Package ‘CHRONOS’

May 21, 2024

Version 1.32.0
Date 2020-09-05
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 (>= 3.5)
SystemRequirements Java version >= 1.7, Pandoc
License GPL-2
NeedsCompilation no
LazyLoad yes
Imports XML, RCurl, RBGL, parallel, foreach, doParallel, openxlsx,
igraph, circlize, graph, stats, utils, grDevices, graphics,
methods, biomaRt, rJava
Suggests RUnit, BiocGenerics, knitr, rmarkdown
VignetteBuilder knitr
biocViews SystemsBiology, GraphAndNetwork, Pathways, KEGG
git_url https://git.bioconductor.org/packages/CHRONOS
git_branch RELEASE_3_19
git_last_commit f074b18
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-21
### Contents

- CHRONOSrun .............................................. 2
- convertMiRNANameclature ................................. 3
- convertNameclature ...................................... 4
- createPathwayGraphs .................................... 5
- downloadKEGGPathwayList .............................. 7
- downloadMirescords ..................................... 8
- downloadPathways ....................................... 9
- extractLinearSubpathways ............................. 10
- extractNonLinearSubpathways ......................... 11
- getEdgeTypes ........................................... 13
- importExpressions ...................................... 14
- pathwayMeasures ....................................... 15
- scoreSubpathways ...................................... 16
- subpathwayKEGGmap .................................... 17
- subpathwayMiRNAs ...................................... 18
- visualizeResults ...................................... 19

### Index

- CHRONOSrun .............................................. 20

### CHRONOSrun

**Default run of CHRONOS**

**Description**

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 `convertNameclature`)
- **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 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)
convertNomenclature

Arguments

org KEGG organism identifier.
miRNAs Vector of miRNAs identifiers.
update Update annotation mapper with latest annotation changes.

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)

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
**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)`

**Details**

<table>
<thead>
<tr>
<th>Gene Identifier</th>
<th>Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td>EntrezGene ID</td>
<td>'entrezgene'</td>
</tr>
<tr>
<td>Ensembl Gene ID</td>
<td>'ensembl_gene_id'</td>
</tr>
<tr>
<td>Ensembl Transcript ID</td>
<td>'ensembl_transcript_id'</td>
</tr>
<tr>
<td>Ensembl Protein ID</td>
<td>'ensembl_peptide_id'</td>
</tr>
<tr>
<td>HGNC ID</td>
<td>'hgnc_id'</td>
</tr>
<tr>
<td>HGNC Symbol</td>
<td>'hgnc_symbol'</td>
</tr>
<tr>
<td>HGNC Transcript name</td>
<td>'hgnc_transcript_name'</td>
</tr>
<tr>
<td>Refseq mRNA ID</td>
<td>'refseq_mrna'</td>
</tr>
<tr>
<td>Refseq Protein ID</td>
<td>'refseq_peptide'</td>
</tr>
<tr>
<td>UniProt/Swissprot Accession</td>
<td>'uniprot_swissprot_accession'</td>
</tr>
<tr>
<td>UniProt/Swissprot ID</td>
<td>'uniprot_swissprot'</td>
</tr>
<tr>
<td>UniGene ID</td>
<td>'unigene'</td>
</tr>
<tr>
<td>UniProt Genename ID</td>
<td>'uniprot_genename'</td>
</tr>
</tbody>
</table>

**Value**

Vector of converted gene identifiers

**Examples**

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

# Convert to HGNC ID, Ensembl Gene ID and UniProt Genename ID
from <- c('entrezgene')
to <- c('hgnc_symbol', 'ensembl_gene_id', 'uniprot_genename')
## Not run: res <- convertNomenclature(ids=ids, org='hsa', from=from, to=to)
```
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

```r
# 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

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

Examples

# 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**

```r
downloadPathways(org, pathways)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>org</code></td>
<td>KEGG organism identifier</td>
</tr>
<tr>
<td><code>pathways</code></td>
<td>Download pathways for specified organism:</td>
</tr>
<tr>
<td></td>
<td>'All' All organism pathways</td>
</tr>
<tr>
<td></td>
<td>'Metabolic' Metabolic pathways</td>
</tr>
<tr>
<td></td>
<td>'Non-Metabolic' Non metabolic pathways</td>
</tr>
<tr>
<td></td>
<td><code>&lt;vector of indexes&gt;</code> Using indexes from <code>downloadKEGGPathwayList</code></td>
</tr>
<tr>
<td></td>
<td><code>&lt;vector of names&gt;</code> Using pathway identifiers (i.e. c('00010', '00020'))</td>
</tr>
</tbody>
</table>

**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

```r
# View all available hsa pathways
## Not run: pathways <- downloadKEGGPathwayList(org='hsa')

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

## Not run: 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`.
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

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

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) Requires previous execution of importExpressions.

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 correspond to actual interactions between genes make up the non linear subpath.

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>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>unknown</td>
<td>3 02</td>
<td>activation 1</td>
</tr>
<tr>
<td>inhibition</td>
<td>2 04</td>
<td>binding/association 3</td>
</tr>
<tr>
<td>expression</td>
<td>1 06</td>
<td>repression 2</td>
</tr>
<tr>
<td>phosphorylation</td>
<td>3 08</td>
<td>dephosphorylation 3</td>
</tr>
<tr>
<td>ubiquitination</td>
<td>3 10</td>
<td>dissociation 3</td>
</tr>
<tr>
<td>indirect effect</td>
<td>3 12</td>
<td>state change 3</td>
</tr>
<tr>
<td>compound</td>
<td>3 14</td>
<td>hidden compound 3</td>
</tr>
<tr>
<td>missing interaction</td>
<td>3 16</td>
<td>activation_phosphorylation 1</td>
</tr>
<tr>
<td>activation_dephosphorylation</td>
<td>1 18</td>
<td>activation_ubiquitination 1</td>
</tr>
<tr>
<td>activation_indirect effect</td>
<td>1 20</td>
<td>activation_binding/association 1</td>
</tr>
<tr>
<td>activation_inhibition</td>
<td>3 22</td>
<td>activation_methylation 1</td>
</tr>
<tr>
<td>inhibition_phosphorylation</td>
<td>2 24</td>
<td>inhibition_dephosphorylation 2</td>
</tr>
<tr>
<td>inhibition_ubiquitination</td>
<td>2 26</td>
<td>inhibition_indirect effect 2</td>
</tr>
<tr>
<td>inhibition_binding/association</td>
<td>2 28</td>
<td>inhibition_expression 2</td>
</tr>
<tr>
<td>inhibition_methylation</td>
<td>2 30</td>
<td>compound_expression 1</td>
</tr>
<tr>
<td>compound_activation</td>
<td>1 32</td>
<td>compound_inhibition 2</td>
</tr>
<tr>
<td>compound_activation_indirect effect</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>compound_activation_phosphorylation</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>phosphorylation_indirect effect</td>
<td>3</td>
<td></td>
</tr>
</tbody>
</table>
If an interaction type has been supplied, the corresponding edge types are returned. If not, the complete mapping is returned.

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
Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>data</td>
<td>Expressions data filename or matrix.</td>
</tr>
<tr>
<td>type</td>
<td>Expressions data type. (or mRNA expressions, type=&lt;nomenType&gt;. Available gene expression nomenclature can be found in <code>convertNomenclature</code> function. For miRNA expressions, type='miRNA'.)</td>
</tr>
<tr>
<td>sep</td>
<td>File delimiter.</td>
</tr>
<tr>
<td>org</td>
<td>KEGG organism identifier</td>
</tr>
<tr>
<td>mRNANameclature</td>
<td>Nomenclature of user's mRNA expressions</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>graphs</td>
<td>Pathway graphs as returned from <code>createPathwayGraphs</code>.</td>
</tr>
</tbody>
</table>
scoreSubpathways

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

Value
Matrix with pathness, betweeness centrality and degree values for each gene in the pathway graphs at it’s 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.

<table>
<thead>
<tr>
<th>parameter</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvalue</td>
<td>Statistical evaluation</td>
</tr>
<tr>
<td>measures</td>
<td>Structural and functional evaluation</td>
</tr>
<tr>
<td>subScore</td>
<td>mRNA-mRNA interaction scoring</td>
</tr>
<tr>
<td>mirScore</td>
<td>miRNA-mRNA interaction scoring</td>
</tr>
</tbody>
</table>

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.
subpathwayKEGGmap

Details

...

Value

<table>
<thead>
<tr>
<th>subpathways</th>
<th>High ranking subpathways</th>
</tr>
</thead>
<tbody>
<tr>
<td>subScores</td>
<td>miRNA-subpathway scores</td>
</tr>
<tr>
<td>mRNAScores</td>
<td>mRNA-mRNA scores for each subpathway and for each time point</td>
</tr>
<tr>
<td>miRNAsOverSubpathway</td>
<td>High ranking miRNAs hitting each subpathway</td>
</tr>
<tr>
<td>pValues</td>
<td>P-value of each subpathway</td>
</tr>
<tr>
<td>filters</td>
<td>Filters used for the evaluation</td>
</tr>
</tbody>
</table>

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 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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subpathways</td>
<td>Subpathways as returned by <code>extractLinearSubpathways</code> or <code>extractNonLinearSubpathways</code></td>
</tr>
<tr>
<td>type</td>
<td>Subpathway type (Linear, Non-Linear)</td>
</tr>
<tr>
<td>openInBrowser</td>
<td>Open link in default browser.</td>
</tr>
</tbody>
</table>

**Value**

Vector of links of KEGG pathway maps.

**Examples**

```r
# 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)
```

---

**Description**

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

**Usage**

```r
subpathwayMiRNAs(summary, subIdx, timePoints)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>summary</td>
<td>Output from <code>scoreSubpathways</code></td>
</tr>
<tr>
<td>subIdx</td>
<td>Subpathway index</td>
</tr>
<tr>
<td>timePoints</td>
<td>Time points to include in visualization, default to all.</td>
</tr>
</tbody>
</table>

**Value**

.

**Examples**

```r
# 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

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

visualizeResults(linSubsScored, export='txt')
Index

CHRONOSrun, 2
convertMiRNAOMomenclature, 3
convertNomenclature, 2, 4, 15, 19
createPathwayGraphs, 5, 10, 12, 15

downloadKEGGPathwayList, 7, 9
downloadMiRecords, 3, 8
downloadPathways, 9

extractLinearSubpathways, 10, 16, 18
extractNonLinearSubpathways, 11, 16, 18

gEdgeTypes, 13

importExpressions, 10–12, 14

pathwayMeasures, 15, 16

scoreSubpathways, 16, 18, 19
subpathwayKEGGmap, 17
subpathwayMiRNAs, 18

visualizeResults, 19