

Package: gDR (via r-universe)

October 9, 2026

Type Package

Title Umbrella package for R packages in the gDR suite

Version 1.11.11

Date 2026-10-08

Description Package is a part of the gDR suite. It reexports functions from other packages in the gDR suite that contain critical processing functions and utilities. The vignette walks through the full processing pipeline for drug response analyses that the gDR suite offers.

License Artistic-2.0

Depends R (>= 4.2), gDRcore (>= 1.7.1), gDRimport (>= 1.7.1), gDRutils (>= 1.7.1)

Imports BumpyMatrix, checkmate, data.table, MultiAssayExperiment, qs2, rmarkdown, S4Vectors, stats, SummarizedExperiment, tools, utils, whisker

Suggests BiocStyle, futile.logger, gDRstyle (>= 1.7.1), gDRtestData (>= 1.7.1), ggplot2, kableExtra, knitr, markdown, mockery, patchwork, purrr, RColorBrewer, stringr, summarytools, testthat, writexl, yaml

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

BugReports <https://github.com/gdrplatform/gDR/issues>

biocViews Software, DataImport, ShinyApps

VignetteBuilder knitr

ByteCompile TRUE

LazyLoad yes

Roxygen list(markdown = TRUE)

SwitchrLibrary gDR

DeploySubPath gDR

Encoding UTF-8

Config/roxygen2/version 8.0.0
Config/pak/sysreqs
cmake make libicu-dev libuv1-dev libxml2-dev libssl-dev libnode-dev libx11-dev zlib1g-dev
Repository <https://bioc.r-universe.dev>
Date/Publication 2026-10-09 06:04:05 UTC
RemoteUrl <https://github.com/bioc/gDR>
RemoteRef HEAD
RemoteSha d0522c982283f99a820f39d76df8b2f4213df2c5

Contents

add_treatment_column	2
apply_custom_metadata_tags	3
cast_to_wide	3
create_symlink_to_current	4
filter_treatments	5
generate_incucyte_yaml_draft	5
get_next_version_dir	6
import_data	6
resolve_incucyte_config_path	7
run_report	8
small_combo_data	13
small_data	14
Index	15

add_treatment_column	<i>Add Treatment Column to a Data Table</i>
----------------------	---

Description

Creates a formatted treatment label from available drug/concentration columns. It dynamically detects which columns (e.g. DrugName, DrugName_2) are present using environment identifiers from gDRutils.

Usage

```
add_treatment_column(  
  dt,  
  control_name = "DMSO",  
  vehicle_name = "vehicle",  
  unit = "uM"  
)
```

Arguments

dt	data.table; The input data.table.
control_name	character; The name to use when all drugs are vehicles. Default is 'DMSO'.
vehicle_name	character; The identifier for a vehicle/control substance. Default is 'vehicle'.
unit	character; The concentration unit to append to the value. Default is 'uM'.

Value

A copy of the data.table dt with a new "treatment" column.

```
apply_custom_metadata_tags
```

Apply Custom Metadata Tags to Drug Names

Description

Finds custom metadata columns by convention (e.g., 'drug_time', 'drug3_washout') and appends their values as tags to the corresponding DrugName column.

Usage

```
apply_custom_metadata_tags(dt)
```

Arguments

dt	data.table; The input data.table containing drug names and custom metadata.
----	---

Value

A modified data.table with tags appended to the relevant drug name columns.

```
cast_to_wide
```

Reshape a data.table from long to wide format

Description

This function utilizes the powerful dcast function from the data.table package to pivot data. It turns unique values from a specified column into new columns in the output data.table.

Usage

```
cast_to_wide(long_data, id_vars, col_var, val_var)
```

Arguments

long_data	data.table; A data.table in long format.
id_vars	character vector; Column names that uniquely identify each row (e.g. c("CellLineName", "range")).
col_var	character; The column whose unique values will become the new column headers (e.g., "treatment").
val_var	character; The column whose values will fill the cells (e.g., "rate").

Value

A new data.table in wide format.

create_symlink_to_current

Create symbolic link to current version

Description

This function creates a symbolic link (current or current_verbose) that points to the latest version directory.

Usage

```
create_symlink_to_current(base_dir, latest_version, verbose_mode = FALSE)
```

Arguments

base_dir	Character. The base directory where the symbolic link should be created.
latest_version	Character. The path to the latest version directory.
verbose_mode	Logical, indicating if the run is for a verbose report.

Examples

```
# Assume "/path/to/dir/v3" is the latest version
## Not run:
create_symlink_to_current("/path/to/dir", "/path/to/dir/v3")

## End(Not run)
```

filter_treatments	<i>Filter Treatments from SummarizedExperiment</i>
-------------------	--

Description

Filters a SummarizedExperiment object to extract normalized cell counts for specific treatments and formats them into a long data.table for plotting.

Usage

```
filter_treatments(se, sel_trt)
```

Arguments

se	SummarizedExperiment; The object containing the assay data.
sel_trt	character vector; A list of treatment names to filter and extract.

Value

A data.table containing columns: treatment, CellCount_norm, CellLineName, and Duration.

generate_incucyte_yaml_draft	<i>Generate a Draft YAML Configuration for Incucyte Analysis</i>
------------------------------	--

Description

Automatically creates a boilerplate YAML configuration file based on the processed input data. It provides working example groupings for treatment_comparisons and a commented-out dictionary of all available treatment names.

Usage

```
generate_incucyte_yaml_draft(  
  dt,  
  out_file,  
  early_period = c(24, 48),  
  late_period = c(120, 144)  
)
```

Arguments

dt	data.table; The preprocessed data.table (e.g., after import).
out_file	character; Path where the draft YAML should be saved.
early_period	numeric vector of length 2; Default c(24, 48).
late_period	numeric vector of length 2; Default c(120, 144).

Value

Invisible path to the generated file.

get_next_version_dir	<i>Get next version directory</i>
----------------------	-----------------------------------

Description

This function determines the next version directory name based on existing version directories. The naming convention depends on whether the report is verbose.

Usage

```
get_next_version_dir(base_dir, verbose_mode = FALSE)
```

Arguments

base_dir	Character. The base directory where version directories are stored.
verbose_mode	Logical, indicating if the run is for a verbose report.

Value

Character. The path to the next version directory (e.g., v1 or v1_verbose).

Examples

```
# Assuming "/path/to/dir" has subdirectories "v1", "v2"
# get_next_version_dir("/path/to/dir") # Returns "/path/to/dir/v3"
```

import_data	<i>Import raw data</i>
-------------	------------------------

Description

Import raw data

Usage

```
import_data(
  manifest_file,
  template_file,
  results_file,
  instrument = "EnVision",
  cell_line_annotation = NULL,
  drug_annotation = NULL
)
```

Arguments

manifest_file	character, file path(s) to manifest(s)
template_file	character, with datapaths and names of results file(s) or character with file path of templates file(s)
results_file	character, with datapaths and names of results file(s) or character with file path of results file(s)
instrument	string with type of instrument used
cell_line_annotation	optional data.table with cell line annotations; if NULL (default), annotations are looked up from gDRinternal or gDRtestData
drug_annotation	optional data.table with drug annotations; if NULL (default), annotations are looked up from gDRinternal or gDRtestData

Value

a data.table

Examples

```
td <- get_test_data()
i_df <- import_data(manifest_path(td), template_path(td), result_path(td))
```

resolve_incucyte_config_path

Resolve the configuration file an Incucyte report reads

Description

An Incucyte run either carries a configuration file supplied by the caller, staged into raw_data/ under its own name, or none, in which case the first step generates a draft under a fixed name and that becomes the configuration for the run. This resolves both cases to one existing file, so that a report can never quietly analyse generated defaults while the caller believes their own configuration is in force.

Usage

```
resolve_incucyte_config_path(
  configuration_file_path,
  output_dir,
  default_name = "time_course_plot_params.yml"
)
```

Arguments

configuration_file_path string; staged path of the configuration file supplied by the caller, or "" when none was supplied.

output_dir string; versioned output directory of the run.

default_name string; name of the configuration file the first step generates, looked up in raw_data/ when the caller supplied none.

Value

Path of an existing configuration file.

Examples

```
## Not run:
resolve_incucyte_config_path("/out/v1/raw_data/my_windows.yml", "/out/v1")
resolve_incucyte_config_path("", "/out/v1")

## End(Not run)
```

run_report

Run a gDR report pipeline

Description

Infrastructure-agnostic driver for the gDR reporting pipeline. It stages all input files into a self-contained, versioned output directory, validates them, and renders the requested RMarkdown steps through whisker and rmarkdown. The function can process different types of input data, including QCS ID, PRISM data, and a MultiAssayExperiment (MAE) object or a path to a .qs2 file containing an MAE object.

Usage

```
run_report(
  qcs_id = NULL,
  long_table = NULL,
  manifest = NULL,
  treatment = NULL,
  raw_data = NULL,
  prism_data_path = NULL,
  cell_line_data_path = NULL,
  treatment_data_path = NULL,
  drug_annotation_path = NULL,
  cell_line_annotation_path = NULL,
  meta_data_path = NULL,
  feat_data_path = NULL,
```

```

feature_sets = c("CRISPRGeneEffect", "OmicsExpressionProteinCodingGenesTPMLogp1",
  "OmicsSomaticMutationsMatrixHotspot", "OmicsSomaticMutationsMatrixDamaging",
  "OmicsCNGene"),
metadata_columns = c("OncotreeLineage", "PatientRace"),
output_dir,
viz_format = "svg",
rmd_template_path,
steps = seq_len(3),
mae = NULL,
special_sections = NULL,
verbose_mode = FALSE,
with_decoding = TRUE,
dpi = 96,
embed = TRUE,
configuration_file_path = NULL,
remove_on_failure = TRUE,
resume = FALSE,
extra_render_params = list(),
on_resolve_inputs = NULL,
on_start = NULL,
on_before_render = NULL,
on_success = NULL,
on_failure = NULL
)

```

Arguments

<code>qcs_id</code>	Character vector. One or more QCS IDs. Provide these if you want to process data using QCS IDs.
<code>long_table</code>	Character. Path to a single CSV/TSV file with a tidy 'long table' of drug-response data (the same shape as the <code>data_imported</code> object). Provide this to import tabular data directly, without going through a screening system. The file must contain the <code>Gnumber</code> , <code>clid</code> , <code>Duration</code> , <code>Concentration</code> , and <code>ReadoutValue</code> columns (plus optional <code>Gnumber_2/Concentration_2</code> for combinations). See load_long_table .
<code>manifest</code>	Character. Path to the manifest file(s). Can be a single path, a comma-separated string of paths, or a regular expression pattern to match files in a directory (e.g., <code>"/path/to/manifests/*.xlsx"</code>). Required if not using <code>qcs_id</code> or <code>prism_data_path</code> .
<code>treatment</code>	Character. Path to the treatment file(s). Can be a single path, a comma-separated string of paths, or a regular expression pattern. Required if not using <code>qcs_id</code> or <code>prism_data_path</code> .
<code>raw_data</code>	Character. Path to the raw data file(s). Can be a single path, a comma-separated string of paths, or a regular expression pattern. Required if not using <code>qcs_id</code> or <code>prism_data_path</code> .
<code>prism_data_path</code>	Character. Path to the PRISM data file. Provide this if you want to process PRISM data.

cell_line_data_path	Character. Path to the cell line data file. Required if using prism_data_path and processing PRISM level 6 data.
treatment_data_path	Character. Path to the treatment data file. Required if using prism_data_path and processing PRISM level 6 data.
drug_annotation_path	Character. Path to the drug annotation CSV file. This param can be used for using custom annotations for drugs. Annotation file for drugs should contain Gnumber, DrugName, and drug_moa columns.
cell_line_annotation_path	Character. Path to the cell line annotation CSV file. This param can be used for using custom annotations for cell lines. Annotation file for cell line should contain clid, CellLineName, Tissue, ReferenceDivisionTime, subtype, and parental_identifier columns.
meta_data_path	Character. Path to metadata file describing all cancer models/cell lines which are referenced by a dataset contained within the DepMap portal. It is usually a file named Model.csv. Required if using prism_data_path
feat_data_path	Character. Path to the directory containing of the molecular feature set files to load from DepMap.
feature_sets	Character. Vector containing the names of the molecular feature sets to load from DepMap. These names should also correspond to the file names containing the feature data in feat_data_path (without the extension, which is assumed to be csv) Required if using special_sections equal "PRISM"
metadata_columns	Character. Metadata names from file pointed by meta_data_path Required if using special_sections equal "PRISM"
output_dir	Character. Directory to store reports and output data. This is a required parameter.
viz_format	Character. Format for visualizations. Default is "svg".
rmd_template_path	Character vector. One or more directories holding the RMarkdown templates. Directories are searched in order and the first match wins, so a deployment can both override individual templates and supply steps that the other directories do not carry (for instance a publication step living outside the open-source template set). This is a required parameter.
steps	Integer vector. Steps to run in the pipeline: 1: data import, 2: processing and QC, 3: analysis, 4: publish results. Default is 1:3. Note: If mae is provided, steps 1 and 2 are automatically skipped.
mae	MultiAssayExperiment or Character. A MultiAssayExperiment object or a path to a .qs2 file containing a MultiAssayExperiment object.
special_sections	String or NULL. Provide this if you want to process data, type PRISM or chemical genomics, starting from mae to see a section specific to them. One of: "PRISM", "chemical_genomics".

verbose_mode	Logical. Whether to generate a detailed report, including detailed visualizations by cell line name and drug name. This also affects the output directory naming.
with_decoding	logical whether the feature OmicsArmLevelCNA, OmicsSomaticMutationsMatrixHotspot and OmicsSomaticMutationsMatrixDamaging should be encoded into a 0-1 scheme
dpi	number defining dpi of all images in report
embed	Logical. Whether to generate a self-contained HTML report.
configuration_file_path	Character. Path to the configuration file. Currently, required only for Incucyte data
remove_on_failure	Logical. Whether to remove the output directory if the pipeline fails. Default is TRUE. If set to FALSE, incomplete output will be kept for debugging.
resume	Logical. Whether to resume execution in the existing 'current' output directory instead of creating a new version. Default is FALSE.
extra_render_params	Named list. Additional parameters injected into every RMarkdown template alongside the ones derived from the arguments above. Use it to pass values that are specific to a given deployment (e.g. dataset registry metadata) without adding them to the signature of this function. Values are also persisted in args.qs2. Names colliding with parameters computed by this function are dropped with a warning. Default is an empty list.
on_resolve_inputs	Function or NULL. Hook called before the data type is determined, used to resolve input paths from an external source. It receives args (the list of input arguments) and extra_render_params, and must return a list with exactly these two elements, which then replace the originals. Only the fields describing the inputs themselves (e.g. manifest, treatment, raw_data) may be rewritten; rewriting any of the run-control arguments (output_dir, rmd_template_path, steps, verbose_mode, resume, remove_on_failure) raises an error. This is a hard boundary rather than a convenience check: rewriting steps would change what the run actually renders and could append the publication step behind the back of the caller that decided which steps to run. Default is NULL (no resolution).
on_start	Function or NULL. Hook called once the output directory exists but before any rendering starts, e.g. to notify the user that the run began. It receives output_dir. Errors raised by this hook abort the run. Default is NULL.
on_before_render	Function or NULL. Hook called after the data type is determined and before any file is staged, e.g. to verify connectivity to external services. It receives data_type. Errors raised by this hook abort the run. Default is NULL.
on_success	Function or NULL. Hook called after all steps rendered successfully, e.g. to publish the results or notify the user. It receives output_dir. Errors raised by this hook are caught and reported as a warning, so that a failing notification cannot discard an otherwise successful run. Default is NULL.

`on_failure` Function or NULL. Hook called when the run fails, before the output directory is cleaned up, e.g. to notify the user. It receives `output_dir` and `error_message`. Errors raised by this hook are caught and reported as a message. Default is NULL.

Details

Every integration with external infrastructure (dataset registries, notification systems, authentication, service health checks) is delegated to optional hooks, so that the driver itself only depends on `checkmate`, `qs2`, `whisker`, `rmarkdown` and the open-source `gDR` packages. See `run_gDR_with_report()` for a wrapper that supplies these hooks for a specific deployment.

The report templates shipped in `inst/report_templates` additionally attach `gDRplots` when they are rendered. That package is deliberately absent from `Suggests`: it is not released on CRAN or Bioconductor yet, and an unresolvable soft dependency breaks dependency resolution for everything that installs this package. Install it from <https://github.com/gdrplatform/gDRplots> before rendering the bundled templates, or point `rmd_template_path` at templates that do not need it.

By default, if the pipeline fails at any point, it will automatically clean up by removing the output directory it created for the run. This can be disabled for debugging.

Value

None. The function generates reports and saves them to the specified output directory.

Examples

```
## Not run:
templates <- "path/to/report_templates"

# Standard run from bench files (Runs steps 1-3)
run_report(
  manifest = "path/to/manifest.xlsx",
  treatment = "path/to/treatments.xlsx",
  raw_data = "path/to/plate1.xlsx,path/to/plate2.xlsx",
  output_dir = "~/test_report",
  rmd_template_path = templates
)

# Run from a pre-existing MAE object
run_report(
  mae = "path/to/mae.qs2",
  output_dir = "~/test_report_mae",
  rmd_template_path = templates
)

# Run with a hook notified before rendering starts
run_report(
  manifest = "path/to/manifest.xlsx",
  treatment = "path/to/treatments.xlsx",
  raw_data = "path/to/plate1.xlsx",
  output_dir = "~/test_report_hooked",
  rmd_template_path = templates,
```

```

    on_before_render = function(data_type) message("About to render ", data_type)
  )

  # Assemble the template set from two directories: the first one wins, so it can both
  # override individual steps and contribute steps the shared set does not carry.
  run_report(
    mae = "path/to/mae.qs2",
    output_dir = "~/test_report_layered",
    rmd_template_path = c("path/to/deployment_templates", templates),
    steps = 3
  )

  ## End(Not run)

```

small_combo_data

*Small data.table with raw combo data used for processing via gDR***Description**

A dataset containing the ReadoutValues for combo experiments made up of 3 drugs, 2 co-drugs, and 2 cell lines

Usage

```
data(small_combo_data)
```

Format

A data frame with 3600 rows and 16 variables:

Barcode barcode
clid cell line id
CellLineName cell line name
Tissue cell line tissue
ReferenceDivisionTime cell line division time
Gnumber drug id
DrugName drug name
drug_moa drug mode of action
Concentration drug concentration
Gnumber_2 co-drug id
DrugName_2 co-drug name
drug_moa_2 co-drug mode of action
Concentration_2 co-drug concentration
ReadoutValue readout value
BackgroundValue background value
Duration duration

Value

data.table

small_data	<i>Small data.table with raw data used for processing via gDR</i>
------------	---

Description

A dataset containing the ReadoutValues for single-agent experiments made up of 10 drugs and 10 cell lines

Usage

data(small_data)

Format

A data frame with 3300 rows and 12 variables:

- Barcode** barcode
- clid** cell line id
- CellLineName** cell line name
- Tissue** cell line tissue
- ReferenceDivisionTime** cell line division time
- Gnumber** drug id
- DrugName** drug name
- drug_moa** drug mode of action
- Concentration** drug concentration
- ReadoutValue** readout value
- BackgroundValue** backgroud value
- Duration** duration

Value

data.table

Index

- * **data**
 - small_combo_data, [13](#)
 - small_data, [14](#)
- * **import**
 - import_data, [6](#)
- * **incucyte**
 - add_treatment_column, [2](#)
 - apply_custom_metadata_tags, [3](#)
 - cast_to_wide, [3](#)
 - filter_treatments, [5](#)
 - generate_incucyte_yaml_draft, [5](#)
 - resolve_incucyte_config_path, [7](#)
- * **report**
 - create_symlink_to_current, [4](#)
 - get_next_version_dir, [6](#)
 - run_report, [8](#)

add_treatment_column, [2](#)
apply_custom_metadata_tags, [3](#)

cast_to_wide, [3](#)
create_symlink_to_current, [4](#)

filter_treatments, [5](#)

generate_incucyte_yaml_draft, [5](#)
get_next_version_dir, [6](#)

import_data, [6](#)

load_long_table, [9](#)

resolve_incucyte_config_path, [7](#)
run_report, [8](#)

small_combo_data, [13](#)
small_data, [14](#)