Package 'CHRONOS'

May 15, 2025

Version 1.37.0

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 devel

git_last_commit beddcd9

git_last_commit_date 2025-04-15

Repository Bioconductor 3.22

Date/Publication 2025-05-15

Contents

`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 `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 availiable 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 availiable, 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

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 retreive 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

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

Li, C., Han, J., Yao, Q., Zou, C., Xu, Y., Zhang, C., ... & Li, X. (2013). Subpathway-GM: identification of metabolic subpathways via joint power of interesting genes and metabolites and their topologies within pathways. Nucleic acids research, 41(9), e101-e101.

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 availiable pathways for an organism.*

Description

Retrieve all availiable pathways for an organism.

Usage

```
downloadKEGGPathwayList(org)
```

Arguments

`org`	KEGG organism identifier.

Details

.

Value

Data frame of pathway ids and names.

References

- http://www.genome.jp/kegg/pathway.html

Examples

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

# Retrieve all availiable 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/TargertScanS.

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

```
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

- http://www.genome.jp/kegg/pathway.html

Examples

```
# View all availiable 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

```
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. Availiable 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

```
# 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)
```

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. Availiable 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 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

01	unknown	3	02	activation	1
03	inhibition	2	04	binding/association	3
05	expression	1	06	repression	2
07	phosphorylation	3	08	dephosphorylation	3
09	ubiquitination	3	10	dissociation	3
11	indirect effect	3	12	state change	3
13	compound	3	14	hidden compound	3
16	missing interaction	3	16	activation_phosphorylation	1
17	activation_dephosphorylation	1	18	activation_ubiquitination	1
19	activation_indirect effect	1	20	activation_binding/association	1
21	activation_inhibition	3	22	activation_methylation	1
23	inhibition_phosphorylation	2	24	inhibition_dephosphorylation	2
25	inhibition_ubiquitination	2	26	inhibition_indirect effect	2
27	inhibition_binding/association	2	28	inhibition_expression	2
29	inhibition_methylation	2	30	compound_expression	1
31	compound_activation	1	32	compound_inhibition	2

33	compound_activation_indirect effect	1
34	compound_activation_phosphorylation	1
35	phosphorylation_indirect effect	3
36	phosphorylation_binding/association	3
37	phosphorylation_dissociation	3
38	dephosphorylation_indirect effect	3
39	binding/association_missing interaction	3
40	binding/association_indirect effect	3
41	expression_indirect effect	1
42	repression_indirect effect	2
43	ubiquitination_inhibition	2
44	dissociation_missing interaction	3
45	indirect effect_phosphorylation	3
46	activation_phosphorylation_binding/association	1
47	activation_phosphorylation_indirect effect	1

Value

If an interaction type has been supplied, the corresponging 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 innteraction - 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, mRNAnomenclature)
```

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

```
# Example

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

importExpressions(data=mRNAexpr, type='mRNA',
                mRNAnomenclature='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.

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 scorng scheme (IS) for each time point.*

Description

Evaluate subpathways using an interacting scorng 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.

pvalue	Statistical evaluation
measures	Structural and functional evaluation
subScore	mRNA-mRNA interaction scoring
mirScore	miRNA-mRNA interaction scoring

`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

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

Jethava, V., Bhattacharyya, C., Dubhashi, D., & Vemuri, G. N. (2011). Netgem: Network embedded temporal generative model for gene expression data. BMC bioinformatics, 12(1), 327.

Kim,Y. et al. (2011). Principal network analysis: identification of subnetworks representing major dynamics using gene expression data. Bioinformatics, 27(3), 391-398

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

`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

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

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

Index