Package ‘PIMENTo’

August 12, 2016

Title Microarray Analysis Pipeline

Version 1.0

Description PIMENTo is an R package to normalize, analyze, and visualize microarray data

Depends R (>= 3.2.0)

Imports limma,
affy,
samr,
tools,
stringr,
DESeq2,
gplots,
ggplot2,
RColorBrewer

License GPL

LazyData true

RoxygenNote 5.0.1

R topics documented:

`BackgroundCutoff` *Visualize arrays to identify cutoff for background subtraction*

Description

Plot the histogram of max gene illumination across all arrays to identify cutoff for background subtraction. Upper and lower limits can be changed to narrow focus. Plots will be saved in analysis pipeline directory.

Usage

```
BackgroundCutoff(preprocess.data.obj, method=c("mloess","quantile"),
xlim.lo=0, xlim.hi)
```

Arguments

`preprocess.data.obj`	Object returned from call to preprocess.data
`method`	Type of normalization to use: "quantile" for quantile normalization; "mloess" for MLOESS normalization
`xlim.lo`	Lower bound on X for histogram plot (binary logarithm)
`xlim.hi`	Upper bound on X for histogram plot (binary logarithm)

BackgroundSubtraction *Perform background subtraction on normalized data*

Description

Remove all genes that have an maximum intensity level below that of the provided cutoff.

Usage

```
BackgroundSubtraction(preprocess.data.obj,
method=c("mloess", "quantile"), cutoff)
```

Arguments

`preprocess.data.obj`	Object returned from call to preprocess.data
`method`	Type of normalization to use: "quantile" for quantile normalization; "mloess" for MLOESS normalization
`cutoff`	The cutoff value (binary logarithm) identified through background.cutoff

Value

A list with components

`ntext`	Number of leading text columns
`data.col`	Vector of column indices containing array data
`id`	Vector containing gene ID's
`id.ind`	Column index containing gene ID information
`symbol`	Vector containing gene symbols
`symbol.index`	Column index containing gene symbol information
`desc.stats`	Vector of column indices containing descriptive statistics
`pipeline.name`	Name of pipeline generated from input file name sans extension
`normalized`	Data frame of descriptive stats and normalized data for the chosen method

CreateHeatmap *Create heatmaps on desired subsets of genes*

Description

Normalize input raw data using quantile and mloess methods. Plots of the normalized data along with a dendrogram clustering all samples will be stored in newly created pipeline directory.

Usage

```
CreateHeatmap(sig.genes.sam.obj, subsets.dir, file.format=c("geneid",
"symbol"))
```

Arguments

`sig.genes.sam.obj`	Object returned from call to SigGenesSAM
`subsets.dir`	Directory containing files of genes output from pathway analysis or simply genes of interest
`file.format`	Indicator of how genes are identified in each file, be it "geneid" or "symbol"

PreprocessData *Generate MA plots of raw and normalized data*

Description

Normalize input raw data using quantile and mloess methods. Plots of the normalized data along with a dendrogram clustering all samples will be stored in newly created pipeline directory.

Usage

```
preprocessPlots(input.file, file.sheet=1, ntext=2, data.col,
symbol.index=1, id.index=2)
```

Arguments

`input.file`	Path to the microarray expression file, be it .xlsx or .csv
`file.sheet`	Sheet number in the spreadsheet with data
`ntext`	Number of leading text columns
`data.col`	Range of columns which contain data (indexing begins with first column of file)
`symbol.index`	Column index which contains gene symbols
`id.index`	Column index which contains gene ID’s
`batch.vector`	Character vector indicating to which batch each sample belongs

Value

A list with components

`ntext` Number of leading text columns

`data.col` Vector of column indices containing array data

`id` Vector containing gene ID's

`id.index` Column index containing gene ID information

`symbol` Vector containing gene symbols

`symbol.index` Column index containing gene symbol information

`desc.stats` Vector of column indices containing descriptive statistics

`pipeline.name` Name of pipeline generated from input file name sans extension

`mloess` Data rame of quantile normalized data

`quantile` Data rame of quantile normalized data

`SigGenesSAM` *Identify significant genes through SAM*

Description

Implement SAM and compute significant genes given delta. Output will consist of all significant genes ordered by increasing q-value and decreasing d-score.

Usage

```
SigGenesSAM(background.subtraction.obj, class.compare.cols,
class.compare.name, fdr.cutoff=0.1, response)
```

Arguments

`background.subtraction.obj`
Object returned from call to BackgroundSubtraction

`class.compare.cols`
Vector of column indices indicating which subset of arrays are to be compared for this comparison

`class.compare.name`
String title given to the name of the comparison

`fdr.cutoff` Max FDR for SAM, will use delta value which results in max FDR below this cutoff

`response` For two class unpaired: vector of 1, 2 values that indicate group membership. For two class paired: vector of -1, 1, -2, 2, etc. values that indicate pairings.

Value

A list with components

`siggenes.table` Combined data frame of genes having significant positive and negative correlation

`data.col` Vector of column indices containing array data

`ntext` Number of leading text columns

`response` Vector of array group membership, 1=control, 2=experimental

`pipeline.name` Name of pipeline generated from input file name sans extension

`data` Data frame of chosen normalization method data

`class.compare.cols` Value entered through class.compare.cols parameter

`class.compare.name` Value entered through class.compare.name parameter

`symbol.index` Column index that contains gene symbol

Index