

Package: BiSEp (via r-universe)

September 2, 2024

Type Package

Title Toolkit to Identify Candidate Synthetic Lethality

Version 2.3

Date 2024-05-04

Author Mark Wappett

Maintainer Mark Wappett <m.wappett@qub.ac.uk>

Depends R (>= 3.3.1)

Imports AnnotationDbi, mclust, GO.db, org.Hs.eg.db, GOSemSim

Description Enables the user to infer potential synthetic lethal relationships by analysing relationships between bimodally distributed gene pairs in big gene expression datasets.
Enables the user to visualise these candidate synthetic lethal relationships.

License Artistic-2.0

NeedsCompilation no

Repository CRAN

Date/Publication 2024-05-04 23:50:02 UTC

Contents

BiSEp-package	2
BEEM	2
BIGEE	3
BISEP	4
BISEP_dat	5
BISEP_data	5
expressionPlot	6
FURE	7
FURE_data	7
INPUT_data	8
MUT_data	8
waterfallPlot	8

Index**10**

BiSEp-package	<i>BiSEp: Bimodality in gene expression to dissect tumours and reveal synthetic lethal drug targets and biomarkers</i>
---------------	--

Description

A set of tools that enable the user to accurately identify bimodality and non-normality in gene expression data and stratify samples as high or low expression for bimodal genes. Enables identification of candidate synthetic lethal gene pairs. Enables the user to assess and visualise functional redundancy between candidate synthetic lethal gene pairs.

Details

Package: BiSEp
 Type: Package
 Version: 2.0
 Date: 2014-10-21
 License: GPL-2

This package has a mixture of CRAN and bioconductor packages listed as dependencies. Please ensure that you have Bioconductor installed.

Author(s)

Author: Mark Wappett

Maintainer: Mark Wappett <mark.wappett@astrazeneca.com>

 BEEM

BEEM: Bimodal Expression Exclusive with Mutation

Description

Takes the output from the function BISEP and a discrete mutation matrix as input. The mutation matrix samples (columns) must mirror or overlap with the gene expression matrix. The data in the mutation matrix must be a discrete 'WT' or 'MUT' call based on the status of each gene with each sample. Detects mutations of genes enriched in either the high or low gene expression modes.

Usage

```
BEEM(
  bisepData=data,
  mutData=mutData,
  sampleType=c("cell_line", "cell_line_low", "patient", "patient_low"),
  minMut=10
)
```

Arguments

bisepData	This should be the output from the BISEP function.
mutData	This should be a matrix with genes rownames and samples as column names. All cells should be made up of a discreet 'WT' or 'MUT' call. There should be overlap (by sample) with the gene expression matrix.
sampleType	The type of sample being analysed. Select 'cell_line' or 'patient' for datasets with greater than ~200 samples. For datasets with less than ~200 samples, use 'cell_line_low' or 'patient_low'.
minMut	The minimum number of mutations you for a gene would consider for analysis.

Details

Lower sample numbers have more stringent bimodality hurdles to clear in order to keep the false positive rate lower. The tool returns a percentage complete text window so the user can observe the status of the job.

Value

A matrix containing 10 columns. Column 1 contains the bimodal genes from the expression data (gene 1) and column 2 contains the mutated candidate synthetic lethal gene pair (gene 2). Columns 3 and 4 contain the number of mutations of gene 2 in the low and high expression modes of gene 1. Column 5 contains the fishers p value that evaluates enrichment of mutation in either the high or low mode (indicated by column 10). Columns 6 and 7 contain the percentage of samples in the low and high expression modes of gene 1 that are mutated for gene 2. Columns 8 and 9 contain information on the overall size (in terms of sample) of the low and high expression modes of gene 1.

Author(s)

Mark Wappett

BIGEE

BIGEE: Bimodal Gene Expression Exclusivity.

Description

Part of the Synthetic Lethality detection in Genomics toolkit. Detects bimodality and non-normality in all genes across the dataset. Compares all pairwise combinations of bimodal genes and searches for mutually exclusive low expression as evidence of potential synthetic lethality. Scores gene-pairs based on the presence of mutual exclusive bimodality and the distribution of signal intensity across the rest of the dataset.

Usage

```
BIGEE(
  bisepData=data,
  sampleType=c("cell_line", "cell_line_low", "patient", "patient_low")
)
```

Arguments

bisepData	This should be the output from the BISEP function.
sampleType	The type of sample being analysed. Select 'cell_line' or 'patient' for datasets with greater than ~200 samples. For datasets with less than ~200 samples, use 'cell_line_low' or 'patient_low'.

Details

Lower sample numbers have more stringent bimodality hurdles to clear in order to keep the false positive rate lower. The tool returns a percentage complete text window so the user can observe the status of the job.

Value

A matrix containing three columns. Columns 1 and 2 are the gene symbols that make up the candidate synthetic lethal gene pairs. Column 3 is the score calculated the tool to rank the statistical significance of the gene pairs.

Author(s)

Mark Wappett

BISEP

BISEP: Bimodality in Gene Expression data.

Description

Detects bimodality and non-normality in all genes across the dataset.

Usage

```
BISEP(
  data = data
)
```

Arguments

data	This should be a log2 gene expression matrix with genes as rownames and samples as column names. Suitable for gene expression data from any platform - NGS datasets should be RPKM or RSEM values.
------	--

Details

The lower confidence calls will dramatically affect the number of gene pairs that the tool produces and increase the false positive rate. The tool will take approximately 10 minutes to run a 5,000 row and 200 column input matrix using a 'medium,' confidence interval.

Value

A list containing three matrices. Matrix 1 contains the output of the BISEP algorithm - including the midpoint of the bimodal distribution and the associated p value. Matrix 2 contains the output from the BI algorithm - including the delta, pi and BI values. Matrix 3 contains the input matrix.

Author(s)

Mark Wappett

Examples

```
data(INPUT_data)
outputBISEP <- BISEP(data=INPUT_data)
```

BISEP_dat	<i>A list object containing 3 data frames.</i>
-----------	--

Description

Matrix 1 contains the output of the BISEP algorithm - including the midpoint of the bimodal distribution and the associated p value. Matrix 2 contains the output from the BI algorithm - including the delta, pi and BI values. Matrix 3 contains the input matrix.

Usage

```
data(BISEP_dat)
```

Format

13 observations across 100 variables.

BISEP_data	<i>A list object containing 3 data frames.</i>
------------	--

Description

Matrix 1 contains the output of the BISEP algorithm - including the midpoint of the bimodal distribution and the associated p value. Matrix 2 contains the output from the BI algorithm - including the delta, pi and BI values. Matrix 3 contains the input matrix.

Usage

```
data(BISEP_data)
```

Format

13 observations across 442 variables.

`expressionPlot`*expressionPlot: Create visualisations from BIGEE output*

Description

Takes the output from the function BISEP and two gene names that correspond to a relevant gene pair. Gene names must be available in the input BISEP object.

Usage

```
expressionPlot(  
  bisepData=data,  
  gene1,  
  gene2  
)
```

Arguments

<code>bisepData</code>	This should be the output from the BISEP function.
<code>gene1</code>	The first gene whose expression you would like to plot.
<code>gene2</code>	The second gene whose expression you would like to plot.

Details

The function will return an error if any of the input information is incorrect or missing. The resulting plot will be returned in real time.

Value

A scatter plot of the two genes you have identified as bimodal. The red lines correspond to the mid-points of the bimodal distribution for these two genes. Ideally the lower left quadrant would be empty when observing a candidate SL interaction.

Author(s)

Mark Wappett

Examples

```
data(BISEP_data)  
data(MUT_data)  
expressionOut <- expressionPlot(BISEP_data, gene1="SMARCA1", gene2="SMARCA4")
```

FURE	<i>FURE: Functional redundancy between synthetic lethal gene pairs</i>
------	--

Description

Utilises gene ontology information from the GO database bioconductor package. Assesses gene pairs output from the SLinG and BEEM tools for gene ontology functional redundancy. Performs semantic similarity scoring utilising the GOSemSim bioconductor package

Usage

```
FURE(
  data=data,
  inputType=inputType)
```

Arguments

data	This should be the output matrix (or similar) from the SLinG and BEEM tools. Columns 1 and 2 should be gene symbols.
inputType	Either 'BIGEE' or 'BEEM' based on origin of the input matrix.

Value

A list of matrices containing gene pairs with associated synthetic lethal statistical significance values + gene ontology annotation/ scores.

Author(s)

Mark Wappett

FURE_data	<i>Output matrix from the BIGEE tool</i>
-----------	--

Description

Output matrix from the BIGEE tool.

Usage

```
data(FURE_data)
```

Format

A data frame with 1 observation across 3 variables.

INPUT_data	<i>A Log2 Gene Expression matrix</i>
------------	--------------------------------------

Description

A Log2 Gene Expression matrix where rownames are genes and colnames are samples

Usage

```
data(INPUT_data)
```

Format

13 observations across 442 variables.

MUT_data	<i>A matrix containing discreet mutation calls</i>
----------	--

Description

A matrix containing discreet mutation calls of either 'WT' or 'MUT' where rownames are genes and column names are samples

Usage

```
data(MUT_data)
```

Format

4 observations across 442 variables.

waterfallPlot	<i>waterfallPlot: Create visualisations from BEEM output</i>
---------------	--

Description

Takes the output from the function BISEP and a discreet mutation matrix as input. The mutation matrix samples (columns) must mirror or overlap with the gene expression matrix. The data in the mutation matrix must be a discreet 'WT' or 'MUT' call based on the status of each gene with each sample. Gene names must be available in the input matrices.

Usage

```
waterfallPlot(  
  bisepData=data,  
  mutData=mutData,  
  expressionGene,  
  mutationGene  
)
```

Arguments

bisepData	This should be the output from the BISEP function.
mutData	This should be a matrix with genes rownames and samples as column names. All cells should be made up of a discreet 'WT' or 'MUT' call. There should be overlap (by sample) with the gene expression matrix.
expressionGene	The gene whose expression you would like to plot.
mutationGene	The gene whose mutation status you would like to overlap with the expression gene.

Details

The function will return an error if any of the input information is incorrect or missing. The resulting plot will be returned in real time.

Value

A waterfall plot. The plot is made up of two panels: the left panel is a density distribution of the expression gene provided and the right panel is a bar-chart of the gene expression level coloured by mutation status.

Author(s)

Mark Wappett

Examples

```
data(BISEP_data)  
data(MUT_data)  
waterfallOut <- waterfallPlot(BISEP_data, MUT_data, expressionGene="micb", mutationGene="PBRM1")
```

Index

- * **Bimodality**
 - BIGEE, 3
 - BISEP, 4
 - * **Expression**
 - expressionPlot, 6
 - * **GeneOntology**
 - FURE, 7
 - * **Mutation**
 - BEEM, 2
 - waterfallPlot, 8
 - * **SyntheticLethal**
 - BEEM, 2
 - BIGEE, 3
 - BISEP, 4
 - expressionPlot, 6
 - FURE, 7
 - waterfallPlot, 8
 - * **datasets**
 - BISEP_dat, 5
 - BISEP_data, 5
 - FURE_data, 7
 - INPUT_data, 8
 - MUT_data, 8
 - * **package**
 - BiSEp-package, 2
- BEEM, 2
- BIGEE, 3
- BISEP, 4
- BiSEp (BiSEp-package), 2
- BiSEp-package, 2
- BISEP_dat, 5
- BISEP_data, 5
- expressionPlot, 6
- FURE, 7
- FURE_data, 7
- INPUT_data, 8
- MUT_data, 8
- waterfallPlot, 8