Package ‘pogos’

May 30, 2024

Title PharmacOGenomics Ontology Support
Description Provide simple utilities for querying bhklab PharmacoDB, modeling API outputs, and integrating to cell and compound ontologies.
Version 1.24.0
Author Vince Carey <stvjc@channing.harvard.edu>
Suggests knitr, DT, ontologyPlot, testthat, rmarkdown, BiocStyle
Imports methods, S4Vectors, utils, shiny, ontoProc, ggplot2, graphics
Depends R (>= 3.5.0), rjson (>= 0.2.15), http (>= 1.3.1)
Maintainer VJ Carey <stvjc@channing.harvard.edu>
License Artistic-2.0
LazyLoad yes
LazyData yes
biocViews Pharmacogenomics, PooledScreens, ImmunoOncology
RoxygenNote 7.2.3
VignetteBuilder knitr
git_url https://git.bioconductor.org/packages/pogos
git_branch RELEASE_3_19
git_last_commit da20fbf
git_last_commit_date 2024-04-30
Repository Bioconductor 3.19
Date/Publication 2024-05-29

Contents

<table>
<thead>
<tr>
<th>Contents</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>basicDecoder</td>
<td>2</td>
</tr>
<tr>
<td>compoundsByCell</td>
<td>2</td>
</tr>
<tr>
<td>compounds_v1</td>
<td>3</td>
</tr>
<tr>
<td>DRProfile-class</td>
<td>4</td>
</tr>
<tr>
<td>DRTraceSet-class</td>
<td>5</td>
</tr>
</tbody>
</table>
**compoundsByCell**

initial version of compound browser over pharmacoDb cells

**Description**

initial version of compound browser over pharmacoDb cells

**Usage**

compoundsByCell()

**Value**

only used for side effect of running shiny app

---

**basicDecoder**

convert binary output of GET($content) to list

**Description**

convert binary output of GET($content) to list

**Usage**

basicDecoder(x)

**Arguments**

x string suitable for input to GET as GET(x)

**Value**

output of fromJSON, typically a list

**Examples**

```
cl = basicDecoder('https://pharmacodb.pmgenomics.ca/api/v1/cell_lines')
unlist(cl)
```
**Note**

Simple shiny app demonstrating coverage of PharmacoDb compounds by CHEBI. If a cell line selected is not present in selected dataset, the app will wait for a compatible selection to be made.

**Examples**

```r
if (!requireNamespace("shiny")) stop("install shiny to use compoundsByCell")
if (interactive()) print(compoundsByCell())
```

---

**compounds_v1**

*compounds_v1: serialization of compounds info from PharmacoDb v1*

**Description**

compounds_v1: serialization of compounds info from PharmacoDb v1

**Usage**

- `compounds_v1`
- `tissues_v1`
- `cell_lines_v1`
- `datasets_v1`
- `CCLE_drts`

**Format**

- S4Vectors DataFrame instance
- S4Vectors DataFrame instance
- S4Vectors DataFrame instance
- S4Vectors DataFrame instance
- DRTraceSet instance

**Source**

- PharmacoDb Sept 2017
- PharmacoDb Sept 2017
- PharmacoDb Sept 2017
- PharmacoDb Sept 2017
- PharmacoDb April 2018
Examples

data(compounds_v1)
head(compounds_v1)
data(tissues_v1)
head(tissues_v1)
data(cell_lines_v1)
head(cell_lines_v1)
data(datasets_v1)
head(datasets_v1)
data(CCLE_drts)
CCLE_drts

DRProfile-class  DRProfSet is a class for managing dose-response information about cell lines from a pharmacogenomics dataset

Description

DRProfSet is a class for managing dose-response information about cell lines from a pharmacogenomics dataset
getDrugs extracts drug list
DRProfSet manages all data from a given cell line from a pharmacogenomics source

Usage

getDrugs(x)

DRProfSet(cell_line = "MCF7", dataset = "CCLE")

## S4 method for signature 'DRProfSet,missing'
plot(x, y, ...)

Arguments

x instance of DRProfSet
cell_line character(1) cell line name, entries in cell_lines_v1
dataset character(1) resource name, entries in datasets_v1
y for plot: not used
... not used

Value

getDrugs: character vector
instance of DRProfSet
Examples

```r
if (interactive()) trs = DRTraceSet() else trs = iriCCLE()
p = traces(trs)[[1]]
ps
getDrugs(ps)
if (interactive()) DRProfSet()
```

Description

DRTraceSet class manages dose-response information for a single cell line, multiple drugs
DRTraceSet constructor for multiple cell lines, single drug, single dataset

Usage

```r
## S4 method for signature 'DRTraceSet,missing'
plot(x, y, ...)

DRTraceSet(
  cell_lines = c("SK-ES-1", "TC-71", "MHH-ES-1", "HCC-56", "SK-HEP-1"),
  drug = "Irinotecan",
  dataset = "CCLE"
)
```

Arguments

- `x` for plot: instance of DRTraceSet
- `y` for plot: not used
- `...` not used
- `cell_lines` character vector of cell line names, must be found in ‘cell_lines_v1’ data of pogos package
- `drug` character(1) drug name in ‘compounds_v1’
- `dataset` character(1) dataset known to pharmacodb.pmgenomics.ca

Value

instance of DRTraceSet

Note

Will query pharmacodb for relevant dose-response information

Examples

```r
DRTraceSet()
```
**iriCCLE**

*obtain an example trace set stored locally, for irinotecan and selected cell lines*

**Description**

obtain an example trace set stored locally, for irinotecan and selected cell lines

**Usage**

```r
iriCCLE()
```

**Value**

an instance of DRTraceSet

**Examples**

```r
iri = iriCCLE()
iri
plot(iri)
```

---

**rxdbQuery_v1**

*very simple query formulation, build queries using endpoints of bhklab PharmacoDB API*

**Description**

very simple query formulation, build queries using endpoints of bhklab PharmacoDB API

**Usage**

```r
rxdbQuery_v1(
  ...,
  url = "https://pharmacodb.pmgenomics.ca/api/v1/",
  decoder = basicDecoder
)
```

**Arguments**

- `...`  
  typically a string representing an API endpoint, will be processed by unlist() and then to paste0 preceded by `url`

- `url`  
  of a PharmacoDB server API target

- `decoder`  
  a function of one argument that will be applied to API response (typically JSON)
**topEndpoints_v1**

**Value**

typically a list, dependent on decoder parameter

**Examples**

```r
tqout = rxdbQuery_v1('cell_lines') # yields 30; append '?all=true' to retrieve all
unlist(lapply(tqout, function(x) x[[2]]))
```

---

**topEndpoints_v1**

*enumerate top level endpoint terms for bhklab PharmacoDB API*

**Description**

enumerate top level endpoint terms for bhklab PharmacoDB API

**Usage**

```r
topEndpoints_v1()
```

**Value**

a character vector of available endpoints

**Examples**

```r
topEndpoints_v1()
```

---

**traces**

*trace extractor*

**Description**

trace extractor

**Usage**

```r
traces(x)
```

**Arguments**

```r
x               instance of DRTraceSet
```

**Value**

a list of DRProfile instances
Examples

```
iri = iriCCLE()
str(traces(iri)[[1]])
```

Description

Subscripting on DRProfSet extracts a profile for a single drug whose name constitutes the index.

Usage

```
## S4 method for signature 'DRProfSet,character,ANY,ANY'
x[i, j, ..., drop = TRUE]
```

Arguments

- `x`: instance of DRProfSet
- `i`: character(1) drug name
- `j`: not used
- `...`: not used
- `drop`: logical(1) not used

Value

A DRProfSet instance restricted to experiments involving the selected drug.
Index

* datasets
  compounds_v1, 3
  [,DRProfSet,character,ANY,ANY-method, 8

  basicDecoder, 2
  CCLE_drts (compounds_v1), 3
cell_lines_v1 (compounds_v1), 3
  compounds_v1, 3
  compoundsByCell, 2

data_v1 (compounds_v1), 3
  DRProfile-class, 4
  DRProfSet (DRProfile-class), 4
  DRProfSet-class (DRProfile-class), 4
  DRTraceSet (DRTraceSet-class), 5
  DRTraceSet-class, 5

  getDrugs (DRProfile-class), 4
  getDrugs, DRProfSet-method
    (DRProfile-class), 4

  iriCCLE, 6

  plot, DRProfSet, missing-method
    (DRProfile-class), 4
  plot, DRTraceSet, missing-method
    (DRTraceSet-class), 5

  rxdbQuery_v1, 6

tissues_v1 (compounds_v1), 3
topEndpoints_v1, 7
traces, 7