

# Package ‘padma’

July 28, 2025

**Type** Package

**Title** Individualized Multi-Omic Pathway Deviation Scores Using  
Multiple Factor Analysis

**Version** 1.18.1

**Depends** R (>= 4.1.0), SummarizedExperiment, S4Vectors

**Imports** FactoMineR, MultiAssayExperiment, methods, graphics, stats,  
utils

**Suggests** testthat, BiocStyle, knitr, rmarkdown, KEGGREST, missMDA,  
ggplot2, ggrepel, car, cowplot, reshape2

**Description** Use multiple factor analysis to calculate individualized pathway-centric scores of deviation with respect to the sampled population based on multi-omic assays (e.g., RNA-seq, copy number alterations, methylation, etc). Graphical and numerical outputs are provided to identify highly aberrant individuals for a particular pathway of interest, as well as the gene and omics drivers of aberrant multi-omic profiles.

**License** GPL (>=3)

**URL** <https://github.com/andreamrau/padma>

**https** [//github.com/andreamrau/padma/issues](https://github.com/andreamrau/padma/issues)

**biocViews** Software, StatisticalMethod, PrincipalComponent,  
GeneExpression, Pathways, RNASeq, BioCarta, MethylSeq

**Encoding** UTF-8

**LazyData** TRUE

**RoxygenNote** 7.3.1

**VignetteBuilder** knitr

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

**git\_branch** RELEASE\_3\_21

**git\_last\_commit** be1bb74

**git\_last\_commit\_date** 2025-05-08

**Repository** Bioconductor 3.21

**Date/Publication** 2025-07-27

**Author** Andrea Rau [cre, aut] (ORCID: <<https://orcid.org/0000-0001-6469-488X>>),  
Regina Manansala [aut],  
Florence Jaffrézic [ctb],  
Denis Laloë [aut],  
Paul Auer [aut]

**Maintainer** Andrea Rau <[andrea.rau@inrae.fr](mailto:andrea.rau@inrae.fr)>

Contents

|                                       |           |
|---------------------------------------|-----------|
| factorMap . . . . .                   | 2         |
| LUAD_subset . . . . .                 | 4         |
| mirtarbase . . . . .                  | 5         |
| msigdb . . . . .                      | 5         |
| omicsContrib . . . . .                | 6         |
| padma . . . . .                       | 7         |
| padmaResults-class . . . . .          | 10        |
| padmaRun . . . . .                    | 11        |
| pathway_name . . . . .                | 13        |
| summary,padmaResults-method . . . . . | 15        |
| <b>Index</b>                          | <b>17</b> |

---

|           |                                                                                             |
|-----------|---------------------------------------------------------------------------------------------|
| factorMap | <i>Plot an MFA factor map for individuals or partial factor map based on padma analysis</i> |
|-----------|---------------------------------------------------------------------------------------------|

---

Description

Produce an MFA factor map for individuals, or MFA partial factor map for a given individual, for a pair of dimensions provided by the user.

Usage

```
factorMap(  
  padma_obj,  
  partial_id = NULL,  
  dim_x = 1,  
  dim_y = 2,  
  plot_ellipse = TRUE,  
  ggplot = TRUE,  
  repel_labels = ifelse(ggplot == TRUE, TRUE, FALSE)  
)
```

**Arguments**

|              |                                                                      |
|--------------|----------------------------------------------------------------------|
| padma_obj    | Output from running the padma function (with 'full_results = TRUE')  |
| partial_id   | Index or sample name to be plotted for a partial factor map.         |
| dim_x        | Dimension number of the MFA to be plotted on the x-axis.             |
| dim_y        | Dimension number of the MFA to be plotted on the y-axis.             |
| plot_ellipse | If TRUE, superimpose a normal confidence ellipsis on the factor map. |
| ggplot       | If TRUE, use ggplot2 for plotting                                    |
| repel_labels | If TRUE, use ggrepel to repel sample labels from each other          |

**Value**

Plot, or factor map of class ggplot if ggplot2 = TRUE.

**Examples**

```

LUAD_subset <- padma::LUAD_subset
## Create MultiAssayExperiment object with LUAD data
omics_data <-
  list(rnaseq = as.matrix(LUAD_subset$rnaseq),
       methyl = as.matrix(LUAD_subset$methyl),
       mirna = as.matrix(LUAD_subset$mirna),
       cna = as.matrix(LUAD_subset$cna))
pheno_data <-
  data.frame(LUAD_subset$clinical,
             row.names = LUAD_subset$clinical$bcr_patient_barcode)
mae <-
  suppressMessages(
    MultiAssayExperiment::MultiAssayExperiment(
      experiments = omics_data, colData = pheno_data))

## Run padma
run_padma <-
  padma(mae, gene_map = padma::mirtarbase,
        pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY", verbose = FALSE)

summary(run_padma)

## padma plots
## Not run:
factorMap(run_padma, dim_x = 1, dim_y = 2)
factorMap(run_padma, dim_x = 1, dim_y = 2,
          partial_id = "TCGA-78-7536")
omicsContrib(run_padma, max_dim = 10)

## End(Not run)

```

---

|             |                                                                              |
|-------------|------------------------------------------------------------------------------|
| LUAD_subset | <i>Subset of batch-corrected multi-omic TCGA data in lung adenocarcinoma</i> |
|-------------|------------------------------------------------------------------------------|

---

## Description

List of multi-omic (RNA-seq, copy number alterations, methylation, and miRNA-seq) and phenotypic data in 144 individuals in the TCGA-LUAD data for the 13 genes in the D4-GDI signaling pathway.

## Usage

```
data(LUAD_subset)
```

## Format

A named list of five objects of class `data.frame` containing a subset of the batch-corrected multi-omic TCGA data from lung adenocarcinoma, corresponding to the 13 genes in the D4 GDI signaling pathway: 'clinical' is of dimension 144 x 55 and contains clinical variables for the 144 individuals. 'rnaseq', 'methylation', 'cna', and 'mirna' are of dimension 13 (genes) or 1 (miRNAs) x 144 (samples), where the row names contain the gene symbol or miRNA name.

## Source

TCGA

## References

The Cancer Genome Atlas Research Network (2014) Nature 511, 543-550. <https://doi.org/10.1038/nature13385>.

Rau et al. (2019) bioRxiv, <https://doi.org/10.1101/827022>.

## Examples

```
data(LUAD_subset)
head(LUAD_subset)
```

---

mirtarbase*Curated miR-target interaction predictions from miRTarBase*

---

**Description**

Data.frame of 10,754 predicted miRNA gene targets from miRTarBase (version 7.0), filtered to include only predictions with the 'Functional MTI' support type.

**Usage**

```
data(mirtarbase)
```

**Format**

An object of class `data.frame` with two columns: `miRNA`, which provides the miRNA identifier (e.g., 'hsa-miR-20a-5p') and `Target Gene`, which provides the corresponding predicted gene target.

**Source**

<http://mirtarbase.mbc.nctu.edu.tw/php/index.php>

**References**

Chou et al. (2018) Nucleic Acids Research 46, D296-D302. <https://doi.org/10.1093/nar/gkx1067>.

**Examples**

```
data(mirtarbase)
head(mirtarbase)
```

---

msigdb*MSigDB canonical pathways and corresponding gene lists*

---

**Description**

Data.frame of 1322 pathways and corresponding gene symbols included in the MSigDB canonical pathways curated gene set catalog, which includes genes whose products are involved in metabolic and signaling pathways reported in curated public databases. This specifically corresponds to the 'C2 curated gene sets' catalog from MSigDB v5.2 available at <http://bioinf.wehi.edu.au/software/MSigDB/> as described in the limma Bioconductor package.

**Usage**

```
data(msigdb)
```

## Format

An object of class `data.frame` with two columns: `geneset`, which provides the 1322 MSigDB curated pathway names (e.g., 'c2\_cp\_BIOCARTA\_41BB\_PATHWAY') and `symbol`, which provides the comma-separated corresponding list of gene symbols.

## Source

MSigDB Gene sets <http://bioinf.wehi.edu.au/software/MSigDB/>

## References

Liberzon et al. (2011) Bioinformatics 27:12, 1739-1740. <https://doi.org/10.1093/bioinformatics/btr260>.

## Examples

```
data(msigdb)
head(msigdb)
```

---

|              |                                                                                       |
|--------------|---------------------------------------------------------------------------------------|
| omicsContrib | <i>Plot the omics contribution per MFA axis and the overall weighted contribution</i> |
|--------------|---------------------------------------------------------------------------------------|

---

## Description

Plot barplots indicating the percent contribution of each omics to each MFA dimension, as well as the overall weighted (by eigenvalue) percent contribution to the full analysis.

## Usage

```
omicsContrib(
  padma_obj,
  max_dim = min(10, nrow(MFA_results(padma_obj)$eig)),
  ggplot = TRUE
)
```

## Arguments

|                        |                                                                                                 |
|------------------------|-------------------------------------------------------------------------------------------------|
| <code>padma_obj</code> | Output from running the <code>padma</code> function (with ' <code>full_results = TRUE</code> ') |
| <code>max_dim</code>   | Maximum dimension number of the MFA to be plotted                                               |
| <code>ggplot</code>    | If TRUE, use <code>ggplot2</code> for plotting (and <code>cowplot</code> for combining ggplots) |

## Value

Barplots of percent variance contribution, optionally of class `ggplot`.

## Examples

```

LUAD_subset <- padma::LUAD_subset
## Create MultiAssayExperiment object with LUAD data
omics_data <-
  list(rnaseq = as.matrix(LUAD_subset$rnaseq),
       methyl = as.matrix(LUAD_subset$methyl),
       mirna = as.matrix(LUAD_subset$mirna),
       cna = as.matrix(LUAD_subset$cna))
pheno_data <-
  data.frame(LUAD_subset$clinical,
             row.names = LUAD_subset$clinical$bcr_patient_barcode)
mae <-
  suppressMessages(
    MultiAssayExperiment::MultiAssayExperiment(
      experiments = omics_data, colData = pheno_data))

## Run padma
run_padma <-
  padma(mae, gene_map = padma::mirtarbase,
        pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY", verbose = FALSE)

summary(run_padma)

## padma plots
## Not run:
factorMap(run_padma, dim_x = 1, dim_y = 2)
factorMap(run_padma, dim_x = 1, dim_y = 2,
          partial_id = "TCGA-78-7536")
omicsContrib(run_padma, max_dim = 10)

## End(Not run)

```

---

padma

*Calculate individualized deviation scores from multi-omic data*

---

## Description

This is the primary user interface for the padma package. Generic S4 methods are implemented to calculate individualized pathway deviation scores on the basis of matched, multi-omic data. The supported classes for input are list and MultiAssayExperiment. The output of padma is an S4 object of class padmaResults.

## Usage

```

padma(object, ...)

## S4 method for signature 'list'
padma(
  object,

```

```

colData,
gene_map = padma::mirtarbase,
base_ids = NULL,
supp_ids = NULL,
pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY",
impute = FALSE,
variance_threshold = 1e-04,
full_results = TRUE,
verbose = TRUE,
...
)

## S4 method for signature 'MultiAssayExperiment'
padma(
  object,
  gene_map = padma::mirtarbase,
  base_ids = NULL,
  supp_ids = NULL,
  pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY",
  impute = FALSE,
  variance_threshold = 1e-04,
  full_results = TRUE,
  verbose = TRUE,
  ...
)

```

## Arguments

|          |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
|----------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| object   | Matched multi-omic data. May be provided as (1) a <code>MultiAssayExperiment</code> or (2) a named list, with each element corresponding to a matrix representing an omic, with biological entities in rows. Row names should include unique biological entity IDs (e.g., gene symbols, miRNA names); columns represent individuals. If more than one biological entity is used, a <code>gene_map</code> data.frame providing mappings between IDs and gene names should be provided if the default <code>mirtarbase</code> is not sufficient. |
| ...      | Optional additional arguments                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| colData  | (optional) A <code>DataFrame</code> or <code>data.frame</code> of characteristics for all biological units, to be used in creating a <code>MultiAssayExperiment</code> from an object of class <code>list</code>                                                                                                                                                                                                                                                                                                                               |
| gene_map | (optional) Data frame mapping arbitrary biological entities (e.g. miRNAs) to genes. Contains two columns, where the first provides the IDs of the entity and the second provides the IDs of the corresponding target gene. By default, the miRNA-gene interactions of type 'Functional MTI' from <code>miRTarBase</code> are used (see the preloaded 'mirtarbase' data in the package).                                                                                                                                                        |
| base_ids | (optional) Sample names to be used as reference base data. By default, all samples are used.                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| supp_ids | (optional) Sample names to be used as supplementary individuals to be projected onto the analysis based on the individuals identified in <code>base_ids</code> . By default,                                                                                                                                                                                                                                                                                                                                                                   |

|                                 |                                                                                                                                                                                                       |
|---------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                                 | takes the value NULL, but should not overlap with <code>base_ids</code> if provided by the user.                                                                                                      |
| <code>pathway_name</code>       | Character of either a KEGG pathway identifier or MSigDB pathway names (e.g., see the pathway names in the 'geneset' column of the preloaded msigdb data in the package), or a vector of gene symbols. |
| <code>impute</code>             | If TRUE, impute missing values separately in base and supplementary data using MFA as implemented in the <i>missMDA</i> package; otherwise simple mean imputation is used (default).                  |
| <code>variance_threshold</code> | Minimal variance required across samples to retain a biological entity in the analysis                                                                                                                |
| <code>full_results</code>       | If TRUE (default), include full MFA results in function output; otherwise, provide concise output to save space.                                                                                      |
| <code>verbose</code>            | If TRUE, provide verbose output.                                                                                                                                                                      |

### Value

An S4 object of class `padmaResults`, where individualized pathway deviation scores are stored as the assay data, and the corresponding {pathway name, full MFA results, number of genes, and names of imputed or filtered genes} are stored as slots that can be retrieved using the appropriate accessor functions.

### Author(s)

Andrea Rau

### Examples

```

LUAD_subset <- padma::LUAD_subset
## Create MultiAssayExperiment object with LUAD data
omics_data <-
  list(rnaseq = as.matrix(LUAD_subset$rnaseq),
       methyl = as.matrix(LUAD_subset$methyl),
       mirna = as.matrix(LUAD_subset$mirna),
       cna = as.matrix(LUAD_subset$cna))
pheno_data <-
  data.frame(LUAD_subset$clinical,
             row.names = LUAD_subset$clinical$bcr_patient_barcode)
mae <-
  suppressMessages(
    MultiAssayExperiment::MultiAssayExperiment(
      experiments = omics_data, colData = pheno_data))

## Run padma
run_padma <-
  padma(mae, gene_map = padma::mirtarbase,
        pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY", verbose = FALSE)

summary(run_padma)

```

```
## padma plots
## Not run:
factorMap(run_padma, dim_x = 1, dim_y = 2)
factorMap(run_padma, dim_x = 1, dim_y = 2,
          partial_id = "TCGA-78-7536")
omicsContrib(run_padma, max_dim = 10)

## End(Not run)
```

---

padmaResults-class      *padmaResults* object and constructor

---

## Description

`padmaResults` is a subclass of `RangedSummarizedExperiment`, used to store the individualized pathway deviation scores as well as some additional information useful about the pathway name (`pathway_name`), the gene-level contributions to each deviation score (`pathway_gene_deviation`), a full set of outputs related to the MFA (`MFA_results`, and the number of genes used in the analysis as well as the names of those for which data imputation or filtering was required (`ngenes`, `imputed_genes`, and `removed_genes`, respectively).

## Usage

```
padmaResults(
  SummarizedExperiment,
  pathway_name = NULL,
  pathway_gene_deviation = NULL,
  MFA_results = NULL,
  ngenes = NULL,
  imputed_genes = NULL,
  removed_genes = NULL
)
```

## Arguments

|                                     |                                                                                        |
|-------------------------------------|----------------------------------------------------------------------------------------|
| <code>SummarizedExperiment</code>   | a <code>RangedSummarizedExperiment</code> of padma results                             |
| <code>pathway_name</code>           | The name of the pathway, if applicable                                                 |
| <code>pathway_gene_deviation</code> | Per-gene contributions to each individualized pathway deviation score                  |
| <code>MFA_results</code>            | List of all detailed results from the MFA                                              |
| <code>ngenes</code>                 | Number of genes used in the pathway deviation score calculation                        |
| <code>imputed_genes</code>          | Names of genes, per omic, for which data imputation was used to replace missing values |
| <code>removed_genes</code>          | Names of genes, per omic, which were filtered from the analysis due to low variation   |

Details

This constructor function would not typically be used by 'end users'. This simple class extends the RangedSummarizedExperiment class of the SummarizedExperiment package to allow other packages to write methods for results objects from the padma package. It is used by [padmaRun](#) to wrap up the results table.

Value

a padmaResults object

---

|          |                                                                       |
|----------|-----------------------------------------------------------------------|
| padmaRun | <i>Calculate individualized deviation scores from multi-omic data</i> |
|----------|-----------------------------------------------------------------------|

---

Description

Calculate individualized deviation scores from multi-omic data

Usage

```
padmaRun(
  omics_data,
  gene_map = padma::mirtarbase,
  base_ids = NULL,
  supp_ids = NULL,
  pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY",
  impute = FALSE,
  variance_threshold = 1e-04,
  full_results = TRUE,
  verbose = TRUE,
  ...
)
```

Arguments

|            |                                                                                                                                                                                                                                                                                                                                                                            |
|------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| omics_data | Object of class 'MultiAssayExperiment' containing omics data from n matched individuals.                                                                                                                                                                                                                                                                                   |
| gene_map   | (optional) Data frame mapping arbitrary biological entities (e.g. miRNAs) to genes. Contains two columns, where the first provides the IDs of the entity and the second provides the IDs of the corresponding target gene. By default, the miRNA-gene interactions of type 'Functional MTI' from miRTarBase are used (see the preloaded 'mirtarbase' data in the package). |
| base_ids   | (optional) Sample names to be used as reference base data. By default, all samples are used.                                                                                                                                                                                                                                                                               |
| supp_ids   | (optional) Sample names to be used as supplementary individuals to be projected onto the analysis based on the individuals identified in base_ids. By default, takes the value NULL, but should not overlap with base_ids if provided by the user.                                                                                                                         |

|                    |                                                                                                                                                                                                       |
|--------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| pathway_name       | Character of either a KEGG pathway identifier or MSigDB pathway names (e.g., see the pathway names in the 'geneset' column of the preloaded msigdb data in the package), or a vector of gene symbols. |
| impute             | If TRUE, impute missing values separately in base and supplementary data using MFA as implemented in the <i>missMDA</i> package; otherwise simple mean imputation is used (default).                  |
| variance_threshold | Minimal variance required across samples to retain a biological entity in the analysis                                                                                                                |
| full_results       | If TRUE (default), include full MFA results in function output; otherwise, provide concise output to save space.                                                                                      |
| verbose            | If TRUE, provide verbose output.                                                                                                                                                                      |
| ...                | Optional additional arguments                                                                                                                                                                         |

### Value

An S4 object of class `padmaResults`, where individualized pathway deviation scores are stored as the assay data, and the corresponding {pathway name, full MFA results, number of genes, and names of imputed or filtered genes} are stored as slots that can be retrieved using the appropriate accessor functions.

### Examples

```

LUAD_subset <- padma::LUAD_subset
## Create MultiAssayExperiment object with LUAD data
omics_data <-
  list(rnaseq = as.matrix(LUAD_subset$rnaseq),
       methyl = as.matrix(LUAD_subset$methyl),
       mirna = as.matrix(LUAD_subset$mirna),
       cna = as.matrix(LUAD_subset$cna))
pheno_data <-
  data.frame(LUAD_subset$clinical,
             row.names = LUAD_subset$clinical$bcr_patient_barcode)
mae <-
  suppressMessages(
    MultiAssayExperiment::MultiAssayExperiment(
      experiments = omics_data, colData = pheno_data))

## Run padma
run_padma <-
  padma(mae, gene_map = padma::mirtarbase,
        pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY", verbose = FALSE)

summary(run_padma)

## padma plots
## Not run:
factorMap(run_padma, dim_x = 1, dim_y = 2)
factorMap(run_padma, dim_x = 1, dim_y = 2,
          partial_id = "TCGA-78-7536")

```

```
omicsContrib(run_padma, max_dim = 10)

## End(Not run)
```

---

|              |                                             |
|--------------|---------------------------------------------|
| pathway_name | <i>Accessors for a padmaResults object.</i> |
|--------------|---------------------------------------------|

---

## Description

Accessors for a padmaResults object.

## Usage

```
pathway_name(object, ...)

pathway_gene_deviation(object, ...)

MFA_results(object, ...)

ngenes(object, ...)

imputed_genes(object, ...)

removed_genes(object, ...)

## S4 method for signature 'padmaResults'
pathway_name(object)

## S4 method for signature 'padmaResults'
MFA_results(object)

## S4 method for signature 'padmaResults'
ngenes(object)

## S4 method for signature 'padmaResults'
imputed_genes(object)

## S4 method for signature 'padmaResults'
removed_genes(object)

## S4 method for signature 'padmaResults'
pathway_gene_deviation(object)

## S4 method for signature 'padmaResults'
show(object)
```

**Arguments**

object            a padmaResults object

...               Additional optional parameters

**Value**

Output varies depending on the method.

**Author(s)**

Andrea Rau

**Examples**

```

LUAD_subset <- padma::LUAD_subset
## Create MultiAssayExperiment object with LUAD data
omics_data <-
  list(rnaseq = as.matrix(LUAD_subset$rnaseq),
       methyl = as.matrix(LUAD_subset$methyl),
       mirna = as.matrix(LUAD_subset$mirna),
       cna = as.matrix(LUAD_subset$cna))
pheno_data <-
  data.frame(LUAD_subset$clinical,
             row.names = LUAD_subset$clinical$bcr_patient_barcode)
mae <-
  suppressMessages(
    MultiAssayExperiment::MultiAssayExperiment(
      experiments = omics_data, colData = pheno_data))

## Run padma
run_padma <-
  padma(mae, gene_map = padma::mirtarbase,
        pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY", verbose = FALSE)

summary(run_padma)

## padma plots
## Not run:
factorMap(run_padma, dim_x = 1, dim_y = 2)
factorMap(run_padma, dim_x = 1, dim_y = 2,
          partial_id = "TCGA-78-7536")
omicsContrib(run_padma, max_dim = 10)

## End(Not run)

```

---

`summary, padmaResults-method`*Summarize results from padma*

---

## Description

A function to summarize the pathway deviation results from padma, using the quantiles of the calculated multi-omic pathway deviation scores.

## Usage

```
## S4 method for signature 'padmaResults'
summary(object, ...)
```

## Arguments

|                     |                                   |
|---------------------|-----------------------------------|
| <code>object</code> | An object of class 'padmaResults' |
| <code>...</code>    | Additional arguments              |

## Value

Summary of the `padmaResults` object.

## Author(s)

Andrea Rau

## References

Rau, A., Manansala, R., Flister, M. J., Rui, H., Jaffrézic, F., Laloë, D., and Auer, P. L. (2019) Individualized multi-omic pathway deviation scores using multiple factor analysis bioRxiv, <https://doi.org/10.1101/827022>.

## See Also

[padma](#)

## Examples

```
LUAD_subset <- padma::LUAD_subset
## Create MultiAssayExperiment object with LUAD data
omics_data <-
  list(rnaseq = as.matrix(LUAD_subset$rnaseq),
       methyl = as.matrix(LUAD_subset$methyl),
       mirna = as.matrix(LUAD_subset$mirna),
       cna = as.matrix(LUAD_subset$cna))
pheno_data <-
  data.frame(LUAD_subset$clinical,
             row.names = LUAD_subset$clinical$bcr_patient_barcode)
```

```
mae <-
  suppressMessages(
    MultiAssayExperiment::MultiAssayExperiment(
      experiments = omics_data, colData = pheno_data))

## Run padma
run_padma <-
  padma(mae, gene_map = padma::mirtarbase,
        pathway_name = "c2_cp_BIOCARTA_D4GDI_PATHWAY", verbose = FALSE)

summary(run_padma)

## padma plots
## Not run:
factorMap(run_padma, dim_x = 1, dim_y = 2)
factorMap(run_padma, dim_x = 1, dim_y = 2,
          partial_id = "TCGA-78-7536")
omicsContrib(run_padma, max_dim = 10)

## End(Not run)
```

# Index

- \* **datasets**
  - LUAD\_subset, [4](#)
  - mirtarbase, [5](#)
  - msigdb, [5](#)
- \* **methods**
  - padma, [7](#)
  - summary, padmaResults-method, [15](#)
- factorMap, [2](#)
- imputed\_genes (pathway\_name), [13](#)
- imputed\_genes, padmaResults-method (pathway\_name), [13](#)
- LUAD\_subset, [4](#)
- MFA\_results (pathway\_name), [13](#)
- MFA\_results, padmaResults-method (pathway\_name), [13](#)
- mirtarbase, [5](#)
- msigdb, [5](#)
- ngenes (pathway\_name), [13](#)
- ngenes, padmaResults-method (pathway\_name), [13](#)
- omicsContrib, [6](#)
- padma, [7](#), [15](#)
- padma, list-method (padma), [7](#)
- padma, MultiAssayExperiment-method (padma), [7](#)
- padma-methods (padma), [7](#)
- padmaResults (padmaResults-class), [10](#)
- padmaResults-class, [10](#)
- padmaRun, [11](#), [11](#)
- pathway\_gene\_deviation (pathway\_name), [13](#)
- pathway\_gene\_deviation, padmaResults-method (pathway\_name), [13](#)
- pathway\_gene\_deviation-method (pathway\_name), [13](#)
- pathway\_name, [13](#)
- pathway\_name, padmaResults-method (pathway\_name), [13](#)
- removed\_genes (pathway\_name), [13](#)
- removed\_genes, padmaResults-method (pathway\_name), [13](#)
- show (pathway\_name), [13](#)
- show, padmaResults-method (pathway\_name), [13](#)
- summary (summary, padmaResults-method), [15](#)
- summary, padmaResults-method, [15](#)
- summary-methods (summary, padmaResults-method), [15](#)