

# Package ‘gDRtestData’

January 2, 2025

**Title** gDRtestData - R data package with testing dose response data

**Version** 1.5.1

**Date** 2024-11-05

**Description** R package with internal dose-response test data. Package provides functions to generate input testing data that can be used as the input for gDR pipeline. It also contains qs files with MAE data processed by gDR.

**Depends** R (>= 4.2)

**Imports** checkmate, data.table

**Suggests** BiocStyle, gDRstyle (>= 1.1.5), knitr, qs, reshape2, rmarkdown, SummarizedExperiment, testthat, yaml

**URL** <https://github.com/gdrplatform/gDRtestData>,  
<https://gdrplatform.github.io/gDRtestData/>

**BugReports** <https://github.com/gdrplatform/gDRtestData/issues>

**biocViews** ExperimentData, TechnologyData

**VignetteBuilder** knitr

**License** Artistic-2.0

**Encoding** UTF-8

**Roxygen** list(markdown = TRUE)

**RoxygenNote** 7.3.1

**Config/testthat/edition** 3

**git\_url** <https://git.bioconductor.org/packages/gDRtestData>

**git\_branch** devel

**git\_last\_commit** 8be652c

**git\_last\_commit\_date** 2024-11-06

**Repository** Bioconductor 3.21

**Date/Publication** 2025-01-02

**Author** Aleksander Chlebowski [aut],  
 Bartosz Czech [aut] (ORCID: <<https://orcid.org/0000-0002-9908-3007>>),  
 Arkadiusz Gladki [cre, aut] (ORCID:  
 <<https://orcid.org/0000-0002-7059-6378>>),  
 Marc Hafner [aut] (ORCID: <<https://orcid.org/0000-0003-1337-7598>>),  
 Marcin Kamianowski [aut],  
 Sergiu Mocanu [aut],  
 Natalia Potocka [aut],  
 Dariusz Scigocki [aut],  
 Janina Smola [aut],  
 Allison Young [aut]

**Maintainer** Arkadiusz Gladki <[gladki.arkadiusz@gmail.com](mailto:gladki.arkadiusz@gmail.com)>

## Contents

gDRtestData-package . . . . .	2
add_concentration . . . . .	3
add_data_replicates . . . . .	4
add_day0_data . . . . .	4
cell_lines . . . . .	5
create_synthetic_cell_lines . . . . .	5
create_synthetic_drugs . . . . .	6
drugs . . . . .	6
generate_ec50 . . . . .	7
generate_e_inf . . . . .	7
generate_hill_coef . . . . .	8
generate_response_data . . . . .	8
get_test_dataset_paths . . . . .	9
prepareCodilutionData . . . . .	10
prepareComboMergedData . . . . .	10
prepareData . . . . .	11
prepareMergedData . . . . .	12
<b>Index</b>	<b>13</b>

---

gDRtestData-package	<i>gDRtestData: gDRtestData - R data package with testing dose response data</i>
---------------------	--

---

## Description

R package with internal dose-response test data. Package provides functions to generate input testing data that can be used as the input for gDR pipeline. It also contains qs files with MAE data processed by gDR.

**Value**

package help page

**Note**

To learn more about functions start with `help(package = "gDRtestData")`

**Author(s)**

**Maintainer:** Arkadiusz Gladki <gladki.arkadiusz@gmail.com> ([ORCID](#))

**Authors:**

- Aleksander Chlebowski
- Bartosz Czech ([ORCID](#))
- Marc Hafner ([ORCID](#))
- Marcin Kamianowski
- Sergiu Mocanu
- Natalia Potocka
- Dariusz Scigocki
- Janina Smola
- Allison Voung

**See Also**

Useful links:

- <https://github.com/gdrplatform/gDRtestData>
- <https://gdrplatform.github.io/gDRtestData/>
- Report bugs at <https://github.com/gdrplatform/gDRtestData/issues>

---

add\_concentration      *Add concentrations*

---

**Description**

Add concentrations

**Usage**

```
add_concentration(df_layout, concentrations = 10^(seq(-3, 1, 0.5)))
```

**Arguments**

`df_layout`      data.table that should contains the cell line, drug, concentration, and replicate columns along with the annotations that needs to be propagated

`concentrations`      vector of numeric concentrations that will be added to `df_layout`

**Value**

data.table with concentrations

**Examples**

```
cell_lines <- create_synthetic_cell_lines()
add_concentration(cell_lines)
```

---

add\_data\_replicates    *Add data replicates*

---

**Description**

Add data replicates

**Usage**

```
add_data_replicates(df_layout)
```

**Arguments**

df\_layout        data.table that should contains the cell line, drug, concentration, and replicate columns along with the annotations that needs to be propagated

**Value**

data.table with replicates

**Examples**

```
cell_lines <- create_synthetic_cell_lines()
add_data_replicates(cell_lines)
```

---

add\_day0\_data        *Add data with day 0*

---

**Description**

Add data with day 0

**Usage**

```
add_day0_data(df_merged, noise_level = 0.05)
```

**Arguments**

df\_merged      data.table with merged data  
noise\_level    numeric scalar with the level of noise added to the data

**Value**

data.table with day0 data

**Examples**

```
cell_lines <- create_synthetic_cell_lines()
drugs <- create_synthetic_drugs()
df_merged <- prepareData(cell_lines[seq_len(2), ], drugs[seq_len(4), ])
df_merged$Duration <- 72
df_merged$ReadoutValue <- 0
add_day0_data(df_merged)
```

---

cell_lines	<i>Cell lines</i>
------------	-------------------

---

**Description**

Cell lines

**Value**

data.table

**Examples**

```
path <- system.file("annotation_data", "cell_lines.csv", package = "gDRtestData")
data.table::fread(file = path)
```

---

create_synthetic_cell_lines	<i>Create data.table with synthetic cell lines</i>
-----------------------------	--

---

**Description**

Create data.table with synthetic cell lines

**Usage**

```
create_synthetic_cell_lines()
```

**Value**

data.table with synthetic cell lines

**Examples**

```
create_synthetic_cell_lines()
```

---

```
create_synthetic_drugs
```

*Create data.table with synthetic drugs*

---

**Description**

Create data.table with synthetic drugs

**Usage**

```
create_synthetic_drugs()
```

**Value**

data.table with synthetic drugs

**Examples**

```
create_synthetic_drugs()
```

---

drugs

*Drugs*

---

**Description**

Drugs

**Value**

data.table

**Examples**

```
path <- system.file("annotation_data", "drugs.csv", package = "gDRtestData")
data.table::fread(file = path)
```

---

generate_ec50	<i>Calculate EC50 metric</i>
---------------	------------------------------

---

**Description**

Calculate EC50 metric

**Usage**

```
generate_ec50(drugs, cell_lines)
```

**Arguments**

drugs	data.table with drugs
cell_lines	data.table with cell lines

**Value**

matrix with random EC50

**Examples**

```
generate_ec50(create_synthetic_drugs(), create_synthetic_cell_lines())
```

---

generate_e_inf	<i>Calculate E inf metric</i>
----------------	-------------------------------

---

**Description**

Calculate E inf metric

**Usage**

```
generate_e_inf(drugs, cell_lines)
```

**Arguments**

drugs	data.table with drugs
cell_lines	data.table with cell lines

**Value**

matrix with random E inf

**Examples**

```
generate_e_inf(create_synthetic_drugs(), create_synthetic_cell_lines())
```

---

```
generate_hill_coef
```

*Generate hill coefficient*

---

**Description**

Generate hill coefficient

**Usage**

```
generate_hill_coef(drugs, cell_lines)
```

**Arguments**

drugs	data.table with drugs
cell_lines	data.table with cell lines

**Value**

matrix with random hill coefficient

**Examples**

```
generate_hill_coef(create_synthetic_drugs(), create_synthetic_cell_lines())
```

---

```
generate_response_data
```

*Generate response data*

---

**Description**

Generate response data

**Usage**

```
generate_response_data(df_layout, noise_level = 0.1)
```

**Arguments**

df_layout	data.table that should contains the cell line, drug, concentration, and replicate columns along with the annotations that needs to be propagated
noise_level	numeric scalar with the level of noise added to the data



**Value**

data.table with response data

**Examples**

```
cell_lines <- create_synthetic_cell_lines()
drugs <- create_synthetic_drugs()
df_layout <- prepareData(cell_lines[seq_len(2), ], drugs[seq_len(4), ])
generate_response_data(df_layout)
```

---

`get_test_dataset_paths`  
*get\_test\_dataset\_paths*

---

**Description**

Returns named vector of absolute paths to test datasets.

**Usage**

```
get_test_dataset_paths(datasets_dir = NULL, pattern = "finalMAE_")
```

**Arguments**

`datasets_dir` path to directory with datasets (default NULL). If NULL, then `inst/testdata` directory from `gDRtestData` will be used.

`pattern` used to: (1) filter to `qs` files from the `dataset_dir` path and (2) prettify the labels of the files

**Value**

named vector of absolute paths

**Author(s)**

Kamil Foltynski <kamil.foltynski@contractors.roche.com>

**Examples**

```
get_test_dataset_paths()
path <- system.file("testdata", package = "gDRtestData", mustWork = TRUE)
get_test_dataset_paths(path)
```

---

prepareCodilutionData *prepareCodilutionData*

---

**Description**

Create data.table with input co-dilution data containing noise for testing purposes

**Usage**

```
prepareCodilutionData(  
  cell_lines,  
  drugs,  
  drugsIdx2 = 1,  
  conc = 10^(seq(-3, 1, 0.5)),  
  noise = 0.1  
)
```

**Arguments**

cell_lines	data.table with cell line info
drugs	data.table with drug info
drugsIdx2	numeric vector of ids for secondary drug (in drugs data.table)
conc	vector of doses
noise	number indicating level of noise

**Value**

data.table with input data for testing

**Examples**

```
prepareCodilutionData(create_synthetic_cell_lines()[seq_len(2), ],  
  create_synthetic_drugs()[seq_len(4), ])
```

---

prepareComboMergedData  
*prepareComboMergedData*

---

**Description**

Create data.table with input combination data containing noise for testing purposes

**Usage**

```
prepareComboMergedData(
  cell_lines,
  drugs,
  drugsIdx1 = 2:4,
  drugsIdx2 = c(26, 26, 26),
  concentration = c(0, 0.2, 1),
  noise = 0.1,
  modifyDf2 = FALSE
)
```

**Arguments**

cell_lines	data.table with cell line info
drugs	data.table with drug info
drugsIdx1	numeric vector of ids for primary drug
drugsIdx2	numeric vector of ids for secondary drug
concentration	numeric vector of doses
noise	number indicating level of noise
modifyDf2	Boolean indicating if the table should be modified to keep reverse single agent data

**Value**

data.table with input data for testing

**Examples**

```
prepareComboMergedData(create_synthetic_cell_lines(), create_synthetic_drugs())
```

---

prepareData	<i>prepareData</i>
-------------	--------------------

---

**Description**

Create data.table with input data for testing purposes

**Usage**

```
prepareData(cell_lines, drugs, conc = 10^(seq(-3, 1, 0.5)))
```

**Arguments**

cell_lines	data.table with cell line info
drugs	data.table with drug info
conc	vector of doses

**Value**

data.table with input data for testing

**Examples**

```
prepareData(create_synthetic_cell_lines(), create_synthetic_drugs())
```

---

prepareMergedData	<i>prepareMergedData</i>
-------------------	--------------------------

---

**Description**

Create data.table with input data containing noise for testing purposes

**Usage**

```
prepareMergedData(cell_lines, drugs, noise = 0.1)
```

**Arguments**

cell_lines	data.table with cell line info
drugs	data.table with drug info
noise	number indicating level of noise

**Value**

data.table with input data for testing

**Examples**

```
prepareMergedData(create_synthetic_cell_lines(), create_synthetic_drugs())
```

# Index

- \* **data**
  - cell\_lines, [5](#)
  - drugs, [6](#)
- \* **generate\_data**
  - add\_concentration, [3](#)
  - add\_data\_replicates, [4](#)
  - add\_day0\_data, [4](#)
  - create\_synthetic\_cell\_lines, [5](#)
  - create\_synthetic\_drugs, [6](#)
- \* **generate\_results**
  - generate\_e\_inf, [7](#)
  - generate\_ec50, [7](#)
  - generate\_hill\_coef, [8](#)
  - generate\_response\_data, [8](#)
- \* **generate\_test\_data**
  - get\_test\_dataset\_paths, [9](#)
  - prepareCodilutionData, [10](#)
  - prepareComboMergedData, [10](#)
  - prepareData, [11](#)
  - prepareMergedData, [12](#)
- \* **internal**
  - cell\_lines, [5](#)
  - drugs, [6](#)
  - gDRtestData-package, [2](#)

[add\\_concentration, 3](#)

[add\\_data\\_replicates, 4](#)

[add\\_day0\\_data, 4](#)

[cell\\_lines, 5](#)

[create\\_synthetic\\_cell\\_lines, 5](#)

[create\\_synthetic\\_drugs, 6](#)

[drugs, 6](#)

[gDRtestData \(gDRtestData-package\), 2](#)

[gDRtestData-package, 2](#)

[generate\\_e\\_inf, 7](#)

[generate\\_ec50, 7](#)

[generate\\_hill\\_coef, 8](#)

[generate\\_response\\_data, 8](#)

[get\\_test\\_dataset\\_paths, 9](#)

[prepareCodilutionData, 10](#)

[prepareComboMergedData, 10](#)

[prepareData, 11](#)

[prepareMergedData, 12](#)