Package 'GeneNetworkBuilder'

October 7, 2014

Type Package

Version 1.6.1

Date 2014-7-22

Title Build Regulatory Network from ChIP-chip/ChIP-seq and Expression Data

Author Jianhong Ou and Lihua Julie Zhu

Maintainer Jianhong Ou <jianhong.ou@umassmed.edu>

Imports plyr, graph

Depends R (>= 2.15.1), Rcpp (>= 0.9.13), graph

Suggests RUnit, BiocGenerics, Rgraphviz, XML, RCytoscape, RBGL

LinkingTo Rcpp

Description Appliation for discovering direct or indirect targets of transcription factors using ChIP-chip or ChIP-seq, and microarray or RNA-seq gene expression data. Inputting a list of genes of potential targets of one TF from ChIP-chip or ChIP-seq, and the gene expression results, GeneNetworkBuilder generates a regulatory network of the TF.

License GPL (>= 2)

Lazyload yes

biocViews Sequencing, Microarray, GraphAndNetwork

R topics documented:

`buildNetwork` *construct the regulatory network*

Description

get all the connections of interesting genes from regulatory map.

Usage

```
buildNetwork(TFbindingTable, interactionmap, level=3)
```

Arguments

`TFbindingTable`	a matrix or data.frame with interesting genes. Column names must be 'from', 'to'
`interactionmap`	Transcription regulatory map. Column names of interactionmap must be 'from','to'
`level`	Depth of node path

Value

a dataframe or matrix of all the connections of interesting genes

Examples

```
data("ce.interactionmap")
data("example.data")
xx<-buildNetwork(example.data$ce.bind, ce.interactionmap, level=2)
```

ce.IDsMap — *map file for converting gene name or sequence name of* Caenorhabditis elegans *to wormbase identifier*

Description

map file for converting gene name or sequence name of *Caenorhabditis elegans* to wormbase identifier

Usage

```
data(ce.IDsMap)
```

Format

character vector

Details

character vecotr with gene name or sequence name as names and wormbase identifier as values.

Examples

```
data(ce.IDsMap)
head(ce.IDsMap)
```

ce.interactionmap — *transcript regulatory map of* Caenorhabditis elegans

Description

transcript regulatory map of *Caenorhabditis elegans*

Usage

```
data(ce.interactionmap)
```

Format

dataframe

Details

transcript regulatory map of *Caenorhabditis elegans* is generated using databases edgedb and microCosm Targets.

Examples

```
data(ce.interactionmap)
head(ce.interactionmap)
```

`ce.mapIDs` *map file for converting from wormbase identifier to* Caenorhabditis elegans *gene name*

Description

map file for converting from wormbase identifier to *Caenorhabditis elegans* gene name

Usage

```
data(ce.mapIDs)
```

Format

character vector

Details

character vecotr with wormbase identifier as names and gene name as values.

Examples

```
data(ce.mapIDs)
head(ce.mapIDs)
```

`ce.miRNA.map` *micro RNA of* Caenorhabditis elegans

Description

micro RNA of *Caenorhabditis elegans*

Usage

```
data(ce.miRNA.map)
```

Format

dataframe

Details

The first column is wormbase identifier. And the second column is miRNA names.

Examples

```
data(ce.miRNA.map)
head(ce.miRNA.map)
```

`convertID` *convert gene IDs by id map*

Description

For same gene, there are multple gene alias. In order to eliminate the possibility of missing any connections, convert the gene symbols to unique gene ids is important. This function can convert the gene symbols to unique ids and convert it back according a giving map.

Usage

```
convertID(x,IDsMap,ByName=c("from","to"))
```

Arguments

`x`	a matrix or dataframe contain the columns to be converted.
`IDsMap`	a character vector of the identifier map
`ByName`	the column names to be converted

Value

a matrix or dataframe with converted gene IDs

Examples

```
data("ce.IDsMap")
bind<-cbind(from="daf-16", to=c("fkh-7", "hlh-13", "mxl-3", "nhr-3", "lfi-1"))
convertID(toupper(bind), ce.IDsMap, ByName=c("from", "to"))
```

`example.data` *example datasets for documentation*

Description

example.data is a data list of example datasets. There is a dataset example.microarrayData, which is the example of gene expression data of a gene-chip result of *C.elegans*. Dataset example.data$ce.bind is a TF binding matrix of ChIP-chip experiment of *C.elegans*. Dataset example.data$cd.exprData is expression data of a gene-chip result of *C.elegans*. Dataset example.data$hs.bind is a TF binding matrix of ChIP-chip experiment of *H.sapiens*. Dataset example.data$hs.exprData is expression data of a combination of a gene-chip result and a RNA-SEQ result of *H.sapiens*.

Usage

```
data(example.data)
```

Format

dataframe

Details

The dataset example.microarrayData contains columns: ID, logFC, AveExpr, t, P.Value, adj.P.Val, B, genes and symbols. The columns of ID, logFC and symbols are required by GeneNetworkBuilder. The dataset example.data$hs.bind contains columns: ID, symbols, logFC and P.Value. The dataset example.data$hs.exprData contains columns: from and to.

Examples

```
data(example.data)
names(example.data)
head(example.data$example.microarrayData)
head(example.data$ce.bind)
head(example.data$ce.exprData)
head(example.data$hs.bind)
head(example.data$hs.exprData)
```

---

filterNetwork *filter the regulatory network table by expression profile*

Description

verify every nodes in the regulatory network by expression profile.

Usage

```
filterNetwork(rootgene, sifNetwork, exprsData, mergeBy="symbols",
miRNAlist, remove_miRNA=FALSE, tolerance=0, cutoffPVal=0.01, cutoffLFC=0.5,
minify=TRUE, miRNAtol=FALSE)
```

Arguments

rootgene	name of root gene. It must be the ID used in xx regulatory network
sifNetwork	Transcription regulatory network table. Column names of xx must be 'from','to'
exprsData	dataset of expression comparison data, which should contain column logFC and column given by exprsDataByName
mergeBy	The column name contains ID information used to merge with 'to' column of sifNetwork in exprsData
miRNAlist	vector of microRNA ids.

remove_miRNA	remove miRNA from the network or not. Bool value, TRUE or FALSE
tolerance	maximum number of unverified nodes in each path
cutoffPVal	cutoff p value of valid differential expressed gene/miRNA
cutoffLFC	cutoff log fold change value of a valid differential expressed gene/miRNA
minify	Only keep the best path if multiple paths exists for single node? Bool value, TRUE or FALSE
miRNAtol	take miRNA expression into account for tolerance calculation. Bool value, TRUE or FALSE

Value

a dataframe of filtered regulatory network by expression profile

Examples

```
data("ce.miRNA.map")
data("example.data")
data("ce.interactionmap")
data("ce.IDsMap")
sifNetwork<-buildNetwork(example.data$ce.bind, ce.interactionmap, level=2)
cifNetwork<-filterNetwork(rootgene=ce.IDsMap["DAF-16"], sifNetwork=sifNetwork,
exprsData=uniqueExprsData(example.data$ce.exprData, "Max", condenseName=logFC),
mergeBy="symbols",
miRNAlist=as.character(ce.miRNA.map[ , 1]), tolerance=1)
```

---

hs.IDsMap *map file for converting gene name or sequence name of* Homo sapiens *to Entrez identifier*

Description

map file for converting gene name or sequence name of *Homo sapiens* to Entrez identifier

Usage

```
data(hs.IDsMap)
```

Format

character vector

Details

character vecotr with gene name as names and Entrez identifier as values.

Examples

```
data(hs.IDsMap)
head(hs.IDsMap)
```

hs.interactionmap *transcript regulation map of* Homo sapiens

Description

transcript regulation map of *Homo sapiens*

Usage

```
data(hs.interactionmap)
```

Format

dataframe

Details

transcript regulatory map of *Homo sapiens* is generated using databases FANTOM, mirGen and microCosm Targets.

Examples

```
data(hs.interactionmap)
head(hs.interactionmap)
```

hs.mapIDs *map file for converting from Entrez identifier to* Homo sapiens *gene name*

Description

map file for converting from Entrez identifier to *Homo sapiens* gene name

Usage

```
data(hs.mapIDs)
```

Format

character vector

Details

character vecotr with Entrez identifier as names and gene name as values.

Examples

```
data(hs.mapIDs)
head(hs.mapIDs)
```

hs.miRNA.map *micro RNA of* Homo sapiens

Description

micro RNA of *Homo sapiens*

Usage

```
data(hs.miRNA.map)
```

Format

dataframe

Details

The first column is entrez identifier. And the second column is miRNA names.

Examples

```
data(hs.miRNA.map)
head(hs.miRNA.map)
```

polishNetwork *generate an object of grahpNEL to represent the regulation network*

Description

generate an object of grahpNEL to represent the regulation network. Each node will has three attributes: size, borderColor and fill.

Usage

```
polishNetwork(cifNetwork,
nodesDefaultSize=48, useLogFCAsWeight=FALSE,
nodecolor=colorRampPalette(c("green", "yellow", "red"))(5), nodeBg="white",
nodeBorderColor=list(gene=darkgreen,miRNA=darkblue),
edgelwd=0.25, ...)
```

Arguments

cifNetwork	dataframe used to draw network graph. column names of cifNetwork must contain 'from', 'to', 'logFC' and 'miRNA'
nodesDefaultSize	nodes default size
useLogFCAsWeight	how to determine the weights for each nodes. If TURE, use logFC value as weight. If FALSE, use constant 1 as weight.
nodecolor	a character vector of color set. The node color will be mapped to color set by log fold change
nodeBg	background of node
nodeBorderColor	a list of broder node color set. nodeBorderColor's element must be gene and miRNA
edgelwd	the width of edge
...	any parameters can be passed to graph.par

Value

An object of graphNEL class of the network

Examples

```
data("ce.miRNA.map")
data("example.data")
data("ce.interactionmap")
data("ce.IDsMap")
sifNetwork<-buildNetwork(example.data$ce.bind, ce.interactionmap, level=2)
cifNetwork<-filterNetwork(rootgene=ce.IDsMap["DAF-16"], sifNetwork=sifNetwork,
exprsData=uniqueExprsData(example.data$ce.exprData, "Max", condenseName=logFC),
mergeBy="symbols",
miRNAlist=as.character(ce.miRNA.map[ , 1]), tolerance=1)
gR<-polishNetwork(cifNetwork)
## g1<-Rgraphviz::layoutGraph(gR)
## renderGraph(g1)
## cat(saveXML(toGXL(g1)),sep="\n")
```

---

uniqueExprsData *unique the microarray data*

Description

get unique the microarray data for each gene id.

Usage

```
uniqueExprsData(exprsData, method=Max, condenseName=logFC)
```

Arguments

exprsData	dataset of expression comparison data
method	method must be Max, Median or Min
condenseName	column names to be condensed

Value

a dataframe of expression data without duplicates

Examples

```
data("example.data")
example.microarrayData<-uniqueExprsData(example.data$example.microarrayData,
method="Max", condenseName=logFC)
```

Index