

# Package ‘DriverNet’

October 7, 2014

**Type** Package

**Title** Drivernet: uncovering somatic driver mutations modulating transcriptional networks in cancer

**Version** 1.4.0

**Date** 2012-03-21

**Author** Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah

**Maintainer** Jiarui Ding <jiaruid@cs.ubc.ca>

**Description** DriverNet is a package to predict functional important driver genes in cancer by integrating genome data (mutation and copy number variation data) and transcriptome data (gene expression data). The different kinds of data are combined by an influence graph, which is a gene-gene interaction network deduced from pathway data. A greedy algorithm is used to find the possible driver genes, which may mutated in a larger number of patients and these mutations will push the gene expression values of the connected genes to some extreme values.

**License** GPL-3

**LazyLoad** yes

**Depends** R (>= 2.10), methods

**biocViews** Network

## R topics documented:

DriverNet-package . . . . .	2
actualEvents . . . . .	3
computeDrivers . . . . .	4
computeRandomizedResult . . . . .	5
DriverNetResult-class . . . . .	6
drivers . . . . .	7

getPatientOutlierMatrix . . . . .	7
preprocessMatrices . . . . .	8
resultSummary . . . . .	9
sampleDriversList . . . . .	10
sampleGeneNames . . . . .	11
sampleInfluenceGraph . . . . .	11
samplePatientExpressionMatrix . . . . .	12
samplePatientMutationMatrix . . . . .	12
samplePatientOutlierMatrix . . . . .	13
sampleRandomDriversResult . . . . .	13
totalEvents . . . . .	14

<b>Index</b>	<b>15</b>
--------------	-----------

---

DriverNet-package	<i>Drivernet: uncovering somatic driver mutations modulating transcriptional networks in cancer</i>
-------------------	---

---

## Description

Description: DriverNet is a package to predict functional important driver genes in cancer by integrating genome data (mutation and copy number variation data) and transcriptome data (gene expression data). The different kinds of data are combined by an influence graph, which is a gene-gene interaction network deduced from pathway data. A greedy algorithm is used to find the possible driver genes, which may mutated in a larger number of patients and these mutations will push the gene expression values of the connected genes to some extreme values.

## Details

Package:	DriverNet
Type:	Package
Version:	1.0.0
Date:	2012-03-21
License:	GPL-3
LazyLoad:	yes
Depends:	methods

Users would call `preprocess_matrices` on the three matrices to remove unnecessary data. The matrices returned are then be passed to other functions.

## Author(s)

Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah

Maintainer: Jiarui Ding <jiaruid@cs.ubc.ca>

## References

DriverNet: uncovering somatic driver mutations modulating transcriptional networks in cancer, submitted.

## Examples

```
data(samplePatientMutationMatrix)
data(samplePatientOutlierMatrix)
data(sampleInfluenceGraph)
data(sampleGeneNames)

driversList = computeDrivers(samplePatientMutationMatrix, samplePatientOutlierMatrix,
sampleInfluenceGraph, outputFolder=NULL, printToConsole=FALSE)

drivers(driversList)[1:10]

randomDriversResult = computeRandomizedResult(patMutMatrix=samplePatientMutationMatrix,
patOutMatrix=samplePatientOutlierMatrix, influenceGraph=sampleInfluenceGraph,
geneNameList= sampleGeneNames, outputFolder=NULL, printToConsole=FALSE,
numberOfRandomTests=20, weight=FALSE, purturbGraph=FALSE, purturbData=TRUE)

res = resultSummary(driversList, randomDriversResult, samplePatientMutationMatrix,
sampleInfluenceGraph, outputFolder=NULL, printToConsole=FALSE)
```

---

actualEvents	<i>Actual events covered by driver mutations</i>
--------------	--

---

## Description

It gives a detailed list of events covered by the driver mutations selected by DriverNet.

## Usage

```
actualEvents(x)
```

## Arguments

x

## See Also

computeDrivers DriverNetResult-class

## Examples

```
data(sampleDriversList)
actualEvents(sampleDriversList)
```

---

computeDrivers	<i>Compute a list of driver mutations</i>
----------------	---

---

**Description**

Use a greedy algorithm to rank a list of driver mutations.

**Usage**

```
computeDrivers(patMutMatrix, patOutMatrix, influenceGraph, outputFolder = NULL,  
printToConsole = FALSE, weighted = FALSE)
```

**Arguments**

patMutMatrix	Patient Mutation Matrix
patOutMatrix	Patient Outlier Matrix
influenceGraph	Influence Graph Matrix
outputFolder	The folder to store the log. If set to NULL, no log files will be written. If set to "", the log will be written to the current folder.
printToConsole	If set to TRUE, progress and result of the function will be printed to the console.
weighted	Must be set to FALSE in this version.

**Value**

An object of DriverNetResult class that can be passed to the resultSummary method.

**Author(s)**

Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah  
Maintainer: Jiarui Ding <jjaruid@cs.ubc.ca>

**Examples**

```
data(samplePatientMutationMatrix)  
data(samplePatientOutlierMatrix)  
data(sampleInfluenceGraph)  
driversList = computeDrivers(samplePatientMutationMatrix, samplePatientOutlierMatrix,  
sampleInfluenceGraph, outputFolder=NULL, printToConsole=FALSE)  
  
drivers(driversList)[1:10]
```

---

`computeRandomizedResult`*Randomly compute lists of driver mutations*

---

### Description

It renames the mutations in `patMutMatrix` with a randomized list of gene names and computes a list of driver mutations. It will repeat this process by `numberOfRandomTests` times. The result could be passed to the `resultSummary` method to calculate p-value for the driver mutations.

### Usage

```
computeRandomizedResult(patMutMatrix, patOutMatrix, influenceGraph,  
geneNameList, outputFolder = NULL, printToConsole = FALSE,  
numberOfRandomTests = 500, weighted = FALSE, purturbGraph = FALSE,  
purturbData = TRUE)
```

### Arguments

<code>patMutMatrix</code>	Patient Mutation Matrix
<code>patOutMatrix</code>	Patient Expression(Outlier) Matrix
<code>influenceGraph</code>	Influence Graph Matrix
<code>geneNameList</code>	A list of gene names that the new mutation names will be randomly chosen from.
<code>outputFolder</code>	The folder to store the log. If set to <code>NULL</code> , no log files will be written. If set to <code>""</code> , the log will be written to the current folder.
<code>printToConsole</code>	If set to <code>TRUE</code> , progress and result of the function will be printed to the console.
<code>numberOfRandomTests</code>	Number of lists of randomized driver mutations to be generated.
<code>weighted</code>	Must be set to <code>FALSE</code> in this version.
<code>purturbGraph</code>	Must be set to <code>FALSE</code> in this version.
<code>purturbData</code>	Must be set to <code>TRUE</code> in this version.

### Value

A list of numeric vectors where each vector stores the number of events covered by the randomly identified drivers. The list can be passed to the `resultSummary` method to compute p-values.

### Author(s)

Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah  
Maintainer: Jiarui Ding <jiaruid@cs.ubc.ca>

**Examples**

```

data(samplePatientMutationMatrix)
data(samplePatientOutlierMatrix)
data(sampleInfluenceGraph)
data(sampleGeneNames)

randomDriversResult = computeRandomizedResult(patMutMatrix=samplePatientMutationMatrix,
patOutMatrix=samplePatientOutlierMatrix, influenceGraph=sampleInfluenceGraph,
geneNameList=sampleGeneNames, outputFolder=NULL, printToConsole=FALSE,
numberOfRandomTests=20, weight=FALSE, purturbGraph=FALSE, purturbData=TRUE)

```

---

DriverNetResult-class *Class "DriverNetResult"*

---

**Description**

A class storing the result from running the `computeDrivers` function. It contains the list of driver mutations found, the detailed events covered by these drivers and the total number of events in the test data.

**Objects from the Class**

Objects can be created by calls of the form `new("DriverNetResult", drivers=..., actualEvents=..., totalEvents=...`

**Slots**

**drivers:** Object of class "character" ~~  
**actualEvents:** Object of class "list" ~~  
**totalEvents:** Object of class "numeric" ~~

**Methods**

**actualEvents** signature(x = "DriverNetResult"): ...  
**drivers** signature(x = "DriverNetResult"): ...  
**totalEvents** signature(x = "DriverNetResult"): ...

**Author(s)**

Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah  
Maintainer: Jiarui Ding <jjaruid@cs.ubc.ca>

**See Also**

[computeDrivers](#)

**Examples**

```
data(sampleDriversList)
drivers(sampleDriversList)[1:10]
```

---

drivers	<i>List of driver mutations identified by DriverNet</i>
---------	---

---

**Description**

It gives the list of driver mutations identified by DriverNet ranked by the number of events that they cover.

**Usage**

```
drivers(x)
```

**Arguments**

x

**See Also**

computeDrivers DriverNetResult-class

**Examples**

```
data(sampleDriversList)
actualEvents(sampleDriversList)
```

---

getPatientOutlierMatrix	<i>Compute the patient outlier matrix</i>
-------------------------	---

---

**Description**

Given a real-value patient expression matrix, compute the patient outlier matrix and remove the genes which are not expressed.

**Usage**

```
getPatientOutlierMatrix(patExpMatrix, th=2)
```

**Arguments**

patExpMatrix	Patient Expression Matrix
th	The threshold, the default value is 2 ( the gene expression values outside the 2 standard deviation range are considered as outliers).

**Value**

The output patOutMatrix is an input to the computeDrivers method

**Author(s)**

Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah  
 Maintainer: Jiarui Ding <jjaruid@cs.ubc.ca>

**See Also**

computeDrivers

**Examples**

```
data(samplePatientExpressionMatrix)
samplePatientOutlierMatrix = getPatientOutlierMatrix(samplePatientExpressionMatrix)
```

---

preprocessMatrices      *Remove unnecessary entries from matrices*

---

**Description**

Remove patients (rows) which are not in both matrices from patMutMatrix and patOutMatrix. Remove mutations (columns) from patMutMatrix if they are not a row of influenceGraph. Remove expressions (columns) from patOutMatrix if they are not a column of influenceGraph.

**Usage**

```
preprocessMatrices(patMutMatrix, patOutMatrix, influenceGraph)
```

**Arguments**

patMutMatrix	Patient Mutation Matrix
patOutMatrix	Patient Outlier Matrix
influenceGraph	Influence Graph Matrix

**Value**

1	Reduced version of Patient Mutation Matrix
2	Reduced version of Patient Expression(Outlier) Matrix
3	Influence Graph Matrix

**Note**

The dimension of influenceGraph is not changed.



**Author(s)**

Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah  
Maintainer: Jiarui Ding <jiaruid@cs.ubc.ca>

**Examples**

```
data(samplePatientMutationMatrix)
data(samplePatientOutlierMatrix)
data(sampleInfluenceGraph)
tmp = preprocessMatrices(samplePatientMutationMatrix, samplePatientOutlierMatrix,
sampleInfluenceGraph)

smallerPatientMutationMatrix = tmp[[1]]
smallerPatientOutlierMatrix = tmp[[2]]
influenceGraph = tmp[[3]]
```

---

resultSummary

*Summarize result for drivers ranking.*

---

**Description**

Ranks the drivers.

**Usage**

```
resultSummary(mainResult, randResult, patMutMatrix, influenceGraph,
outputFolder = NULL, printToConsole = FALSE)
```

**Arguments**

**mainResult** An object of DriverNetResult class created by the computeDrivers method.  
**randResult** A list of vectors representing driver coverage generated by computeRandomizedResult.  
**patMutMatrix** Patient Mutation Matrix  
**influenceGraph** Influence Graph Matrix  
**outputFolder** The folder to store the summary file. If set to NULL, no files will be written. If set to "", the files will be written to the current folder.  
**printToConsole** If set to TRUE, progress and result of the function will be printed to the console.

**Value**

A matrix storing the summary result.

**Author(s)**

Ali Bashashati, Reza Haffari, Jiarui Ding, Gavin Ha, Kenneth Liu, Jamie Rosner and Sohrab Shah  
Maintainer: Jiarui Ding <jiaruid@cs.ubc.ca>

**Examples**

```
data(sampleDriversList)
data(sampleRandomDriversResult)
data(samplePatientMutationMatrix)
data(sampleInfluenceGraph)

res = resultSummary(sampleDriversList, sampleRandomDriversResult,
samplePatientMutationMatrix, sampleInfluenceGraph, outputFolder=NULL,
printToConsole=FALSE)
```

---

sampleDriversList	<i>Sample DriverNet result</i>
-------------------	--------------------------------

---

**Description**

An object of DriverNetResult class that obtained from running the codecomputeDrivers function.

**Usage**

```
data(sampleDriversList)
```

**Format**

Formal class 'DriverNetResult' with 3 slots (1) @ drivers: a character vector of the driver gene name; (2) @ actualEvents: the covered events in each mutated patient; (3) @totalEvents: the total number of outliers

**See Also**

computeDrivers

**Examples**

```
data(sampleDriversList)

drivers(sampleDriversList)[1:10]
```

---

sampleGeneNames	<i>Sample gene names</i>
-----------------	--------------------------

---

**Description**

A list of gene names used to generate permutation of names in computeRandomizedResult method.

**Usage**

```
data(sampleGeneNames)
```

**Format**

A factor of gene names

**Examples**

```
data(sampleGeneNames)
```

---

sampleInfluenceGraph	<i>Sample influence graph</i>
----------------------	-------------------------------

---

**Description**

A binary matrix representing influence graph between genes.

**Usage**

```
data(sampleInfluenceGraph)
```

**Format**

A two-dimensional binary matrix. The row and column names are gene names. If two genes  $i$  and  $j$  are in the same pathway,  $\text{sampleInfluenceGraph}[i, j] = 1$ .

**Examples**

```
data(sampleInfluenceGraph)
```

---

samplePatientExpressionMatrix  
*Sample patient expression matrix*

---

**Description**

A real-value matrix representing gene expressions in patients.

**Usage**

```
data(samplePatientExpressionMatrix)
```

**Format**

A two-dimensional real value matrix. The row names are patients, and column names are genes. The entry `samplePatientExpressionMatrix[i,j]` is the gene expression values of gene `j` in patient `i`.

**Examples**

```
data(samplePatientExpressionMatrix)
```

---

samplePatientMutationMatrix  
*Sample patient mutation matrix*

---

**Description**

A binary matrix representing gene mutations in patients.

**Usage**

```
data(samplePatientMutationMatrix)
```

**Format**

A two-dimensional binary matrix. The row names are patients, and column names are genes. If gene `j` is mutated in patient `i`, `samplePatientMutationMatrix[i, j]=1`.

**Examples**

```
data(samplePatientMutationMatrix)
```

---

samplePatientOutlierMatrix  
*Sample patient outlier matrix*

---

**Description**

A binary matrix representing gene expressions in patients.

**Usage**

```
data(samplePatientOutlierMatrix)
```

**Format**

A two-dimensional binary matrix. The row names are patients, and column names are genes. If genes  $j$  is an outlier in patient  $i$ , `samplePatientOutlierMatrix[i, j] = 1`.

**Examples**

```
data(samplePatientOutlierMatrix)
```

---

sampleRandomDriversResult  
*Sample Result from computeRandomizedResult*

---

**Description**

A sample result from running `computeRandomizedResult` with `numberOfRandomTests = 20`.

**Usage**

```
data(sampleRandomDriversResult)
```

**Format**

A list of predicted drivers from different runs.

**See Also**

```
computeRandomizedResult
```

**Examples**

```
data(sampleRandomDriversResult)
```

---

totalEvents	<i>Total number of events in the data</i>
-------------	---

---

**Description**

Total number of events in the data

**Usage**

```
totalEvents(x)
```

**Arguments**

x

**See Also**

computeDrivers DriverNetResult-class

**Examples**

```
data(sampleDriversList)
totalEvents(sampleDriversList)
```

# Index

## \*Topic **classes**

DriverNetResult-class, 6

## \*Topic **datasets**

sampleDriversList, 10  
sampleGeneNames, 11  
sampleInfluenceGraph, 11  
samplePatientExpressionMatrix, 12  
samplePatientMutationMatrix, 12  
samplePatientOutlierMatrix, 13  
sampleRandomDriversResult, 13

## \*Topic **methods**

actualEvents, 3  
computeDrivers, 4  
computeRandomizedResult, 5  
drivers, 7  
getPatientOutlierMatrix, 7  
preprocessMatrices, 8  
resultSummary, 9  
totalEvents, 14

## \*Topic **package**

DriverNet-package, 2

actualEvents, 3

actualEvents, DriverNetResult-method  
(DriverNetResult-class), 6

computeDrivers, 4, 6

computeRandomizedResult, 5

DriverNet (DriverNet-package), 2

DriverNet-package, 2

DriverNetResult-class, 6

drivers, 7

drivers, DriverNetResult-method  
(DriverNetResult-class), 6

getPatientOutlierMatrix, 7

preprocessMatrices, 8

resultSummary, 9

sampleDriversList, 10

sampleGeneNames, 11

sampleInfluenceGraph, 11

samplePatientExpressionMatrix, 12

samplePatientMutationMatrix, 12

samplePatientOutlierMatrix, 13

sampleRandomDriversResult, 13

totalEvents, 14

totalEvents, DriverNetResult-method  
(DriverNetResult-class), 6