

# Package ‘pogos’

May 18, 2024

**Title** PharmacOGenomics Ontology Support

**Description** Provide simple utilities for querying bhklab PharmacODB, modeling API outputs, and integrating to cell and compound ontologies.

**Version** 1.25.0

**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), httr (>= 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** devel

**git\_last\_commit** 3204864

**git\_last\_commit\_date** 2024-04-30

**Repository** Bioconductor 3.20

**Date/Publication** 2024-05-17

## Contents

basicDecoder . . . . .	2
compoundsByCell . . . . .	2
compounds_v1 . . . . .	3
DRProfile-class . . . . .	4
DRTraceSet-class . . . . .	5

iriCCLE . . . . .	6
rxdbQuery_v1 . . . . .	6
topEndpoints_v1 . . . . .	7
traces . . . . .	7
[,DRProfSet,character,ANY,ANY-method . . . . .	8

<b>Index</b>	<b>9</b>
--------------	----------

---

basicDecoder	<i>convert binary output of GET()\$content to list</i>
--------------	--

---

### 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.pmgonomics.ca/api/v1/cell_lines')
unlist(cl)
```

---

compoundsByCell	<i>initial version of compound browser over pharmacDb cells</i>
-----------------	---

---

### Description

initial version of compound browser over pharmacDb cells

### Usage

```
compoundsByCell()
```

### Value

only used for side effect of running shiny app

**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**

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

---

compounds_v1	<i>compounds_v1: serialization of compounds info from PharmacoDb v1</i>
--------------	---

---

**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	<i>DRProfSet is a class for managing dose-response information about cell lines from a pharmacogenomics dataset</i>
-----------------	---

---

**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**

```

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

```

---

DRTraceSet-class	<i>DRTraceSet class manages dose-response information for a single cell line, multiple drugs</i>
------------------	--

---

**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**

```

## 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**

```
DRTraceSet()
```

---

iriCCLE	<i>obtain an example trace set stored locally, for irinotecan and selected cell lines</i>
---------	---

---

**Description**

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

**Usage**

```
iriCCLE()
```

**Value**

an instance of DRTraceSet

**Examples**

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

---

rxdbQuery_v1	<i>very simple query formulation, build queries using endpoints of bhklab PharmacODB API</i>
--------------	--

---

**Description**

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

**Usage**

```
rxdbQuery_v1(
  ...,
  url = "https://pharmacodb.pmggenomics.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)

**Value**

typically a list, dependent on decoder parameter

**Examples**

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

---

topEndpoints_v1	<i>enumerate top level endpoint terms for bhklab PharmacODB API</i>
-----------------	---

---

**Description**

enumerate top level endpoint terms for bhklab PharmacODB API

**Usage**

```
topEndpoints_v1()
```

**Value**

a character vector of available endpoints

**Examples**

```
topEndpoints_v1()
```

---

traces	<i>trace extractor</i>
--------	------------------------

---

**Description**

trace extractor

**Usage**

```
traces(x)
```

**Arguments**

x                    instance of DRTraceSet

**Value**

a list of DRProfile instances

**Examples**

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

---

[,DRProfSet,character,ANY,ANY-method

*subscripting on DRProfSet extracts a profile for a single drug whose name constitutes the index*

---

**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

CCLC\_drts (compounds\_v1), 3  
cell\_lines\_v1 (compounds\_v1), 3  
compounds\_v1, 3  
compoundsByCell, 2

datasets\_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