Package 'paxtoolsr'

April 10, 2015

Type Package

Title PaxtoolsR: Access Pathways from Multiple Databases through BioPAX and Pathway Commons

Version 1.0.13

Date 2015-04-08

Author Augustin Luna

Maintainer Augustin Luna <lunaa@cbio.mskcc.org>

Depends R (>= 3.1.1), rJava (>= 0.9-4), XML, RCurl, rjson, plyr

Suggests testthat, knitr, BiocStyle, rmarkdown, RColorBrewer, igraph, biomaRt, estrogen, affy, hgu95av2, hgu95av2cdf, limma

SystemRequirements Java (>= 1.5)

License LGPL-3

Description The package provides a set of R functions for interacting with BioPAX OWL files using Paxtools and the querying Pathway Commons (PC) molecular interaction database that are hosted by the Computational Biology Center at Memorial Sloan-Kettering Cancer
Center (MSKCC). Pathway Commons databases include: BIND, BioGRID, CORUM, CTD, DIP, DrugBank, HPRD, HumanCyc,
IntAct, KEGG, MirTarBase, Panther, PhosphoSitePlus, Reactome, RECON, TRANSFAC.

VignetteBuilder knitr

LazyData true

biocViews GeneSetEnrichment, GraphAndNetwork, Pathways, Software, SystemsBiology, NetworkEnrichment, Network

URL https://bitbucket.org/cbio_mskcc/paxtoolsr

R topics documented:

downloadPc 2
fetch 3
fromPsimi 4
getNeighbors 5

getPc . . . 5
getPcUrl . . . 6
graphPc . . . 7
idMapping . . . 8
integrateBiopax . . . 9
mergeBiopax . . . 10
pcDirections . . . 10
pcFormats . . . 11
pcGraphQueries . . . 12
readGmt . . . 12
searchPc . . . 13
splitSifnx . . . 14
summarize . . . 15
toGSEA . . . 15
toLevel3 . . . 16
topPathways . . . 17
toSBGN . . . 18
toSif . . . 18
toSifnx . . . 19
traverse . . . 20
validate . . . 21

Index **23**

`downloadPc` *Download Pathway Commons data*

Description

Download Pathway Commons data in various formats

Usage

```
downloadPc(format = c("SIFNX", "GMT"), verbose = FALSE)
```

Arguments

`format`	a string describing the format to be downloaded; currently, only the Extended Simple Interaction Format (SIF) "SIFNX" and Gene Set Enrichment Analysis "GMT" formats for the entire Pathway Commons database are supported.
`verbose`	a boolean debugging information

Details

Description of SIF interactions: http://www.pathwaycommons.org/pc/sif_interaction_rules.do Description of BioPAX classes: http://www.biopax.org/owldoc/Level3/

Value

a named list with named pathways, each entry contains a vector of gene symbols (for GMT) or a list with two data.frames (for SIFNX):

- edges Network edges with the following columns: PARTICIPANT_A: Edge (interaction) participant, INTERACTION_TYPE: Interaction type (see details), PARTICIPANT_B: Edge (interaction) participant, INTERACTION_DATA_SOURCE: Semi-colon delimited list of database sources of the interaction, INTERACTION_PUBMED_ID: Semi-colon delimited list of NCBI Pubmed IDs that give evidence for the interaction
- nodes Node information: PARTICIPANT: Interaction participant, PARTICIPANT_TYPE: BioPAX class (see details), PARTICIPANT_NAME: Display name for the participant, UNIFICATION_XREF: A UnificationXref defines a reference to an entity in an external resource that has the same biological identity as the referring entity, RELATIONSHIP_XREF: An RelationshipXref defines a reference to an entity in an external resource that does not have the same biological identity as the referring entity.

See Also

downloadPc

Examples

```
format <- "GMT"
#downloadPc(format=format)
```

---

fetch *Fetch a set of IDs from a BioPAX OWL file*

Description

This function will create a subsetted object with specified URIs.

Usage

```
fetch(inputFile, outputFile = NULL, idList)
```

Arguments

inputFile	a string of the name of the input BioPAX OWL file
outputFile	a string with the name of the output BioPAX OWL file
idList	a vector of IDs from the BioPAX OWL file

Details

Only entities in the input BioPAX file will be used in the fetch. IDs used must be URIs for the entities of interest. Additional properties such as cross-references for fetched entities will be included in the output.

Value

an XMLInternalDocument representing a BioPAX OWL file

Examples

```
outFile <- tempfile()
tmp <- getPc(uri="http://identifiers.org/reactome/REACT_12034.3",
             format="BIOPAX",
             verbose=TRUE)
ids <- c("http://identifiers.org/uniprot/P36894",
         "http://identifiers.org/uniprot/Q13873")
results <- fetch(tmp, outFile, ids)
```

`fromPsimi` *Read PSIMI file*

Description

This function reads in a PSIMI file.

Usage

```
fromPsimi(inputFile, outputFile = NULL, bpLevelArg = 3)
```

Arguments

`inputFile`	a string of the name of the input PSIMI file
`outputFile`	a string of the name of the output BioPAX OWL file
`bpLevelArg`	a string representing the BioPAX level for the output file (default: NULL)

Details

The Proteomics Standard Initiative (PSIMI) format is described at https://code.google.com/p/psimi/wiki/PsimiTabFormat

Value

an XMLInternalDocument representing a BioPAX OWL file

Examples

```
outFile <- tempfile()
results <- fromPsimi(system.file("extdata", "10523676-compact.xml", package="paxtoolsr"),
                                 outFile,
                                 "3")
```

getNeighbors *Get the neighbors of a set of IDs in a BioPAX file*

Description

This function retrieves a set of neighbors for a set of IDs in a BioPAX file.

Usage

```
getNeighbors(inputFile, outputFile = NULL, idList)
```

Arguments

inputFile	a string with the name of the input BioPAX OWL file
outputFile	a string with the name of the output BioPAX OWL file
idList	a vector of IDs from the BioPAX OWL file

Details

Only entities in the input BioPAX file will be searched for neighbors. IDs used must be URIs for the entities of interest.

Value

an XMLInternalDocument representing a BioPAX OWL file

Examples

```
outFile <- tempfile()
results <- getNeighbors(system.file("extdata",
  "raf_map_kinase_cascade_reactome.owl", package="paxtoolsr"),
  outFile,
  c("HTTP://WWW.REACTOME.ORG/BIOPAX/48887#PROTEIN2360_1_9606",
    "HTTP://WWW.REACTOME.ORG/BIOPAX/48887#PROTEIN1631_1_9606"))
```

getPc *Get Pathway Commons BioPAX elements*

Description

This command retrieves full pathway information for a set of elements such as pathway, interaction or physical entity given the RDF IDs.

Usage

```
getPc(uri, format = NULL, verbose = FALSE)
```

Arguments

uri	a vector that includes valid/existing BioPAX element's URI (RDF ID; for utility classes that were "normalized", such as entity refereneces and controlled vocabularies, it is usually a Idntifiers.org URL. Multiple IDs are allowed per query, for example, c("http://identifiers.org/uniprot/Q06609", "http://identifiers.org/uniprot/Q549Z0") See also about MIRIAM and Identifiers.org in details.
format	output format. Valid options can be found using pcFormats
verbose	a boolean, display the command used to query Pathway Commons

Details

Get commands only retrieve the BioPAX elements that are directly mapped to the ID. Use the "traverse query to traverse BioPAX graph and obtain child/owner elements.

Information on MIRIAM and Identifiers.org http://www.pathwaycommons.org/pc2/#miriam

Value

a XMLInternalDocument object

See Also

pcFormats

Examples

```
uri <- "http://identifiers.org/uniprot/O14503"
#results <- getPc(uri)

uri <- c("http://identifiers.org/uniprot/O14503", "http://identifiers.org/uniprot/Q9P2X7")
#results <- getPc(uri, verbose=TRUE)
```

---

getPcUrl *Get base Pathway Commons URL*

Description

Get base Pathway Commons URL

Usage

```
getPcUrl()
```

Details

paxtoolsr will support versions Pathway Commons 5 and later. Old versions of the webservice will not be not be operational. Users can parse older BioPAX outputs as an alternative.

Value

a string with base Pathway Commons URL

Examples

```
url <- getPcUrl()
```

`graphPc` *Get Pathway Commons BioPAX elements*

Description

This function will retrieve a set of BioPAX elements given a graph query match.

Usage

```
graphPc(kind, source, target = NULL, direction = NULL, limit = NULL,
  format = NULL, datasource = NULL, organism = NULL, verbose = FALSE)
```

Arguments

`kind`	graph query. Valid options can be found using `pcGraphQueries` See Details for information on graph queries.
`source`	source object's URI/ID. Multiple source URIs/IDs are allowed per query, for example c("http://identifiers.org/uniprot/Q06609", "http://identifiers.org/uniprot/Q549Z0") See a note about MIRIAM and Identifiers.org in details
`target`	[Required for PATHSFROMTO graph query] target URI/ID. Multiple target URIs are allowed per query; for example c("http://identifiers.org/uniprot/Q06609", "http://identifiers.org/uniprot/Q549Z0") See a note about MIRIAM and Identifiers.org in details
`direction`	[Optional, for NEIGHBORHOOD and COMMONSTREAM algorithms] - graph search direction. Valid options: `pcDirections`.
`limit`	graph query search distance limit (default: 1).
`format`	output format. Valid options: `pcFormats`
`datasource`	datasource filter (same as for 'search').
`organism`	organism filter (same as for 'search').
`verbose`	a boolean, display the command used to query Pathway Commons

Value

depending on the the output format a different object may be returned. `pcFormats`

See Also

`pcFormats`, `pcDirections`

Examples

```
source <- "http://identifiers.org/uniprot/O14503"
#results <- graphPc(source=, kind="neighborhood", format="EXTENDED_BINARY_SIF")
```

`idMapping` *Map IDs to Primary Uniprot or ChEBI IDs*

Description

Unambiguously maps, e.g., HGNC gene symbols, NCBI Gene, RefSeq, ENS*, and secondary UniProt identifiers to the primary UniProt accessions, or - ChEBI and PubChem IDs to primary ChEBI. You can mix different standard ID types in one query.

Usage

```
idMapping(ids, verbose = FALSE)
```

Arguments

`ids`	a vector of IDs
`verbose`	a boolean, display the command used to query Pathway Commons

Details

This is a specific id-mapping (not general-purpose) for reference proteins and small molecules; it was first designed for internal use, such as to improve BioPAX data integration and allow for graph queries accept not only URIs but also standard IDs. The mapping tables were derived exclusively from Swiss-Prot (DR fields) and ChEBI data (manually created tables and other mapping types and sources can be added in the future versions if necessary).

Value

a list of where each entry is a HGNC symbol provided and the each value is a primary UniProt or ChEBI ID.

Examples

```
genes <- c("BRCA2", "TP53")
#results <- idMapping(genes)
```

integrateBiopax *Integrate two BioPAX OWL files (DEPRECATED)*

Description

This function merges two BioPAX OWL files

Usage

```
integrateBiopax(inputFile1, inputFile2, outputFile = NULL)
```

Arguments

inputFile1	a string of the name of the input BioPAX OWL file
inputFile2	a string of the name of the input BioPAX OWL file
outputFile	a string of the name of the output integrated BioPAX OWL file

Details

This method is deprecated. Use mergeBiopax instead.

Value

an XMLInternalDocument representing a BioPAX OWL file

See Also

mergeBiopax

Examples

```
outFile <- tempfile()
results <- integrateBiopax(system.file("extdata", "raf_map_kinase_cascade_reactome.owl",
  package="paxtoolsr"),
  system.file("extdata", "dna_replication.owl", package="paxtoolsr"),
  outFile)
```

mergeBiopax *Merges two BioPAX OWL files*

Description

This function merges two BioPAX OWL files

Usage

```
mergeBiopax(inputFile1, inputFile2, outputFile = NULL)
```

Arguments

inputFile1	a string of the name of the input BioPAX OWL file
inputFile2	a string of the name of the input BioPAX OWL file
outputFile	a string of the name of the output merged BioPAX OWL file (Optional)

Details

Only entities that share IDs will be merged. No additional merging occurs on cross-references. Merging may result in warning messages caused as a result of redundant actions being checked against by the Java library; these messages may be ignored.

Value

an XMLInternalDocument representing a BioPAX OWL file

Examples

```
outFile <- tempfile()
results <- mergeBiopax(system.file("extdata", "raf_map_kinase_cascade_reactome.owl",
                       package="paxtoolsr"),
                       system.file("extdata", "dna_replication.owl",
                       package="paxtoolsr"),
                       outFile)
```

pcDirections *Acceptable Pathway Commons Directions*

Description

A simple function to see valid options

Usage

```
pcDirections()
```

Details

- BOTHSTREAM where the current entity can either be the source or target of an interaction
- DOWNSTREAM where the current entity can only be the source
- UPSTREAM where the current entity can only be the target

Value

acceptable Pathway Commons directions

Examples

```
pcDirections()
```

---

pcFormats *Acceptable Pathway Commons Formats*

Description

A simple function to see valid options

Usage

```
pcFormats()
```

Details

See references.

Value

acceptable Pathway Commons formats

References

Output Formats Description: http://www.pathwaycommons.org/pc2/help/formats.html

Examples

```
pcFormats()
```

pcGraphQueries *Acceptable Pathway Commons Graph Queries*

Description

A simple function to see valid options

Usage

```
pcGraphQueries()
```

Details

- COMMONSTREAM searches common downstream or common upstream of a specified set of entities based on the given directions within the boundaries of a specified length limit
- NEIGHBORHOOD searches the neighborhood of given source set of nodes
- PATHSBETWEEN finds the paths between specific source set of states or entities within the boundaries of a specified length limit
- PATHSFROMTO finds the paths from a specific source set of states or entities to a specific target set of states or entities within the boundaries of a specified length limit

Value

acceptable Pathway Commons graph queries

Examples

```
pcGraphQueries()
```

readGmt *Read in gene sets from GMT files*

Description

This function will read in gene sets in the GMT format into a named list.

Usage

```
readGmt(inputFile)
```

Arguments

inputFile	an inputFile in the GMT format

Value

a named list where each entry corresponds to a gene set

Examples

```
results <- readGmt(system.file("extdata", "test_gsea.gmt", package="paxtoolsr"))
```

`searchPc` *Search Pathway Commons*

Description

This command provides a text search using the Lucene query syntax.

Usage

```
searchPc(q, page = 0, datasource = NULL, organism = NULL, type = NULL,
  verbose = FALSE)
```

Arguments

`q`	a keyword, name, external identifier, or a Lucene query string.
`page`	an integer giving the search result page number (N>=0, default: 0)
`datasource`	a vector that is a filter by data source (use names or URIs of pathway data sources or of any existing Provenance object). If multiple data source values are specified, a union of hits from specified sources is returned. For example, datasource as c("reactome", "pid") returns hits associated with Reactome or PID.
`organism`	a vector that is an organism filter. The organism can be specified either by official name, e.g. "homo sapiens" or by NCBI taxonomy id, e.g. "9606". Similar to data sources, if multiple organisms are declared a union of all hits from specified organisms is returned. For example organism as c("9606", "10016") returns results for both human and mice. Only humans, "9606" is officially supported.
`type`	BioPAX class filter. See Details.
`verbose`	a boolean, display the command used to query Pathway Commons

Details

Indexed fields were selected based on most common searches. Some of these fields are direct BioPAX properties, others are composite relationships. All index fields are (case-sensitive):comment, ecnumber, keyword, name, pathway, term, xrefdb, xrefid, dataSource, and organism. The pathway field maps to all participants of pathways that contain the keyword(s) in any of its text fields. This field is transitive in the sense that participants of all sub-pathways are also returned. Finally, keyword is a transitive aggregate field that includes all searchable keywords of that element and its child elements - e.g. a complex would be returned by a keyword search if one of its members has a match. Keyword is the default field type. All searches can also be filtered by data source and organism. It is also possible to restrict the domain class using the 'type' parameter. This query can be

used standalone or to retrieve starting points for graph searches. Search strings are case insensitive unless put inside quotes.

BioPAX classes can be found at http://www.pathwaycommons.org/pc2/#biopax_types

Value

an XMLInternalDocument with results

Examples

```
query <- "Q06609"
#results <- searchPc(query)

query <- "glycolysis"
#results <- searchPc(query, type="Pathway")
```

`splitSifnx` *Split Extended SIF File*

Description

Split an extended SIF file into nodes and edges

Usage

```
splitSifnx(con, verbose = FALSE)
```

Arguments

`con`	an opened file connection
`verbose`	a boolean, display debugging information

Details

SIFNX files from Pathway Commons commonly come a single file that includes a tab-delimited sections for nodes and another for edges. The sections are separated by an empty lines. These sections must be split before they are read.

Value

a list with nodes and edges entries

Examples

```
con <- file(system.file("extdata", "test_sifnx.txt", package="paxtoolsr"))
results <- splitSifnx(con, verbose=TRUE)
```

summarize *Summarize a BioPAX file*

Description

This function provides a summary of BioPAX classes.

Usage

```
summarize(inputFile)
```

Arguments

inputFile	a string of the name of the input BioPAX OWL file

Details

BioPAX classes are defined by the BioPAX specification: http://www.biopax.org/

Value

list with BioPAX class counts

Examples

```
summary <- summarize(system.file("extdata", "raf_map_kinase_cascade_reactome.owl",
package="paxtoolsr"))
```

toGSEA *Converts a BioPAX OWL file to a GSEA GMT gene set*

Description

This function converts pathway information stored as BioPAX files into the the GSEA .gmt format.

Usage

```
toGSEA(inputFile, outputFile = NULL, database, crossSpeciesCheckFlag)
```

Arguments

inputFile	a string of the name of the input OWL file
outputFile	a string of the name of the output file
database	a string of the name of the identifier type to be included (e.g. "HGNC Symbol")
crossSpeciesCheckFlag	a boolean that ensures participant protein is from same species

Details

The GSEA GMT format is a tab-delimited format where each row represents a gene set. The first column is the gene set name. The second column is a brief description. Other columns for each row contain genes in the gene set; these rows may be of unequal lengths.

Value

a vector with the GSEA content

Examples

```
outFile <- tempfile()
results <- toGSEA(system.file("extdata", "biopax3-short-metabolic-pathway.owl",
                              package="paxtoolsr"),
                              outFile,
                              "uniprot",
                              crossSpeciesCheckFlag=TRUE)
```

---

toLevel3 *Convert a BioPAX OWL file to BioPAX Level 3*

Description

This file will convert older BioPAX objects to BioPAX Level 3

Usage

```
toLevel3(inputFile, outputFile = NULL)
```

Arguments

inputFile	a string of the name of the input BioPAX OWL file
outputFile	a string of the name of the output BioPAX OWL file

Value

an XMLInternalDocument representing a BioPAX OWL file

Examples

```
outFile <- tempfile()
results <- toLevel3(system.file("extdata", "raf_map_kinase_cascade_reactome.owl",
  package="paxtoolsr"),
  outFile)
```

topPathways *Retrieve top pathways*

Description

This command returns all "top" pathways.

Usage

```
topPathways(datasource = NULL, organism = NULL, verbose = FALSE)
```

Arguments

datasource	filter by data source (same as for 'search').
organism	organism filter (same as for 'search').
verbose	a boolean, display the command used to query Pathway Commons

Details

Pathways that are neither 'controlled' nor 'pathwayComponent' of another process.

Value

a data.frame with the following columns:

- uri URI ID for the pathway
- biopaxClass the type of BioPAX object
- name a human readable name
- dataSource the dataSource for the pathway
- organism an organism identifier
- pathway URI ID for the pathway

Examples

```
datasource <- "panther"
#results <- topPathways(datasource=datasource)
```

toSBGN *Convert a BioPAX OWL file to SBGNML*

Description

This function will convert a BioPAX OWL file into the Systems Biology Graphical Notation (SBGN) Markup Language (SBGNML) XML representation

Usage

```
toSBGN(inputFile, outputFile = NULL)
```

Arguments

inputFile	a string of the name of the input BioPAX OWL file
outputFile	a string of the name of the output SBGNML file

Details

Objects in the SBGNML format are laid out using a Compound Spring Embedder (CoSE) layout

Value

an XMLInternalDocument representing a SBGNML file

References

http://www.cs.bilkent.edu.tr/~ivis/layout/cose-animated-demo/cose.html

Examples

```
outFile <- tempfile()
results <- toSBGN(system.file("extdata", "biopax3-short-metabolic-pathway.owl",
  package="paxtoolsr"),
  outFile)
```

toSif *Convert a BioPAX OWL file to SIF*

Description

Convert a BioPAX OWL file to a binary SIF file

Usage

```
toSif(inputFile, outputFile = NULL)
```

Arguments

inputFile	a string of the name of the input BioPAX OWL file
outputFile	a string of the name of the output SIF file (Optional)

Details

Information on SIF conversion is provided on the Pathway Commons site: http://www.pathwaycommons.org/pc2/

Value

a 3-column data.frame where the columns are named: PARTICIPANT_A, INTERACTION_TYPE, PARTICIPANT_B

Examples

```
outFile <- tempfile()
results <- toSif(system.file("extdata", "raf_map_kinase_cascade_reactome.owl",
  package="paxtoolsr"),
  outFile)
```

---

toSifnx *Converts BioPAX OWL file to extended binary SIF representation*

Description

Converts BioPAX OWL file to extended binary SIF representation

Usage

```
toSifnx(inputFile, outputNodesFile = NULL, outputEdgesFile = NULL,
  nodeProps, edgeProps)
```

Arguments

inputFile	a string with the name of the input BioPAX OWL file
outputNodesFile	a string with the name of the output file for node information
outputEdgesFile	a string with the name of the output file for edge information
nodeProps	a string node properties to be saved; these are set up as XPath like expressions of data in BioPAX files (e.g. c("EntityReference/name", "EntityReference/xref"))
edgeProps	a string edge properties to be saved; these are set up as XPath like expressions of data in BioPAX files (e.g. "Interaction/dataSource/displayName")

Details

Information on SIF conversion is provided on the Pathway Commons site: http://www.pathwaycommons.org/pc2/

Value

a list with two entries

- nodes a data.frame with interaction participant information specified in nodeProps
- edges a data.frame with SIF formatted data (i.e. an edgelist with a additional middle column denoting interaction type) and any additional columns specified in edgeProps.

Examples

```
edgesFile <- tempfile()
nodesFile <- tempfile()
results <- toSifnx(system.file("extdata", "raf_map_kinase_cascade_reactome.owl",
  package="paxtoolsr"),
  nodesFile,
  edgesFile,
  c("EntityReference/name", "EntityReference/xref"),
  "Interaction/dataSource/displayName")
```

---

traverse *Access Pathway Commons using XPath-type expressions*

Description

This command provides XPath-like access to the Pathway Commons.

Usage

```
traverse(uri, path, verbose = FALSE)
```

Arguments

uri	a BioPAX element URI - specified similarly to the 'GET' command above). Multiple IDs are allowed (uri=...&uri=...&uri=...).
path	a BioPAX propery path in the form of property1[:type1]/property2[:type2]; see properties, inverse properties, Paxtools, org.biopax.paxtools.controller.PathAccessor.
verbose	a boolean, display the command used to query Pathway Commons

Details

With traverse users can explicitly state the paths they would like to access. The format of the path query is in the form: [Initial Class]/[property1]:[classRestriction(optional)]/[property2]... A "*" sign after the property instructs path accessor to transitively traverse that property. For example, the following path accessor will traverse through all physical entity components within a complex: "Complex/component*/entityReference/xref:UnificationXref" The following will list display names of all participants of interactions, which are components (pathwayComponent) of a pathway (note: pathwayOrder property, where same or other interactions can be reached, is not considered here): "Pathway/pathwayComponent:Interaction/participant*/displayName" The optional parameter classRestriction allows to restrict/filter the returned property values to a certain subclass of the range of that property. In the first example above, this is used to get only the Unification Xrefs. Path accessors can use all the official BioPAX properties as well as additional derived classes and parameters in paxtools such as inverse parameters and interfaces that represent anonymous union classes in OWL. (See Paxtools documentation for more details).

Value

an XMLInternalDocument with results

References

Paxtools Documentation: http://www.biopax.org/m2site/

Examples

```
uri <- "http://identifiers.org/uniprot/P38398"
#results <- traverse(uri=uri, path="ProteinReference/organism/displayName")
```

---

validate *Validate BioPAX files*

Description

This function validates BioPAX files for errors.

Usage

```
validate(inputFile, outputFile = NULL, type = c("xml", "html", "biopax"),
  autoFix = FALSE, onlyErrors = FALSE, maxErrors = NULL,
  notStrict = FALSE)
```

Arguments

inputFile	a string of the name of the input BioPAX OWL file
outputFile	a string of the name of the output file containing validation results
type	a string denoting the type of output: xml (default), html, biopax

autoFix	a boolean that determines if the input file should be fixed automatically. Errors that can be automatically fixed include generating displayName properties from names, inferring organism, and inferring dataSource
onlyErrors	a boolean of whether to only display errors
maxErrors	a integer denoting the number of errors to return
notStrict	a boolean of whether to be strict in validation (default: FALSE)

Details

See the publication by Rodchenkov, et al. for information on the BioPAX validator. See http://biopax.baderlab.org/validator for additional information on validator. See http://biopax.baderlab.org/validator/errorTypes.html for information on error types.

Value

an XMLInternalDocument is returned if type is set to "xml" otherwise the location of the outputfile is returned.

References

Rodchenkov I, Demir E, Sander C, Bader GD. The BioPAX Validator, http://www.ncbi.nlm.nih.gov/pubmed/23918249

Examples

```
outFile <- tempfile()
rawDoc <- validate(system.file("extdata", "raf_map_kinase_cascade_reactome.owl",
  package="paxtoolsr"), onlyErrors=TRUE)
```

Index

downloadPc, 2, *3*

fetch, 3
fromPsimi, 4

getNeighbors, 5
getPc, 5
getPcUrl, 6
graphPc, 7

idMapping, 8
integrateBiopax, 9

mergeBiopax, *9*, 10

pcDirections, *7*, 10
pcFormats, *6, 7*, 11
pcGraphQueries, *7*, 12

readGmt, 12

searchPc, 13
splitSifnx, 14
summarize, 15

toGSEA, 15
toLevel3, 16
topPathways, 17
toSBGN, 18
toSif, 18
toSifnx, 19
traverse, 20

validate, 21