrict significance results to one pairwise comparison. Must be used with a `y` argument to specify the other half of the pair.
- **y**: See x.
- **alpha**: An alpha value by which to filter multiple-testing corrected q-values to determine significance
- **level**: Feature level to be queried for significance (must be one of c('genes', 'isoforms', 'TSS', 'CDS'))
- **method**: Multiple testing method to be used for correction. (default: "BH")
- **useCuffMTC**: Logical vector whether or not to use the multiple-testing corrected q-values from the cuffdiff analysis directly, or calculate new q-values from a subset of tests.

Details

This is a convenience function to quickly retrieve vectors of identifiers for genes or features that were determined to be significantly regulated between conditions by cuffdiff. This function only returns tracking IDs that correspond to tests with an 'OK' status from cuffdiff, NOTEST values are ignored. By default getSig returns a vector of IDs for all pairwise comparisons together. If you specify an `x` AND `y` values as sample names, then only the features that are significant in that particular pairwise comparison are reported, after appropriate multiple testing correction of output p-values.

Value

A vector of feature IDs.

Note

None.

Author(s)

Loyal A. Goff
getSigTable

References

None.

Examples

```r
a <- readCufflinks(system.file("extdata", package="cummeRbund")) # Read cufflinks data in sample directory and create CuffSet object
mySig <- getSig(a, x = hESC, y = Fibroblasts, alpha = 0.05, level = genes)
head(mySig)
```

Description

Returns the identifiers of significant genes in a test table-like format.

Usage

```r
## S4 method for signature CuffSet
getSigTable(object, alpha = 0.05, level = genes)
```

Arguments

- **object**: A CuffSet object (e.g. cuff)
- **alpha**: An alpha value by which to filter multiple-testing corrected q-values to determine significance
- **level**: Feature level to be queried for significance (must be one of c('genes', 'isoforms', 'TSS', 'CDS')

Details

This is a convenience function to quickly retrieve lists of identifiers for genes or features that were determined to be significantly regulated between conditions by cuffdiff. This function only returns tracking IDs that correspond to tests with an 'OK' status from cuffdiff. NOTEST values are ignored or reported as NA. By default getSig returns a table of genes x conditions, where the column names represent the pairwise comparisons from the cuffdiff analysis. The values in the table are 1 for features that are significant for this comparison and 0 for genes that are not, any failed tests are reported as <NA>. Only includes the features that are significant in at least one comparison.

Value

A data.frame of pairwise test results.

Note

None.
Author(s)
Loyal A. Goff

References
None.

Examples

a<-readCufflinks(system.file("extdata", package="cummeRbund")) #Read cufflinks data in sample directory and create
dist object
mySigTable<-getSigTable(a, alpha=0.05, level=genes)
head(mySigTable)

<table>
<thead>
<tr>
<th>JSdist</th>
<th>Jensen-Shannon distance on columns</th>
</tr>
</thead>
</table>

Description

JSdist takes a matrix of expression probabilities (calculated directly or output from makeprobs()) and returns a dist object of the pairwise Jensen-Shannon distances between columns.

Usage

JSdist(mat, ...)

Arguments

mat
A matrix of expression probabilities (e.g. from makeprobs())

Details

Returns pairwise Jensen-Shannon distance (in the form of a dist object) for a matrix of probabilities (by column)

Value

A dist object of pairwise J-S distances between columns.

Note
None.

Author(s)
Loyal A. Goff
References
None

Examples

```r
mat <- matrix(sample(1:50,50),10)
probs <- makeprobs(mat)
js.distance <- JSdist(probs)
```

---

**JSdistFromP**

*Jensen-Shannon distance on rows from a pre-defined vector of probabilities*

Description

JSdist takes a matrix of expression probabilities (calculated directly or output from makeprobs()) and returns a matrix of Jensen-Shannon distances between individual rows and a specific vector of probabilities (q)

Usage

`JSdistFromP(mat, q)`

Arguments

- `mat` A matrix of expression probabilities (e.g. from makeprobs())
- `q` A vector of expression probabilities.

Details

Returns Jensen-Shannon distance for each row of a matrix of probabilities against a provided probability distribution (q)

Value

A vector of JS distances

Note

None

Author(s)

Loyal A. Goff
References

None

Examples

```r
mat <- matrix(sample(1:5, 10), 1)
q <- c(1, 4, 72, 8, 19)
q <- q / sum(q)
js.distance <- JSdistFromP(mat, q)
```
Examples

```r
p <- sample(1:5000, 20)
q <- sample(1:5000, 20)
p <- makeprobsvec(p)
q <- makeprobsvec(q)
JSdistVec(p, q)
```

### Description

This function takes a matrix of expression values (must be greater than 0) and returns a matrix of probabilities by column. This is a required transformation for the Jensen-Shannon distance which is a metric that operates on probabilities.

### Usage

```r
makeprobs(a)
```

### Arguments

- **a**
  - A matrix of expression values (values must be greater than 0).

### Details

To make a matrix of probabilities by row, use `t()` to transpose prior to calling `makeprobs`.

### Value

- A matrix of expression probabilities by column.

### Note

None

### Author(s)

Loyal A. Goff

### References

None

### Examples

```r
myMat <- matrix(sample(1:50, 50), 10)
probs <- makeprobs(myMat)
```
makeprobsvec

Description
Sums a vector of numerics and divides by the sum

Usage
makeprobsvec(p)

Arguments
p A vector of numerics

Details
None

Value
A vector of probabilities

Note
Should not be called directly by user.

Author(s)
Loyal A. Goff

References
None

Examples
p<-sample(1:5000,20)
makeprobsvec(p)
Description

Creates an M vs A plot (Avg intensity vs log ratio) for a given pair of conditions across all fpkms

Usage

## S4 method for signature CuffData
MAplot(object, x, y, logMode = T, pseudocount = 1, smooth = FALSE, useCount = FALSE)

Arguments

object  An object of class 'CuffData'.
x  Sample name from 'samples' table for comparison
y  Sample name from 'samples' table for comparison
logMode  A logical argument to log10-transform FPKM values prior to plotting.
pseudocount  Value to be added to FPKM for appropriate log transformation and clustering. (Avoids zero-based errors)
smooth  Logical argument whether or not to draw a smoothed line fit through data.
useCount  Logical argument whether or not to use mean counts instead of FPKM values.

Details

None

Value

Returns a ggplot MvsA plot object.

Note

None

Author(s)

Loyal A. Goff and Cole Trapnell

References

None.
Examples

```r
a <- readCufflinks(system.file("extdata", package="cummeRbund")) # Create CuffSet object from sample data
genes <- a@genes # Create CuffData object for all genes
d <- MAplot(genes, hESC, Fibroblasts) # Create csDensity plot
d # Render plot
```

Description

A sample 'CuffGene' dataset

Usage

```r
data(sampleData)
```

Format

PINK1 is a CuffGene object (extends CuffFeature) with all sample gene-, isoform-, TSS-, and CDS-level data for the gene 'PINK1'.

Details

Sample CuffGene data for gene 'PINK1'

Source

None

References

None

Examples

```r
data(sampleData)
PINK1
```
**Description**

A collection of ggplot2 visualizations for quality control assessment of cuffdiff output.

- fpkmSCVPlot: A measure of cross-replicate variability, the squared coefficient of variation is a normalized measure of variance between empirical replicate FPKM values per condition, across the range of FPKM estimates.

**Usage**

```r
## S4 method for signature CuffData
fpkmSCVPlot(object, FPKMLowerBound = 1, showPool = FALSE)
```

**Arguments**

- `object`: An object of class CuffData.
- `FPKMLowerBound`: A lower limit cutoff for FPKM values from which a fit of squared Coefficient of variation (default: 1)
- `showPool`: Logical argument whether to display variability across all replicates independent of condition (TRUE) or the cross-replicate variability for each condition (FALSE)

**Details**

None

**Value**

A ggplot2 plot object with a geom_box layer.

**Note**

None

**Author(s)**

Loyal A. Goff

**References**

None
Examples

```r
a <- readCufflinks(system.file("extdata", package="cummeRbund")) # Read cufflinks data and create CuffSet object
genes <- a@genes # CuffData object for all genes
csBoxplot(genes)
```

readCufflinks

**Description**

This initializes the backend SQLite table and provides a DB connection for all downstream data analysis.

**Usage**

```r
readCufflinks(dir = getwd(), dbFile = "cuffData.db", gtfFile = NULL,
runInfoFile = "run.info", repTableFile = "read_groups.info",
geneFPKM = "genes.fpkm_tracking", geneDiff = "gene_exp.diff", geneCount = "genes.count_tracking", geneRep = "genes.read_group_tracking",
isoformFPKM = "isoforms.fpkm_tracking", isoformDiff = "isoform_exp.diff", isoformCount = "isoforms.count_tracking", isoformRep = "isoforms.read_group_tracking",
TSSFPKM = "tss_groups.fpkm_tracking", TSSDiff = "tss_group_exp.diff", TSSCount = "tss_groups.count_tracking", TSSRep = "tss_groups.read_group_tracking",
CDSFPKM = "cds.fpkm_tracking", CDSExpDiff = "cds_exp.diff", CDSCount = "cds.count_tracking", CDSRep = "cds.read_group_tracking", CDSDiff = "cds.diff",
promoterFile = "promoters.diff",
splicingFile = "splicing.diff",
varModelFile = "var_model.info",
driver = "SQLite",
genome = NULL,
rebuild = FALSE, verbose = FALSE, ...)
```

**Arguments**

- `dir` Directory in which all CuffDiff output files can be located. Defaults to current working directory.
- `dbFile` Name of backend database. Default is 'cuffData.db'
- `gtfFile` Path to .gtf file used in cuffdiff analysis. This file will be parsed to retrieve transcript model information.
- `runInfoFile` run.info file
- `repTableFile` read_groups.info file
- `geneFPKM` genes.fpkm_tracking file
- `geneDiff` gene_exp.diff file
- `geneCount` genes.count_tracking file
- `geneRep` genes.read_group_tracking file
- `isoformFPKM` isoforms.fpkm_tracking file
- `isoformDiff` isoform_exp.diff file
- `isoformRep` isoform_read_group_tracking file
- `TSSFPKM` tss_groups.fpkm_tracking file
- `TSSDiff` tss_group_exp.diff file
- `TSSCount` tss_groups.count_tracking file
- `TSSRep` tss_groups.read_group_tracking file
- `CDSFPKM` cds.fpkm_tracking file
- `CDSExpDiff` cds_exp.diff file
- `CDSCount` cds.count_tracking file
- `CDSRep` cds.read_group_tracking file
- `CDSDiff` cds.diff file
- `promoterFile` promoters.diff file
- `splicingFile` splicing.diff file
- `varModelFile` var_model.info file
- `driver` "SQLite"
- `genome` NULL
- `rebuild` FALSE, verbose = FALSE, ...
Details

This is the initialization function for the cummeRbund package. It creates the SQLite backend database, populates the data, and provides a connection object for all future interactions with the dataset. Once the initial build is complete, this function will default to using the database for all future sessions. IMPORTANT: - Each R session should begin with a call to readCufflinks to re-open the connection to the database. - Should any connectivity issues to the database arise, another call to readCufflinks should create a new connection object and repair any issue. - The database can always be rebuilt (using rebuild=TRUE) from the original CuffDiff output files.

Value

A CuffSet object. A 'pointer' class that allows interaction with cufflinks/cuffdiff data via a SQLite database backend.

Note

None.

Author(s)

Loyal A. Goff
References

None.

Examples

```r
a <- readCufflinks(system.file("extdata", package="cummeRbund")) # Read cufflinks data in sample directory and create
```
Description

Returns a list of replicate names from a CuffData or CuffFeatureSet object

Usage

## S4 method for signature CuffData
replicates(object)

Arguments

object An object of class ('CuffSet', 'CuffData')

Details

None

Value

A list of replicate sample names

Note

None

Author(s)

Loyal A. Goff

References

None

See Also

None

Examples

a <- readCufflinks(system.file("extdata", package="cummeRbund")) # Create CuffSet object
replicates(a@genes)
runInfo

Retrieve run parameters and information from a CuffSet object

Description

Returns a data.frame of cuffdiff run parameters and information

Usage

## S4 method for signature CuffSet
runInfo(object)

Arguments

object An object of class ('CuffSet')

Details

None

Value

A data.frame of run parameters

Note

None

Author(s)

Loyal A. Goff

References

None

See Also

None

Examples

a<-readCufflinks(system.file("extdata", package="cummeRbund")) #Create CuffSet object
runInfo(a)
sampleGeneSet

---

sampleGeneSet

---

**Description**

A sample CuffGeneSet data set for 20 genes.

**Usage**

data(sampleData)

**Format**

sampleGeneSet is a CuffGeneSet (extends CuffFeatureSet) object containing all sample gene-, isoform-, TSS-, and CDS-level data for 20 different genes. These data were derived from a toy set of hESC-vs-iPSC-vs-Fibroblast RNA-Seq expression data.

**Details**

None

**Source**

None

**References**

None

**Examples**

data(sampleData)

---

sampleIDs

---

**Description**

A vector of gene_ids used to create `sampleGeneSet` example

**Usage**

data(sampleData)
Format

The format is: chr "sampleIDs"

Details

None

Source

None

References

None

Examples

```r
data(sampleData)
```

<table>
<thead>
<tr>
<th>samples</th>
<th>Get sample list from CuffData object</th>
</tr>
</thead>
</table>

Description

Returns a list of sample names from a CuffData or CuffFeatureSet object

Usage

```r
## S4 method for signature CuffData
samples(object)
```

Arguments

<table>
<thead>
<tr>
<th>object</th>
<th>An object of class (&quot;CuffData&quot;,&quot;CuffFeatureSet&quot;,&quot;CuffFeature&quot;)</th>
</tr>
</thead>
</table>

Details

None

Value

A list of sample names

Note

None
**Description**

Calculates the Shannon entropy for a probability distribution

**Usage**

```r
shannon.entropy(p)
```

**Arguments**

- `p` A vector of probabilities (must sum to ~1)

**Details**

None

**Value**

Returns a numeric value for the Shannon entropy of the supplied probability distribution

**Note**

None

**Author(s)**

Loyal A. Goff
References
None

Examples
x<-sample(1:500,5)
p<-x/sum(x)
shannon.entropy(p)

Description
Returns a ggplot2 plot object representing a matrix of significant features. This is a useful synopsis of all significant pairwise comparisons within the dataset.

Usage
## S4 method for signature CuffSet
sigMatrix(object,alpha=0.05,level=genes,orderByDist=FALSE)

Arguments
object An object of class CuffSet.
alpha An alpha value by which to filter multiple-testing corrected q-values to determine significance
level Feature level to be queried for significance (must be one of c('genes','isoforms','TSS','CDS'))
orderByDist Logical. If TRUE then samples are re-ordered based on JS-distance from one another (fairly useless unless you have a specific need for this).

Details
Creates a matrix plot to illustrate the number of significant features of type 'level' at a given alpha from a cuffdiff run.

Value
A ggplot2 plot object

Note
None

Author(s)
Loyal A. Goff
sigMatrix

References

None

Examples

a<-readCufflinks(system.file("extdata", package="cummeRbund")) #Create CuffSet object from sample data
d<-sigMatrix(a) #Create csDensity plot
d #Render plot
Index

*Topic **JS**
  - JSdist, 52
  - JSdistFromP, 53

*Topic **Jensen-Shannon**
  - csDistHeat, 14
  - JSdist, 52
  - JSdistFromP, 53

*Topic **classes**
  - CuffData-class, 20
  - CuffDist-class, 21
  - CuffFeature-class, 22
  - CuffFeatureSet-class, 24
  - CuffGene-class, 25
  - CuffGeneSet-class, 27
  - CuffSet-class, 28

*Topic **datasets**
  - csCluster, 9
  - csSpecificity, 18
  - PINK1, 58
  - sampleGeneSet, 65
  - sampleIDs, 65

*Topic **distance**
  - csDistHeat, 14
  - JSdist, 52
  - JSdistFromP, 53

*Topic **heatmap**
  - csDendro, 11
  - csDistHeat, 14
  - csHeatmap, 15
  - MAplot, 57

*Topic **metric**
  - JSdist, 52
  - JSdistFromP, 53

*Topic **package**
  - cummeRbund-package, 3
  - [,CuffSet-method (CuffSet-class), 28

  addFeatures, 5
  addFeatures,CuffData-method (addFeatures), 5

  addFeatures,CuffSet-method (addFeatures), 5
  annotation (CuffFeatureSet-class), 24
  annotation,CuffData-method (CuffData-class), 20
  annotation,CuffFeature-method (CuffFeature-class), 22
  annotation,CuffFeatureSet-method (CuffFeatureSet-class), 24

  CDS (CuffSet-class), 28
  CDS,CuffGene-method (CuffGene-class), 25
  CDS,CuffGeneSet-method (CuffGeneSet-class), 27
  CDS,CuffSet-method (CuffSet-class), 28
  conditions (CuffSet-class), 28
  conditions,CuffSet-method (CuffSet-class), 28
  count (count-methods), 6
  count,CuffData-method (count-methods), 6
  count,CuffFeature-method (count-methods), 6
  count,CuffFeatureSet-method (count-methods), 6
  count-methods, 6
  countMatrix, 7
  countMatrix,CuffData-method (countMatrix), 7
  countMatrix,CuffFeatureSet-method (countMatrix), 7
  csBoxplot, 8
  csBoxplot,CuffData-method (csBoxplot), 8
  csCluster, 9
  csCluster,CuffFeatureSet-method (csCluster), 9
  csClusterPlot, 10
  csDendro, 11
  csDendro,CuffData-method (csDendro), 11
  csDendro,CuffFeatureSet-method (csDendro), 11
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>csDensity</td>
<td>12</td>
</tr>
<tr>
<td>csDensity, CuffData-method (csDensity)</td>
<td>12</td>
</tr>
<tr>
<td>csDensity, CuffFeatureSet-method (csDensity)</td>
<td>12</td>
</tr>
<tr>
<td>csDistHeat</td>
<td>14</td>
</tr>
<tr>
<td>csDistHeat, CuffData-method (csDistHeat)</td>
<td>14</td>
</tr>
<tr>
<td>csDistHeat, CuffFeatureSet-method (csDistHeat)</td>
<td>14</td>
</tr>
<tr>
<td>csFoldChangeHeatmap (csHeatmap)</td>
<td>15</td>
</tr>
<tr>
<td>csFoldChangeHeatmap, CuffFeatureSet-method</td>
<td>15</td>
</tr>
<tr>
<td>csHeatmap</td>
<td>15</td>
</tr>
<tr>
<td>csHeatmap, CuffFeatureSet-method (csHeatmap)</td>
<td>15</td>
</tr>
<tr>
<td>csNMF (Exploratory Analysis)</td>
<td>34</td>
</tr>
<tr>
<td>csNMF, CuffData-method (Exploratory Analysis)</td>
<td>34</td>
</tr>
<tr>
<td>csNMF, CuffFeatureSet-method (Exploratory Analysis)</td>
<td>34</td>
</tr>
<tr>
<td>csPie (CuffGene-class)</td>
<td>25</td>
</tr>
<tr>
<td>csPie, CuffGene-method (CuffGene-class)</td>
<td>25</td>
</tr>
<tr>
<td>csScatter</td>
<td>17</td>
</tr>
<tr>
<td>csScatter, CuffData-method (csScatter)</td>
<td>17</td>
</tr>
<tr>
<td>csScatter, CuffFeatureSet-method (csScatter)</td>
<td>17</td>
</tr>
<tr>
<td>csScatterMatrix (csScatter)</td>
<td>17</td>
</tr>
<tr>
<td>csScatterMatrix, CuffData-method (csScatter)</td>
<td>17</td>
</tr>
<tr>
<td>csSpecificity</td>
<td>18</td>
</tr>
<tr>
<td>csSpecificity, CuffData-method (csSpecificity)</td>
<td>18</td>
</tr>
<tr>
<td>csSpecificity, CuffFeatureSet-method (csSpecificity)</td>
<td>18</td>
</tr>
<tr>
<td>csVolcano</td>
<td>19</td>
</tr>
<tr>
<td>csVolcano, CuffData-method (csVolcano)</td>
<td>19</td>
</tr>
<tr>
<td>csVolcano, CuffFeatureSet-method (csVolcano)</td>
<td>19</td>
</tr>
<tr>
<td>csVolcanoMatrix (csVolcano)</td>
<td>19</td>
</tr>
<tr>
<td>csVolcanoMatrix, CuffData-method (csVolcano)</td>
<td>19</td>
</tr>
<tr>
<td>CuffData-class</td>
<td>20</td>
</tr>
<tr>
<td>CuffDist-class</td>
<td>21</td>
</tr>
<tr>
<td>CuffFeature</td>
<td>25, 26</td>
</tr>
</tbody>
</table>
featureNames, 38
featureNames, CuffData-method (featureNames), 38
featureNames, CuffFeatureSet-method (CuffFeatureSet-class), 24
features, 39
features, CuffFeatureSet-method (CuffFeatureSet-class), 24
findGene, 40
findGene, CuffSet-method (findGene), 40
findSimilar, 41
findSimilar, CuffSet-method (findSimilar), 41
fpkm (fpkm-methods), 42
fpkm, CuffData-method (fpkm-methods), 42
fpkm, CuffFeature-method (fpkm-methods), 42
fpkm, CuffFeatureSet-method (fpkm-methods), 42
fpkm-methods, 42
fpkmMatrix, 43
fpkmMatrix, CuffData-method (fpkmMatrix), 43
fpkmMatrix, CuffFeature-method (CuffFeature-class), 22
fpkmMatrix, CuffFeatureSet-method (fpkmMatrix), 43
fpkmSCWPlot (QCplots), 59
fpkmSCWPlot, CuffData-method (QCplots), 59
genePlot (CuffGene-class), 25
genePlot, CuffGene-method (CuffGene-class), 25
genes (CuffSet-class), 28
genes, CuffGene-method (CuffGene-class), 25
genes, CuffGeneSet-method (CuffGeneSet-class), 27
genes, CuffSet-method (CuffSet-class), 28
geneId, 46
geneId, CuffSet-method (geneId), 46
getGeneId, 46
getGeneId, CuffSet-method (getGeneId), 46
getGenes, 47
getGenes, CuffSet-method (getGenes), 47
getLevels, 48
getLevels, CuffData-method (getLevels), 48
getLevels, CuffSet-method (getLevels), 48
getRepLevels, 49
getRepLevels, CuffData-method (getRepLevels), 49
getRepLevels, CuffSet-method (getRepLevels), 49
getSig, 50
getSig, CuffSet-method (getSig), 50
getSigTable, 51
getSigTable, CuffSet-method (getSigTable), 51
isoforms (CuffSet-class), 28
isoforms, CuffGene-method (CuffGene-class), 25
isoforms, CuffGeneSet-method (CuffGeneSet-class), 27
isoforms, CuffSet-method (CuffSet-class), 28
JSdist, 52
JSdistFromP, 53
JSdistVec, 54
length, CuffFeature-method (CuffFeature-class), 22
length, CuffFeatureSet-method (CuffFeatureSet-class), 24
length, CuffGene-method (CuffGene-class), 25
makeGeneRegionTrack (CuffGene-class), 25
makeGeneRegionTrack, CuffGene-method (CuffGene-class), 25
makeprobs, 55
makeprobsvec, 56
makeRnk (CuffData-class), 20
makeRnk, CuffData-method (CuffData-class), 20
MAplot, 57
MAplot, CuffData-method (MAplot), 57
MDSplot (Dimensionality Reduction), 31
MDSplot, CuffData-method (Dimensionality Reduction), 31
INDEX

MDSplot, CuffFeatureSet-method (Dimensionality Reduction), 31
PCAplot (Dimensionality Reduction), 31
PCAplot, CuffData-method (Dimensionality Reduction), 31
PCAplot, CuffFeatureSet-method (Dimensionality Reduction), 31
PINK1, 58
promoters (CuffSet-class), 28
promoters, CuffGene-method (CuffGene-class), 25
promoters, CuffGeneSet-method (CuffGeneSet-class), 27
promoters, CuffSet-method (CuffSet-class), 28
QCplots, 59
readCufflinks, 60
relCDS (CuffSet-class), 28
relCDS, CuffGene-method (CuffGene-class), 25
relCDS, CuffGeneSet-method (CuffGeneSet-class), 27
relCDS, CuffSet-method (CuffSet-class), 28
repCountMatrix (countMatrix), 7
repCountMatrix, CuffData-method (countMatrix), 7
repCountMatrix, CuffFeatureSet-method (countMatrix), 7
repFpkm (repFpkm-methods), 62
repFpkm, CuffData-method (repFpkm-methods), 62
repFpkm, CuffFeature-method (repFpkm-methods), 62
repFpkm, CuffFeatureSet-method (repFpkm-methods), 62
repFpkm-methods, 62
repFpkmMatrix (fpkmMatrix), 43
repFpkmMatrix, CuffData-method (fpkmMatrix), 43
repFpkmMatrix, CuffFeature-method (fpkmMatrix), 43
repFpkmMatrix, CuffFeatureSet-method (fpkmMatrix), 43
replicates, 63
replicates, CuffData-method (replicates), 63
replicates, CuffFeature-method (replicates), 63
replicates, CuffFeatureSet-method (replicates), 63
replicates, CuffSet-method (replicates, replicates), 63
runInfo, 64
runInfo, CuffSet-method (runInfo), 64
sampleGeneSet, 65
sampleIDs, 65
samples, 66
samples, CuffData-method (samples), 66
samples, CuffDist-method (CuffDist-class), 21
samples, CuffFeature-method (CuffFeature-class), 22
samples, CuffFeatureSet-method (CuffFeatureSet-class), 24
samples, CuffSet-method (samples), 28
shannon.entropy, 67
sigMatrix, 68
sigMatrix, CuffSet-method (sigMatrix), 68
splicing (CuffSet-class), 28
splicing, CuffGene-method (CuffGene-class), 25
splicing, CuffGeneSet-method (CuffGeneSet-class), 27
splicing, CuffSet-method (CuffSet-class), 28
TSS (CuffSet-class), 28
TSS, CuffGene-method (CuffGene-class), 25
TSS, CuffGeneSet-method (CuffGeneSet-class), 27
TSS, CuffSet-method (CuffSet-class), 28
varModel (CuffSet-class), 28
varModel, CuffSet-method (CuffSet-class), 28