Package ‘MoonlightR’

March 28, 2017

Type Package

Title Identify oncogenes and tumor suppressor genes from omics data

Version 1.0.0

Date 09-06-2016

Author Antonio Colaprico*, Catharina Olsen*, Claudia Cava, Thilde Terkelsen, Laura Cantini, Andre Olsen, Gloria Bertoli, Andrei Zinovyev, Emmanuel Barillot, Isabella Castiglioni, Elena Papaleo, Gianluca Bontempi

Maintainer Antonio Colaprico <antonio.colaprico@ulb.ac.be>, Catharina Olsen <colsen@ulb.ac.be>

Depends R (>= 3.3), doParallel, foreach

Imports parmigene, randomForest, SummarizedExperiment, gplots, circlize, RColorBrewer, HiveR, clusterProfiler, DOSE, Biobase, limma, grDevices, graphics, TCGAbiolinks, GEOquery, stats, RISmed, grid, utils

Description Motivation: The understanding of cancer mechanism requires the identification of genes playing a role in the development of the pathology and the characterization of their role (notably oncogenes and tumor suppressors). Results: We present an R/bioconductor package called MoonlightR which returns a list of candidate driver genes for specific cancer types on the basis of TCGA expression data. The method first infers gene regulatory networks and then carries out a functional enrichment analysis (FEA) (implementing an upstream regulator analysis, URA) to score the importance of well-known biological processes with respect to the studied cancer type. Eventually, by means of random forests, MoonlightR predicts two specific roles for the candidate driver genes: i) tumor suppressor genes (TSGs) and ii) oncogenes (OCGs). As a consequence, this methodology does not only identify genes playing a dual role (e.g. TSG in one cancer type and OCG in another) but also helps in elucidating the biological processes underlying their specific roles. In particular, MoonlightR can be used to discover OCGs and TSGs in the same cancer type. This may help in answering the question whether some genes change role between early stages (I, II) and late stages (III, IV) in breast cancer. In the future, this analysis could be useful to determine the causes of different resistances to
chemotherapeutic treatments.

License  GPL (>= 3)

biocViews  DNAmethylation, DifferentialMethylation, GeneRegulation,
            GeneExpression, MethylationArray, DifferentialExpression,
            Pathways, Network, Survival, GeneSetEnrichment,
            NetworkEnrichment

Suggests  BiocStyle, knitr, rmarkdown, testthat, devtools, roxygen2,
          png

VignetteBuilder  knitr

LazyData  true

URL  https://github.com/torongs82/Moonlight

BugReports  https://github.com/torongs82/Moonlight/issues

RoxygenNote  5.0.1

NeedsCompilation  no

\textbf{R topics documented: }

\begin{verbatim}
dataFilt .......................... 3
dataGRN ................................ 3
dataURA ......................... 4
DEGsMatrix ..................... 4
DiseaseList ....................... 5
DPA .................................. 5
EAGenes .......................... 6
FEA .................................. 6
GDCprojects ...................... 7
geneInfo ......................... 7
GEO_TCGAtab ..................... 8
dataGEO ................................ 8
dataTCGA .......................... 9
GRN ................................ 10
GSEA ................................ 10
knownDriverGenes .............. 11
listMoonlight ...................... 12
LPA .................................. 12
moonlight ......................... 13
MoonlightR ........................ 14
plotCircos ....................... 14
plotFEA .......................... 15
plotNetworkHive .................. 16
plotURA .......................... 16
PRA ................................ 17
tabGrowBlock ...................... 17
URA ................................ 18
\end{verbatim}
**dataFilt**

Gene Expression (Rnaseqv2) data from TCGA LUAD

**Description**
A data set containing the following data:

**Usage**
data(dataFilt)

**Format**
A 13742x20 matrix

**Details**
- dataFilt matrix with 13742 rows (genes) and 20 columns samples with TCGA's barcodes (10TP, 10NT)

**Value**
a 13742x20 matrix

---

**dataGRN**

GRN gene regulatory network output

**Description**
output from GRN function

**Usage**
data(dataGRN)

**Format**
A large list of 2 elements

**Details**
- dataGRN list of 2 elements miTFGenes, maxmi from GRN function

**Value**
a large list of 2 elements
## dataURA

**Description**

A data set containing the following data:

**Usage**

```r
data(dataURA)
```

**Format**

A data frame with 100 rows and 2 variables

**Details**

- dataURA matrix with 100 rows (genes) and 2 columns "apoptosis" "proliferation of cells"

**Value**

a 100x2 matrix

---

## DEGsmatrix

**Description**

A data set containing the following data:

**Usage**

```r
data(DEGsmatrix)
```

**Format**

A 3502x5 matrix

**Details**

- DEGsmatrix matrix with 3502 rows (genes) and five columns "logFC" "logCPM" "LR" "PValue" "FDR"

**Value**

the 3502x5 matrix
**DiseaseList**  
*Information on 101 biological processes*

**Description**

A data set containing the following data:

**Usage**

```r
data(DiseaseList)
```

**Format**

A list of 101 matrices

**Details**

- DiseaseList list for 101 biological processes, each containing a matrix with five columns: ID, Genes.in.dataset, Prediction based on expression direction, Log ratio, Findings

**Value**

list of 101 matrices

---

**DPA**

*DPA*

**Description**

This function carries out the differential phenotypes analysis

**Usage**

```r
DPA(dataType, dataFilt, dataConsortium = "TCGA", fdr.cut = 0.01, logFC.cut = 1, diffmean.cut = 0.25, samplesType, colDescription, gset, gsetFile = "gsetFile.RData")
```

**Arguments**

- **dataType** selected
- **dataFilt** obtained from `getDataTCGA`
- **dataConsortium** is TCGA or GEO, default TCGA
- **fdr.cut** is a threshold to filter DEGs according their p-value corrected
- **logFC.cut** is a threshold to filter DEGs according their logFC
- **diffmean.cut** `diffmean.cut` for DMR
- **samplesType** `samplesType`
- **colDescription** `colDescription`
- **gset** `gset`
- **gsetFile** `gsetFile`
Value
result matrix from differential phenotype analysis

Examples
dataDEGs <- DPA(dataFilt = dataFilt, dataType = "Gene expression")

<table>
<thead>
<tr>
<th>EAGenes</th>
<th>Information about genes</th>
</tr>
</thead>
</table>

Description
A data set containing the following data:

Usage
data(EAGenes)

Format
A 20038x5 matrix

Details
- EAGenes matrix with 20038 rows (genes) and five columns "ID" "Gene" "Description" "Location" "Family"

Value
a 20038x5 matrix

Description
This function carries out the functional enrichment analysis (FEA)

Usage
FEA(BPname = NULL, DEGsmatrix)

Arguments
BPname biological process such as "proliferation of cells", "ALL" (default) if FEA should be carried out for all 101 biological processes
DEGsmatrix DEGsmatrix output from DEA such as dataDEGs

Value
matrix from FEA
GDCprojects

Examples

dataDEGs <- DPA(dataFilt = dataFilt, dataType = "Gene expression")
dataFEA <- FEA(DEGsmatrix = dataDEGs)

GDCprojects

Description
A character vector of GDC projects:

Usage
data(GDCprojects)

Format
A character vector of 39 elements

Details
• character vector for GDC projects.

Value
character vector of 39 elements

geneInfo

Description
A data set containing the following data:

Usage
data(geneInfo)

Format
A data frame with 20531 rows and 3 variables

Details
• geneInfo matrix with 20531 rows (genes) and 3 columns "geneLength" "gcContent" "chr"

Value
a 20531x3 matrix
getDataGEO

GEO_TCGAtab  Information on GEO data (and overlap with TCGA)#’ A data set containing the following data:

Description

• GEO_TCGAtab a 18x12 matrix that provides the GEO data set we matched to one of the 18 given TCGA cancer types

Usage

data(GEO_TCGAtab)

Format

A 101x3 matrix

Value

a 101x3 matrix

gDataGEO  getDataGEO

description

This function retrieves and prepares GEO data

Usage

gDataGEO(GEOobject = "GSE39004", platform = "GPL6244", TCGAtumor = NULL)

Arguments

GEOobject  GEOobject
platform  platform
TCGAtumor  tumor name

Value

return GEO gset

Examples

## Not run:
dataGEO <- getDataGEO(GEOobject = "GSE20347", platform = "GPL571")

## End(Not run)
Description

This function retrieves and prepares TCGA data

Usage

```
getDataTCGA(cancerType, dataType, directory, cor.cut = 0.6, qnt.cut = 0.25, 
nSample, stage = "ALL", subtype = 0, samples = NULL, seed = 12345)
```

Arguments

- `cancerType`: select cancer type for which analysis should be run. panCancer for all available cancer types in TCGA. Defaults to panCancer
- `dataType`: is dataType such as gene expression, cnv, methylation etc.
- `directory`: Directory/Folder where the data was downloaded. Default: GDCdata
- `cor.cut`: cor.cut
- `qnt.cut`: qnt.cut
- `nSample`: nSample
- `stage`: stage
- `subtype`: subtype
- `samples`: samples
- `seed`: set to get same result

Value

returns filtered TCGA data

Examples

```r
## Not run:
dataFilt <- getDataTCGA(cancerType = "LUAD", 
 dataType = "Gene expression", directory = "data", nSample = 4)
## End(Not run)
```
### GRN

**Generate network**

#### Description

This function carries out the gene regulatory network inference using parmigene

#### Usage

```
GRN(TFs, DEGsmatrix, DiffGenes = FALSE, normCounts, kNearest = 3,
    nGenesPerm = 10, nBoot = 10, seed = 12345)
```

#### Arguments

- **TFs**: a vector of genes.
- **DEGsmatrix**: DEGsmatrix output from DEA such as `dataDEGs`.
- **DiffGenes**: if TRUE consider only `diff.expr` genes in GRN.
- **normCounts**: is a matrix of gene expression with genes in rows and samples in columns.
- **kNearest**: the number of nearest neighbors to consider to estimate the mutual information.
- **nGenesPerm**: nGenesPerm
- **nBoot**: nBoot
- **seed**: set to get same result Must be less than the number of columns of `normCounts`.

#### Value

an adjacent matrix

#### Examples

```
dataDEGs <- DEGsmatrix
dataGRN <- GRN(TFs = rownames(dataDEGs)[1:100],
                 DEGsmatrix = dataDEGs,
                 DiffGenes = TRUE,
                 normCounts = dataFilt)
```

### GSEA

**GSEA**

#### Description

This function carries out the GSEA enrichment analysis.

#### Usage

```
GSEA(DEGsmatrix, top, plot = FALSE)
```
knownDriverGenes

Arguments

DEGs\text{matrix} \quad \text{DEGsmatrix output from DEA such as dataDEGs}

\text{top} \quad \text{is the number of top BP to plot}

\text{plot} \quad \text{if TRUE return a GSEA's plot}

Value

return GSEA result

Examples

dataDEGs <- DEGsmatrix
# dataFEA <- GSEA(DEGsmatrix = dataDEGs)

\hline
\text{knownDriverGenes} \quad \text{Information on known cancer driver gene from COSMIC}
\hline

Description

A data set containing the following data:

Usage

data(knownDriverGenes)

Format

A 101x3 matrix

Details

• TSG known tumor suppressor genes
• OCG known oncogenes

Value

a 101x3 matrix
**listMoonlight**  
*Output list from Moonlight*

**Description**  
A list containing the following data:

**Usage**  
```
data(listMoonlight)
```

**Format**  
A large list with 5 elements

**Details**  
- listMoonlight output from moonlight’s pipeline containing dataDEGs, dataURA, listCandidates

**Value**  
output from moonlight pipeline

---

**LPA**  
*LPA*

**Description**  
This function carries out the literature phenotype analysis (LPA)

**Usage**  
```
LPA(dataDEGs, BP, BPlist)
```

**Arguments**  
- `dataDEGs` is output from DEA  
- `BP` is biological process  
- `BPlist` is list of genes annotated in BP

**Value**  
table with number of pubmed that affects, increase or decrease genes annotated in BP

**Examples**  
```
data(DEGsmatrix)  
BPselected <- c("apoptosis")  
BPannotations <- DiseaseList[[match(BPselected, names(DiseaseList))]]$ID  
dataLPA <- LPA(dataDEGs = DEGsmatrix[1:5,],  
                BP = BPselected,  
                BPlist = BPannotations)
```
**moonlight pipeline**

Description

moonlight is a tool for identification of cancer driver genes. This function wraps the different steps of the complete analysis workflow. Providing different solutions:

1. MoonlighR::FEA
2. MoonlighR::URA
3. MoonlighR::PIA

Usage

```r
moonlight(cancerType = "panCancer", dataType = "Gene expression",
directory = "GDCdata", BPname = NULL, cor.cut = 0.6, qnt.cut = 0.25,
Genelist = NULL, fdr.cut = 0.01, logFC.cut = 1, corThreshold = 0.6,
kNearest = 3, nGenesPerm = 10, DiffGenes = FALSE, nBoot = 100,
nTF = NULL, nSample = NULL, thres.role = 0, stage = NULL,
subtype = 0, samples = NULL)
```

Arguments

cancerType select cancer type for which analysis should be run. panCancer for all available cancer types in TCGA. Defaults to panCancer
dataType dataType
directory directory
BPname biological processes to use, if NULL: all processes will be used in analysis, RF for candidate; if not NULL the candidates for these processes will be determined (no learning)
cor.cut cor.cut Threshold
qnt.cut qnt.cut Threshold
Genelist Genelist
fdr.cut fdr.cut Threshold
logFC.cut logFC.cut Threshold
corThreshold corThreshold
kNearest kNearest
nGenesPerm nGenesPerm
DiffGenes DiffGenes
nBoot nBoot
nTF nTF
nSample nSample
thres.role thres.role
stage stage
subtype subtype
samples samples
plotCircos

Value

table with cancer driver genes TSG and OCG.

Examples

dataDEGs <- DPA(dataFilt = dataFilt, dataType = "Gene expression")
# to change with moonlight

Description

This function visualize the plotCircos

Usage

plotCircos(listMoonlight, listMutation = NULL, additionalFilename = NULL, intensityColOCG = 0.5, intensityColTSG = 0.5, intensityColDual = 0.5, fontSize = 1)

Arguments

listMoonlight output Moonlight function
listMutation listMutation
additionalFilename additionalFilename
intensityColOCG intensityColOCG
intensityColTSG intensityColTSG
intensityColDual intensityColDual
fontSize fontSize

Value

no return value, plot is saved

Examples

plotCircos(listMoonlight = listMoonlight, additionalFilename = ".ncancer5")
Description

This function visualize the functional enrichment analysis (FEA)’s barplot.

Usage

```r
plotFEA(dataFEA, topBP = 10, additionalFilename = NULL, height, width,
        offsetValue = 5, angle = 90, xleg = 35, yleg = 5, minY = -5,
        maxY = 10)
```

Arguments

- `dataFEA`: dataFEA
- `topBP`: topBP
- `additionalFilename`: additionalFilename
- `height`: Figure height
- `width`: Figure width
- `offsetValue`: offsetValue
- `angle`: angle
- `xleg`: xleg
- `yleg`: yleg
- `minY`: minY
- `maxY`: maxY

Value

No return value, FEA result is plotted.

Examples

```r
dataFEA <- FEA(DEGsmatrix = DEGsmatrix)
plotFEA(dataFEA = dataFEA, additionalFilename = "_example", height = 20, width = 10)
```
plotNetworkHive: Hive network plot

Description
This function visualizes the GRN as a hive plot

Usage
plotNetworkHive(dataGRN, namesGenes, thres, additionalFilename = NULL)

Arguments
- dataGRN: output GRN function
- namesGenes: list TSG and OCG to define axes
- thres: threshold of edges to be included
- additionalFilename: additionalFilename

Value
no results Hive plot is executed

Examples
```r
data(knownDriverGenes)
data(dataGRN)
plotNetworkHive(dataGRN = dataGRN, namesGenes = knownDriverGenes, thres = 0.55)
```

plotURA: Upstream regulatory analysis heatmap plot

Description
This function visualizes the URA in a heatmap

Usage
plotURA(dataURA, additionalFilename = "URAplot")

Arguments
- dataURA: output URA function
- additionalFilename: figure name

Value
heatmap
**Examples**

```r
data(dataURA)
dataDual <- PRA(dataURA = dataURA,
BPname = c("apoptosis", "proliferation of cells"),
thres.role = 0)
plotURA(dataURA = dataURA[c(names(dataDual$TSG), names(dataDual$OCG)),],
additionalFilename = ".example")
```

---

**PRA**

*Pattern Recognition Analysis (PRA)*

---

**Description**

This function carries out the pattern recognition analysis

**Usage**

```r
PRA(dataURA, BPname, thres.role = 0, seed = 12345)
```

**Arguments**

- `dataURA`: output URA function
- `BPname`: BPname
- `thres.role`: thres.role
- `seed`: seed value

**Value**

returns list of TSGs and OCGs when biological processes are provided, otherwise a randomForest based classifier that can be used on new data

**Examples**

```r
data(dataURA)
dataDual <- PRA(dataURA = dataURA,
BPname = c("apoptosis", "proliferation of cells"),
thres.role = 0)
```

---

**tabGrowBlock**

*Information growing/blocking characteristics for 101 selected biological processes*

---

**Description**

A data set containing the following data:

**Usage**

```r
data(tabGrowBlock)
```
URA

Format
A 101x3 matrix

Details
- tabGrowBlock matrix that defines if a process is growing or blocking cancer development, for each 101 biological processing

Value
a 101x3 matrix

URA Upstream Regulator Analysis

Description
This function carries out the upstream regulator analysis

Usage
URA(dataGRN, DEGsmatrix, BPname, nCores = 1)

Arguments
dataGRN output GNR function
DEGsmatrix output DPA function
BPname biological processes
nCores number of cores to use

Value
an adjacent matrix

Examples
dataDEGs <- DEGsmatrix
dataGRN <- GRN(TFs = rownames(dataDEGs)[1:100], DEGsmatrix = dataDEGs, DiffGenes = TRUE, normCounts = dataFilt)
dataURA <- URA(dataGRN = dataGRN, DEGsmatrix = dataDEGs, BPname = c("apoptosis", "proliferation of cells"))
Index

∗Topic datasets
  dataFilt, 3
  dataGRN, 3
  dataURA, 4
  DEGsmatrix, 4
  DiseaseList, 5
  EAGenes, 6
  GDCprojects, 7
  geneInfo, 7
  GEO_TCGAtab, 8
  knownDriverGenes, 11
  listMoonlight, 12
  tabGrowBlock, 17

  dataFilt, 3
  dataGRN, 3
  dataURA, 4
  DEGsmatrix, 4
  DiseaseList, 5
  DPA, 5

  EAGenes, 6

  FEA, 6

  GDCprojects, 7
  geneInfo, 7
  GEO_TCGAtab, 8
  getDataGEO, 8
  getDataTCGA, 9
  GRN, 10
  GSEA, 10

  knownDriverGenes, 11

  listMoonlight, 12
  LPA, 12

  moonlight, 13
  MoonlightR, 14
  MoonlightR-package (MoonlightR), 14

  plotCircos, 14
  plotFEA, 15
  plotNetworkHive, 16

  plotURA, 16
  PRA, 17

  tabGrowBlock, 17

  URA, 18

19