Package ‘CHRONOS’

December 21, 2016

Version 1.2.0
Date 2016-03-21
Title CHRONOS: A time-varying method for microRNA-mediated sub-pathway enrichment analysis
Author Aristidis G. Vrahatis, Konstantina Dimitrakopoulou, Panos Balomenos
Maintainer Panos Balomenos <balomenos@upatras.gr>
Description A package used for efficient unraveling of the inherent dynamic properties of pathways. MicroRNA-mediated subpathway topologies are extracted and evaluated by exploiting the temporal transition and the fold change activity of the linked genes/microRNAs.
Depends R (>= 3.3)
SystemRequirements Java version >= 1.7, Pandoc
License GPL-2
Repository Bioconductor
Date/Publication
NeedsCompilation no
LazyLoad yes
Imports XML, RCurl, RBGL, parallel, foreach, doParallel, openxlsx, circlize, graph, stats, utils, grDevices, graphics, biomaRt
Suggests RUnit, BiocGenerics, knitr
VignetteBuilder knitr
biocViews SystemsBiology, GraphAndNetwork, Pathways, KEGG

R topics documented:

CHRONOSrun ................................................................. 2
convertMiRNAExpressions .............................................. 3
convertNomenclature ..................................................... 4
createPathwayGraphs ...................................................... 5
downloadKEGGPathwayList ............................................. 6
downloadMiRecords ........................................................ 7
downloadPathways ......................................................... 8
Default run of CHRONOS

Usage

CHRONOSrun(mRNAexp, mRNAlabel, miRNAexp, pathType, subType, measures, thresholds, org, export, verbose, miRNAinteractions)

Arguments

mRNAexp mRNA expressions filename located in CHRONOS/extdata/Input
mRNAlabel mRNA nomenclature (for supported types see convertNomenclature)
miRNAexp miRNA expressions filename located in CHRONOS/extdata/Input
pathType Pathway type (‘Metabolic’, ‘Non-Metabolic’, ‘All’ or vector of pathway ids)
subType Subpathway type (‘Linear’, ‘Non-Linear’, ‘All’)
measures Include subpathway structural and functional aspects (‘TRUE’, ‘FALSE’)
thresholds Subscore, mirscore and p-value thresholds
   c(‘pvalue’=pvalue, ‘subscore’=subscore, ‘mirscore’=mirscore)
org KEGG organism identifier
export Export file type (‘.xlsx’, ‘.txt’)
verbose Show informative messages (TRUE/FALSE).
miRNAinteractions Edgelist of miRNA-mRNA interactions.

Details

• Imports gene and miRNA expressions from CHRONOS/extdata/Input/<mRNAexpFile>.txt
  and CHRONOS/extdata/Input/<miRNAexpFile>.txt
• Downloads all available pathways for the specified organism from KEGG.
• Creates pathway graphs from downloaded KGML files.
• Extracts linear subpathways from metabolic and non metabolic graphs.
• Extracts non linear subpathways from metabolic and non metabolic graphs.
• Downloads miRecords miRNA-mRNA interactions.
• Scores and evaluates (linear and non linear) subpathways to extract significant results.
• Organism identifier.
• Visualizes most the significant results (’.xlsx’ or ‘.txt’).
• Display informative messages (TRUE/FALSE).
• User-defined miRNA-mRNA interactions can be supplied in the form of an edgelist with two columns. If no such information is available, a missing or a NULL argument forces the use of default interactions by using `downloadMiRecords`.

Value

Examples

# Default run

load(system.file('extdata', 'Examples//data.RData', package='CHRONOS'))

res <- CHRONOSrun( mRNAexp=mRNAexpr,
                   mRNAlabel='entrezgene',
                   miRNAexp=miRNAexpr,
                   pathType=c('04915', '04917', '04930', '05031'),
                   org='hsa',
                   subType='Linear',
                   thresholds=c('subScore'=0.4, 'mirScore'=0.4),
                   miRNAinteractions=miRNAinteractions)

convertMiRNANomenclature

`Conform miRNA annotations to the ones currently used by miRecords.`

Description

Conform miRNA annotations to the ones currently used by miRecords.

Usage

`convertMiRNANomenclature(org, miRNAs, update)`

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>org</td>
<td>KEGG organism identifier.</td>
</tr>
<tr>
<td>miRNAs</td>
<td>Vector of miRNAs identifiers.</td>
</tr>
<tr>
<td>update</td>
<td>Update annotation mapper with latest annotation changes.</td>
</tr>
</tbody>
</table>

Details

Determine which miRNAs are incompatible with miRecords annotations and retrieve the suitable ones from www.mirbase.org.
Value

Examples

data <- c('hsa-let-7g-5p', 'hsa-miR-154-5p', 'hsa-miR-376b-3p')

convertMiRNANomenclature(org='hsa', miRNAs=data)

convertNomenclature  Convert genes identifier nomenclature.

Description

Convert genes identifier nomenclature.

Usage

convertNomenclature(ids, org, from, to)

Arguments

ids  Vector of gene identifiers
org  KEGG organism identifier
from Initial identifier type
to  A vector of final identifier types

Details

EntrezGene ID        'entrezgene'
Ensembl Gene ID      'ensembl_gene_id'
Ensemble Transcript ID 'ensembl_transcript_id'
Ensemble Protein ID  'ensembl_peptide_id'
HGNC ID             'hgnc_id'
HGNC Symbol          'hgnc_symbol'
HGNC Transcript name 'hgnc_transcript_name'
Refseq mRNA ID       'refseq_mrna'
Refseq Protein ID    'refseq_peptide'
UniProt/Swissprot Accession 'uniprot_swissprot_accession'
UniProt/Swissprot ID 'uniprot_swissprot'
UniGene ID           'unigene'
UniProt Genename ID  'uniprot_genename'

Value

Vector of converted gene identifiers
createPathwayGraphs

Examples

# Identifiers to be converted
ids <- c('5091', '5105')

# Convert to HGNC ID, Ensembl Gene ID and UniProt Genename ID
from <- 'entrezgene'
to <- c('hgnc_symbol', 'ensembl_gene_id', 'uniprot_genename')

## Not run: res <- convertNomenclature(ids=ids, org='hsa', from=from, to=to)

createPathwayGraphs

Convert KEGG Pathways to Gene-Gene Network Graphs.

Description

Convert KEGG Pathways to Gene-Gene Network Graphs.

Usage

createPathwayGraphs(org, pathways, edgeTypes, doubleEdges, choice, groupMode)

Arguments

org KEGG organism identifier.
pathways Vector of KEGG pathway identifiers.
edgeTypes Vector of edge types mappings.
doubleEdges Specify which edgeTypes should be considered bidirectional.
choice Create metabolic graph either by using relations or reactions from KGML file ('reactions', 'relations')
groupMode 'expand' to consider each group member a node, or 'collapse' to consider all components' genes as a node

Details

KEGG pathways consist of nodes each one containing one or more genes. Thus, two kinds of adjacency matrices are created. The compact adjacency matrix retains the groupings and stores edge types between genes and genes, genes and groups of genes or between group of genes. The expanded adjacency matrix stores edge type information between individual genes.

Value

A list containing a list of compact adjacency matrices, a list of expanded adjacency matrices, and list detailing all nodes, edges and interaction types.

References

Examples

# Download Insulin Signaling Pathway
pathways <- c("04915", "04917", "04930", "05031")
paths <- downloadPathways(org="hsa", pathways=pathways)

# Create pathway graph
graphs <- createPathwayGraphs(org='hsa', pathways=paths)

downloadKEGGPathwayList

Retrieve all available pathways for an organism.

Description

Retrieve all available pathways for an organism.

Usage

downloadKEGGPathwayList(org)

Arguments

org KEGG organism identifier.

Details

-

Value

Data frame of pathway ids and names.

References


Examples

# Load extracted linear subpathways from toy data
load(system.file("extdata", 'Examples//data.RData', package='CHRONOS'))

# Retrieve all available hsa pathways
## Not run: pathways <- downloadKEGGPathwayList(org='hsa')
**downloadMiRecords**  
*Download miRNA-mRNA interactions for an organism.*

**Description**

Download miRNA-mRNA interactions for an organism.

**Usage**

```r
downloadMiRecords(org, pn, update, databases)
```

**Arguments**

- `org`  
  KEGG organism identifier.

- `pn`  
  Number of databases that verify miRNA-mRNA interactions.

- `update`  
  Download preprocessed data (update=FALSE) or new data from miRecords (update=TRUE).

- `databases`  
  Specify which miRNA-mRNA interaction databases will be used.

**Details**

miRecords is a resource for animal miRNA-target interactions. The Predicted Targets component of miRecords is an integration of predicted miRNA targets produced by 11 established miRNA target prediction tools, namely DIANA-microT, MicroInspector, miRanda, MirTarget2, miTarget, NBmiRTar, PicTar, PITA, RNA22, RNAhybrid, and TargetScan/TargetScanS.

**Value**

Downloaded data is stored in CHRONOS/extdata/Downloads/miRecords/<org>/miRNATargets.RData

**References**

- [http://c1.accurascience.com/miRecords](http://c1.accurascience.com/miRecords)

**Examples**

```r
# Load extracted linear subpathways from toy data
load(system.file('extdata', 'Examples//data.RData', package='CHRONOS'))

## Not run: downloadMiRecords(org='hsa', pn=5, update=FALSE, databases='All')
```
downloadPathways  Download KEGG pathways in KGML format.

Description

Download KEGG pathways in KGML format.

Usage

downloadPathways(org, pathways)

Arguments

org
  KEGG organism identifier

pathways
  Download pathways for specified organism:

  'All'  All organism pathways

  'Metabolic'  Metabolic pathways

  'Non-Metabolic'  Non metabolic pathways

  <vector of indexes>  Using indexes from downloadKEGGPathwayList

  <vector of names>  Using pathway identifiers (i.e. c('00010', '00020'))

Details

KEGG (Kyoto Encyclopedia of Genes and Genomes) is a database resource for understanding high-level functions and utilities of the biological system such as the cell, the organism and the ecosystem, from molecular-level information, especially large-scale molecular datasets generated by genome sequencing and other high-throughput experimental technologies.

Files are downloaded in CHRONOS/extdata/Downloads/KEGG/<org> folder. Downloading is skipped for existing files.

Value

Downloaded data is stored in CHRONOS/extdata/Downloads/KEGG/<org>

References


Examples

# View all available hsa pathways
pathways <- downloadKEGGPathwayList(org='hsa')

# Download pathway KGML files
pathways <- c('04915', '04917', '04930', '05031')

pathways <- downloadPathways(org='hsa', pathways=pathways)
**extractLinearSubpathways**

*Linear subpathway extraction from pathway graphs*

### Description
Linear subpathway extraction from pathway graphs

### Usage

```r
extractLinearSubpathways(graphs, pathways, a, b, filter, export, groupMode, verbose)
```

### Arguments
- **graphs**: Pathway graphs as returned from `createPathwayGraphs`.
- **pathways**: The subset of pathways from whom subpathways are to be extracted. If missing, all pathway graphs are used.
- **a**: Minimum subpathway length.
- **b**: Maximum subpathway length.
- **filter**: Filter the subpaths with user genes (TRUE).
- **export**: Exports subpaths in CHRONOS/extdata/Output/Subpaths/Linear/<org> folder. Available formats are `.txt` and/or `.RData`.
- **groupMode**: Expand paralogues (`expand`) or collapse them to a single entry (`collapse`).
- **verbose**: Display informative messages (TRUE). Requires previous execution of `importExpressions`.

### Details
Subpath filtering supports the removal of subpaths that have at least one member not belonging to the set of user supplied genes. These genes are extracted from the user’s mRNA expressions matrix. Thus, the execution of `importExpressions` is a prerequisite.

To extract linear subpathways from a pathway graph, all possible start and end nodes are considered. A start node has only outgoing edges while an end node only has incoming edges. For each such pair, all linear subpathways are found by traversing the corresponding graph. Since the initial pathway graph’s nodes contain one or more genes, resulting subpathways consist of bins of one or more genes. These subpaths are expanded to subpathways with one gene per bin in order to obtain usable subpathways.

### Value
Returns a list consisting of
- A matrix of linear subpathways (subpaths)
- A list of processed pathway graphs adjacency matrices (adjMats)
- A list of processed pathway genes and interactions between them (lexicon)
Examples

```r
# Load pathway graphs from toy data
load(system.file("extdata", 'Examples//data.RData', package='CHRONOS'))

# Extract linear subpathways
linSubs <- extractLinearSubpathways(graphs=graphs)
```

extractNonLinearSubpathways

Non linear subpathway extraction from pathway graphs

Description

Non linear subpathway extraction from pathway graphs

Usage

```r
extractNonLinearSubpathways(graphs, pathways, a, b, k, filter, groupMode, export, verbose)
```

Arguments

- `graphs` Pathway graphs as returned from `createPathwayGraphs`.
- `pathways` The subset of pathways from whom subpathways are to be extracted. If missing, all pathway graphs are used.
- `a` Minimum subpathway length.
- `b` Maximum subpathway length.
- `k` Clique size.
- `filter` Filter the subpaths with user genes (TRUE).
- `groupMode` Expand paralogues ('expand') or collapse them to a single entry ('collapse').
- `export` Exports subpaths in CHRONOS/extdata/Output/Subpaths/Non-Linear/ <org> folder. Available formats are '.txt' and/or '.RData'.
- `verbose` Display informative messages (TRUE).

Value

Returns a list consisting of

- A matrix of linear subpathways (subpaths)
- A list of processed pathway graphs adjacency matrices(adjMats)
- A list of processed pathway genes and interactions between them (lexicon)

To extract non linear subpaths from a pathway graph, all interactions between nodes of belonging to k-cliques are found. The ones that correspondTo extract non linear subpaths from a pathway graph, all interactions between nodes of belonging to k-cliques are found. The ones that correspond to actual interactions between genes make up the non linear subpath.
getEdgeTypes

Examples

# Load pathway graphs from toy data
load(system.file('extdata', 'Examples//data.RData', package='CHRONOS'))

# Extract linear subpathways
nliSubs <- extractNonLinearSubpathways(graphs=graphs)

getEdgeTypes

Map various types of gene-gene interactions in KGML files to edge types in corresponding pathway graphs.

Description

Map various types of gene-gene interactions in KGML files to edge types in corresponding pathway graphs.

Usage

getEdgeTypes(type)

Arguments

type A vector of interaction types.

Details

Edge types

activation 1 inhibition 2 apathetic 3 no interaction 4

Default interaction - edge type mapping

<table>
<thead>
<tr>
<th>Type</th>
<th>Edge Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>unknown 3</td>
<td>activation 1</td>
</tr>
<tr>
<td>inhibition 2</td>
<td>binding/association 3</td>
</tr>
<tr>
<td>expression 1</td>
<td>repression 2</td>
</tr>
<tr>
<td>phosphorylation 3</td>
<td>dephosphorylation 3</td>
</tr>
<tr>
<td>ubiquitination 3</td>
<td>dissociation 3</td>
</tr>
<tr>
<td>indirect effect 3</td>
<td>state change 3</td>
</tr>
<tr>
<td>compound 3</td>
<td>hidden compound 3</td>
</tr>
<tr>
<td>missing interaction 3</td>
<td>activation_phosphorylation 1</td>
</tr>
<tr>
<td>activation_dephosphorylation 1</td>
<td>activation_ubiquitination 1</td>
</tr>
<tr>
<td>activation_indirect effect 1</td>
<td>activation_binding/association 1</td>
</tr>
<tr>
<td>activation_inhibition 3</td>
<td>activation_methylation 1</td>
</tr>
<tr>
<td>inhibition_phosphorylation 2</td>
<td>inhibition_dephosphorylation 2</td>
</tr>
<tr>
<td>inhibition_ubiquitination 2</td>
<td>inhibition_indirect effect 2</td>
</tr>
<tr>
<td>inhibition_binding/association 2</td>
<td>inhibition_expression 2</td>
</tr>
<tr>
<td>inhibition_methylation 2</td>
<td>compound_expression 1</td>
</tr>
<tr>
<td>compound_methylation 1</td>
<td>compound_methylation 1</td>
</tr>
<tr>
<td>compound_methylation 2</td>
<td>compound_methylation 2</td>
</tr>
</tbody>
</table>
### importExpressions

<table>
<thead>
<tr>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>If an interaction type has been supplied, the corresponding edge types are returned. If not, the complete mapping is returned.</td>
</tr>
</tbody>
</table>

### Examples

#### Example 1

# Retreive edge types for phosphorylation and dephosphorylation.
getEdgeTypes(c(7,8))

#### Example 2

# Returns a data frame containing the interaction - edge type mapper.
types <- getEdgeTypes()

# Set phosphorylation to inhibition.
types[8,2] <- 2

---

### importExpressions

**Import gene and miRNA expressions from**

### Description

Import gene and miRNA expressions from

### Usage

importExpressions(data, type, sep, org, mRNAomnomenclature)
Arguments

**data**
Expressions data filename or matrix.

**type**
Expressions data type. (or mRNA expressions, type=<nomenType>). Available gene expression nomenclature can be found in `convertNomenclature`. For miRNA expressions, type='miRNA'.

**sep**
File delimiter.

**org**
KEGG organism identifier

**mRNAnomenclature**
Nomenclature of user’s mRNA expressions

Details

- Import gene expressions data from CHRONOS/extdata/Input/<userFile>.txt or a supplied matrix.
- Import miRNA expressions data from CHRONOS/extdata/Input/<userFile>.txt or a supplied matrix.

Value

Examples

```r
# Example
load(system.file('extdata', 'Examples/data.RData', package='CHRONOS'))
importExpressions(data=mRNAexpr, type='mRNA',
                   mRNAnameclature='entrezgene', sep='\t', org='hsa')
importExpressions(data=miRNAexpr, type='miRNA', sep='\t', org='hsa')
```

---

**pathwayMeasures**

**Pathway structural and functional aspects**

Description

Pathway structural and functional aspects

Usage

`pathwayMeasures(graphs)`

Arguments

**graphs**
Pathway graphs as returned from `createPathwayGraphs`.

Details

Structural and functional aspects of a pathway are calculated in respect to all organism pathways.
**scoreSubpathways**

Value

Matrix with pathness, betweeness centrality and degree values for each gene in the pathway graphs at its columns.

Examples

# Load pathway graphs from toy data
load(system.file('extdata', 'Examples//data.RData', package='CHRONOS'))

# Calculate pathway structural and functional aspects
measures <- pathwayMeasures(graphs)

| scoreSubpathways | Evaluate subpathways using an interacting scoring scheme (IS) for each time point. |

Description

Evaluate subpathways using an interacting scoring scheme (IS) for each time point.

Usage

scoreSubpathways(subpathways, filters, measures, parameters, miRNAinteractions)

Arguments

- **subpathways**
  - Subpaths as returned from `extractLinearSubpathways` and `extractNonLinearSubpathways`.

- **filters**
  - Named vector of filters used for subpathway evaluation. Values denote corresponding thresholds.

- **measures**
  - Subpathway structural and functional aspects as returned from `pathwayMeasures`.

- **parameters**
  - C.K,T parameters of scoring scheme.

- **miRNAinteractions**
  - An edgelist of miRNA-mRNA interactions used to override downloaded interactions from miRecords.

Details

...

Value
subpathwaysKEGGmap

subpathways   High ranking subpathways
subScores     miRNA-subpathway scores
mRNAScores    mRNA-mRNA scores for each subpathway and for each time point
miRNAsOverSubpathway High ranking miRNAs hitting each subpathway
pValues       P-value of each subpathway
filters       Filters used for the evaluation

References


Examples

# Load extracted subpathways from toy data
load(system.file("extdata", "Examples/data.RData", package="CHRONOS"))

# Import mRNA expressions
mRNAexpr <- importExpressions(data=mRNAexpr, type="mRNA", org="hsa")

# Score extracted linear subpathways
filters <- c("subScore"=0.4)
linSubsScored <- scoreSubpathways(subpathways=linSubs, filters=filters)

subpathwayKEGGmap(subpathways, type, openInBrowser)

Create links to KEGG pathway map with highlighted subpathways.

Description

Create links to KEGG pathway map with highlighted subpathways.

Usage

subpathwayKEGGmap(subpathways, type, openInBrowser)

Arguments

subpathways Subpathways as returned by extractLinearSubpathways or extractNonLinearSubpathways
type Subpathway type (Linear, Non-Linear)
openInBrowser Open link in default browser.

Value

Vector of links of KEGG pathway maps.
Examples

# Load extracted linear subpathways from toy data
load(system.file('extdata', 'Examples//data.RData', package='CHRONOS'))

# Opening selected subpathways in default browser
subs <- linSubs$subpaths[1:3, ]

subpathwayKEGGmap(subpathways=subs, type='Linear', openInBrowser=FALSE)

---

subpathwayMiRNAs Create a circulat plot of a subpathway and the miRNAs that target it.

Description

Create a circulat plot of a subpathway and the miRNAs that target it.

Usage

subpathwayMiRNAs(summary, subIdx, timePoints)

Arguments

summary Output from scoreSubpathways
subIdx Subpathway index
timePoints Time points to include in visualization, default to all.

Value

.

Examples

# Load scored subpathways from toy data
load(system.file('extdata', 'Examples//data.RData', package='CHRONOS'))
# Visualize one or more subpathways.
subpathwayMiRNAs(summary=linSubsScored, subIdx=2)
**visualizeResults**

**Visualize results in tabular form (txt, xlsx)**

**Description**

Visualize results in tabular form (txt, xlsx)

**Usage**

`visualizeResults(summary, export, expand, colors, from, to)`

**Arguments**

- **summary**: Evaluation results as returned from `scoreSubpathways`
- **export**: `.xlsx` exports a xlsx file and `.txt` a .txt file.
- **expand**: TRUE if each subpathway member and miRNA belongs to a single cell, FALSE if all subpathway members belong to one cell and miRNAs to another cell.
- **colors**: The color scheme used in subScores heatmap.
- **from**: Primary annotation `convertNomenclature`. Defaults to EntrezGene ID.
- **to**: Secondary annotation `convertNomenclature`

**Value**

A txt or a xlsx file in CHRONOS/extdata/Output/Scores/Linear/<org> or CHRONOS/extdata/Output/Scores/Non-Linear/<org>

**Examples**

```r
# Load scored subpathways from toy data
load(system.file('extdata', 'Examples//data.RData', package='CHRONOS'))

visualizeResults(linSubsScored, export='txt')
```
Index

CHRONOSrun, 2
convertMiRNANomenclature, 3
convertNomenclature, 2, 4, 13, 17
createPathwayGraphs, 5, 9, 10, 13

downloadKEGGPathwayList, 6, 8
downloadMiRecords, 3, 7
downloadPathways, 8

extractLinearSubpathways, 9, 14, 15
extractNonLinearSubpathways, 10, 14, 15

gEdgeTypes, 11

importExpressions, 9, 10, 12

pathwayMeasures, 13, 14

scoreSubpathways, 14, 16, 17
subpathwayKEGGmap, 15
subpathwayMiRNAs, 16

visualizeResults, 17