

Package: MultiOmicsCancer (via r-universe)

October 20, 2026

Title Integrative Multi-Omics Analysis for Pan-Cancer Research

Version 0.99.2

Description A modular and cancer-agnostic framework for integrative analysis of multi-omics data in cancer research. The package provides tools for differential methylation analysis, gene expression analysis, miRNA profiling, multi-omics integration, survival analysis, and biomarker discovery. It supports data from TCGA, cBioPortal, GEO, and user-provided datasets, and is designed to work across any cancer type.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

Depends R (>= 4.5.0)

Imports methods, MultiAssayExperiment, SummarizedExperiment, limma, DESeq2, survival, survminer, ggplot2, rlang, httr2, stats, utils, tidyestimate

Suggests cBioPortalData, curatedTCGADData, testthat (>= 3.0.0), knitr, rmarkdown, BiocCheck, S4Vectors

biocViews Epigenetics, DNAMethylation, GeneExpression, Transcriptomics, DifferentialExpression, DifferentialMethylation, MultipleComparison, Survival

URL <https://github.com/maryam6770/MultiOmicsCancer>

BugReports <https://github.com/maryam6770/MultiOmicsCancer/issues>

Config/testthat/edition 3

Config/roxygen2/version 8.1.0

VignetteBuilder knitr

Config/pak/sysreqs cmake make libicu-dev libjpeg-dev libpng-dev libxml2-dev libssl-dev zlib1g-dev

Repository <https://biocstaging.r-universe.dev>

Date/Publication 2026-10-20 08:08:36 UTC

RemoteUrl <https://github.com/BiocStaging/MultiOmicsCancer>

RemoteRef HEAD
RemoteSha 9cf1d79d0627a0a4add0ac3b52028803a83eafaa

Contents

analyze_expression	2
analyze_methylation	4
analyze_survival	6
bootstrap_validate_ims	7
build_ims	8
compare_ims_clinical	9
estimate_purity	10
integrate_meth_expr	11
load_cancer_data	12
plot_cancer	14
validate_proteins	15

Index	16
--------------	-----------

analyze_expression	<i>Differential Gene Expression Analysis for Pan-Cancer Studies</i>
--------------------	---

Description

Performs differential gene expression analysis between tumor and normal samples (or any two groups defined in the clinical data) for any cancer type. Supports both limma (for log-transformed or normalized data) and DESeq2 (for raw count data).

Usage

```
analyze_expression(  
  mae,  
  assay_name = "expression",  
  group_column = "sample_type",  
  group_levels = NULL,  
  method = "limma",  
  log_transform = TRUE,  
  adj_method = "BH",  
  p_cutoff = 0.05,  
  logFC_cutoff = 1,  
  verbose = TRUE  
)
```

Arguments

<code>mae</code>	A MultiAssayExperiment object created by <code>load_cancer_data</code> .
<code>assay_name</code>	Character. Name of the expression assay in the MultiAssayExperiment. Default "expression".
<code>group_column</code>	Character. Name of the clinical column used to define groups. Default "sample_type".
<code>group_levels</code>	Character vector of length 2. The two groups to compare. The first is the reference (e.g., normal), the second is the comparison (e.g., tumor). If NULL, the function attempts to auto-detect common tumor/normal labels.
<code>method</code>	Character. Method for differential expression. One of "limma" (default) or "DESeq2". Use "DESeq2" only for raw count data.
<code>log_transform</code>	Logical. If TRUE and method = "limma", applies $\log_2(x + 1)$ transformation to the data. Default TRUE.
<code>adj_method</code>	Character. Multiple testing correction method. Default "BH" (Benjamini-Hochberg).
<code>p_cutoff</code>	Numeric. Adjusted p-value threshold for significance. Default 0.05. Lower values increase specificity; higher values increase sensitivity.
<code>logFC_cutoff</code>	Numeric. Minimum absolute log2 fold change to call a gene differentially expressed. Default 1 (i.e., 2-fold). Lower values (e.g., 0.5) increase sensitivity and return more genes, but may include false positives. Higher values (e.g., 1.5) increase specificity and return fewer genes.
<code>verbose</code>	Logical. Whether to print progress messages. Default TRUE.

Value

A data frame with the following columns:

gene Gene identifier

logFC Log2 fold change (group2 - group1)

AveExpr Average expression across all samples

t Moderated t-statistic (limma) or Wald statistic (DESeq2)

P.Value Raw p-value

adj.P.Val Adjusted p-value

direction "Up" or "Down" in group2 vs group1

significant Logical, whether the gene passes both p-value and logFC cutoffs

Examples

```
if (interactive()) {
  # Load data
  brca <- load_cancer_data("brca_tcga")

  # Differential expression: normal vs tumor with limma
  deg <- analyze_expression(brca, method = "limma")

  # With DESeq2 on raw counts
```

```
deg_deseq <- analyze_expression(brca, method = "DESeq2")

# Filter significant genes
sig_deg <- deg[deg$significant, ]

# Lower the logFC cutoff to get more genes (higher sensitivity)
deg_sensitive <- analyze_expression(brca, logFC_cutoff = 0.5)

# Raise the logFC cutoff for stricter results (higher specificity)
deg_strict <- analyze_expression(brca, logFC_cutoff = 1.5)
}
```

analyze_methylation *Differential Methylation Analysis for Pan-Cancer Studies*

Description

Performs differential methylation analysis between tumor and normal samples (or any two groups defined in the clinical data) for any cancer type. Uses the limma framework for linear modeling of methylation beta values.

Usage

```
analyze_methylation(
  mae,
  assay_name = "methylation",
  group_column = "sample_type",
  group_levels = NULL,
  method = "limma",
  adj_method = "BH",
  p_cutoff = 0.05,
  delta_beta = 0.1,
  verbose = TRUE
)
```

Arguments

mae	A MultiAssayExperiment object created by load_cancer_data .
assay_name	Character. Name of the methylation assay in the MultiAssayExperiment. Default "methylation".
group_column	Character. Name of the clinical column used to define groups. Default "sample_type".
group_levels	Character vector of length 2. The two groups to compare. The first is the reference (e.g., normal), the second is the comparison (e.g., tumor). If NULL, the function attempts to auto-detect common tumor/normal labels.
method	Character. Differential methylation method. Currently supports "limma" (default).

analyze_survival	<i>Survival Analysis for Pan-Cancer Studies</i>
------------------	---

Description

Performs survival analysis using clinical data and a molecular feature (gene expression, methylation, or a risk score). Supports Kaplan-Meier curves, log-rank tests, and univariate Cox proportional hazards models.

Usage

```
analyze_survival(
  mae,
  time_column = "OS.time",
  event_column = "OS",
  feature_name = NULL,
  assay_name = "expression",
  split_method = "median",
  verbose = TRUE
)
```

Arguments

mae	A MultiAssayExperiment object created by load_cancer_data , containing clinical survival data.
time_column	Character. Name of the clinical column containing follow-up time (in days or months). Default "OS.time".
event_column	Character. Name of the clinical column containing event status (1 = event/death, 0 = censored). Default "OS".
feature_name	Character. Name of the molecular feature to test (gene symbol or probe ID). If NULL, only clinical summary is returned.
assay_name	Character. Which assay to use for the feature. One of "expression" (default), "methylation", or "mirna".
split_method	Character. How to split samples into high/low groups. One of "median" (default), "tertile", or "quartile".
verbose	Logical. Whether to print progress messages. Default TRUE.

Value

A list with the following components:

km_fit A survfit object for Kaplan-Meier curves
logrank_p P-value from the log-rank test
cox_model A coxph object for univariate Cox regression
cox_summary Summary table of the Cox model

data Data frame used for the analysis

group Factor of high/low groups

Examples

```
if (interactive()) {
# Load data
brca <- load_cancer_data("brca_tcga")

# Survival analysis for TP53 expression
surv <- analyze_survival(brca, feature_name = "TP53")

# Survival analysis for a methylation probe
surv_meth <- analyze_survival(brca,
                             feature_name = "cg00000029",
                             assay_name = "methylation")
}
```

bootstrap_validate_ims

Nested Bootstrap Validation for IMS

Description

Nested Bootstrap Validation for IMS

Usage

```
bootstrap_validate_ims(
  meth_data,
  surv_data,
  immune_genes = NULL,
  B = 100L,
  p_threshold = 0.05,
  min_cpgs = 3L,
  seed = 123
)
```

Arguments

meth_data	Matrix of methylation (CpGs x samples).
surv_data	Data frame with columns 'time' and 'status'.
immune_genes	Character vector of immune gene symbols.
B	Integer. Number of bootstrap replicates. Default 100.
p_threshold	Numeric. P-value cutoff. Default 0.05.
min_cpgs	Integer. Minimum CpGs required. Default 3.
seed	Integer. Random seed. Default 123.

Value

A list with apparent C-index, optimism, corrected C-index, and the bootstrap distribution.

Examples

```
# Source: Mock data simulated for demonstration purposes.
set.seed(123)
meth_data <- matrix(runif(100), nrow = 10, ncol = 10)
rownames(meth_data) <- paste0("CD274_cg", 1:10)
colnames(meth_data) <- paste0("Sample", 1:10)
surv_data <- data.frame(
  time = runif(10, 10, 100),
  status = sample(c(0, 1), 10, replace = TRUE)
)
rownames(surv_data) <- colnames(meth_data)
res <- bootstrap_validate_ims(
  meth_data = meth_data, surv_data = surv_data,
  immune_genes = c("CD274"), B = 5,
  p_threshold = 1, min_cpgs = 2
)
```

 build_ims

Build an Immune Methylation Score (IMS)

Description

Build an Immune Methylation Score (IMS)

Usage

```
build_ims(
  meth_data,
  surv_data,
  immune_genes = NULL,
  p_threshold = 0.05,
  hr_range = c(0.1, 10),
  min_cpgs = 3L
)
```

Arguments

meth_data	A matrix of methylation beta values (CpGs x samples).
surv_data	A data frame with columns 'time' and 'status'.
immune_genes	Character vector of immune gene symbols. If NULL, uses a default 35-gene immune panel.
p_threshold	Numeric. P-value threshold for CpG selection. Default 0.05.
hr_range	Numeric vector length 2. Allowed HR range. Default c(0.1, 10).
min_cpgs	Integer. Minimum CpGs required. Default 3.

Value

A list with IMS, selected_cpgs, coefs, model.

Examples

```
# Source: Mock data simulated for demonstration purposes.
set.seed(123)
meth_data <- matrix(runif(100), nrow = 10, ncol = 10)
rownames(meth_data) <- paste0("CD274_cg", 1:10)
colnames(meth_data) <- paste0("Sample", 1:10)
surv_data <- data.frame(
  time = runif(10, 10, 100),
  status = sample(c(0, 1), 10, replace = TRUE)
)
rownames(surv_data) <- colnames(meth_data)
res <- build_ims(
  meth_data = meth_data, surv_data = surv_data,
  immune_genes = c("CD274"), p_threshold = 1,
  hr_range = c(0, 100), min_cpgs = 2
)
```

compare_ims_clinical *Compare IMS with Clinical Variables*

Description

Compares prognostic performance of IMS vs a clinical variable (e.g., Stage) using Cox models and likelihood ratio test.

Usage

```
compare_ims_clinical(
  ims_values,
  surv_data,
  clinical_var,
  var_name = "Clinical"
)
```

Arguments

ims_values	Named numeric vector of IMS values (from build_ims).
surv_data	Data frame with columns 'time' and 'status'.
clinical_var	Named vector or factor of clinical variable (e.g., Stage_binary) aligned with surv_data rownames.
var_name	Character. Name of clinical variable for output labels.

Value

A list with C-index for IMS alone, clinical alone, both, Delta C-index, and LRT p-value.

Examples

```
# Source: Mock data simulated for demonstration purposes.
set.seed(123)
n <- 60
ims_values <- setNames(rnorm(n), paste0("Sample", 1:n))
surv_data <- data.frame(
  time = runif(n, 10, 100),
  status = sample(c(0, 1), n, replace = TRUE),
  row.names = names(ims_values)
)
clinical_var <- setNames(
  sample(c("Low", "High"), n, replace = TRUE),
  names(ims_values)
)
res <- compare_ims_clinical(
  ims_values = ims_values, surv_data = surv_data,
  clinical_var = clinical_var, var_name = "Stage"
)
```

estimate_purity

Estimate Tumor Purity and Immune Score

Description

Computes ESTIMATE scores (Stromal, Immune, ESTIMATE) and Tumor Purity from gene expression data using the ESTIMATE algorithm. Requires package 'tidyestimate'.

Usage

```
estimate_purity(expr_data, is_affymetrix = FALSE)
```

Arguments

expr_data A matrix of gene expression (genes x samples).

is_affymetrix Logical. TRUE if data is from Affymetrix arrays. Default FALSE (RNA-seq).

Value

A data frame with columns: sample, stromal, immune, estimate, purity.

Examples

```
# Source: Mock data simulated to demonstrate the ESTIMATE algorithm.
set.seed(123)
expr_mat <- matrix(rnorm(1000), nrow = 100, ncol = 10)
rownames(expr_mat) <- paste0("GENE", 1:100)
colnames(expr_mat) <- paste0("Sample", 1:10)
tryCatch({
  res <- estimate_purity(expr_mat)
  head(res)
}, error = function(e) {
  message("estimate_purity requires 'tidyestimate' package.")
})
```

integrate_meth_expr *Integrate Methylation and Gene Expression Data*

Description

Integrates DNA methylation and gene expression data to identify genes whose expression is significantly correlated with their promoter methylation status. This is a key step in multi-omics analysis for discovering epigenetically regulated genes in any cancer type.

Usage

```
integrate_meth_expr(
  mae,
  meth_assay = "methylation",
  expr_assay = "expression",
  gene_map = NULL,
  correlation_method = "pearson",
  min_samples = 5,
  p_cutoff = 0.05,
  cor_cutoff = 0.3,
  verbose = TRUE
)
```

Arguments

mae	A MultiAssayExperiment object created by load_cancer_data , containing both methylation and expression assays.
meth_assay	Character. Name of the methylation assay. Default "methylation".
expr_assay	Character. Name of the expression assay. Default "expression".
gene_map	Optional. A data frame mapping methylation probes to genes. Must have columns probe and gene. If NULL, the function assumes probe and gene identifiers match (i.e., row names of both assays are gene symbols).
correlation_method	Character. Correlation method, one of "pearson" (default) or "spearman".

<code>min_samples</code>	Integer. Minimum number of samples required for correlation analysis. Default 5.
<code>p_cutoff</code>	Numeric. Adjusted p-value threshold for significance. Default 0.05.
<code>cor_cutoff</code>	Numeric. Minimum absolute correlation coefficient to consider a methylation-expression pair significant. Default 0.3.
<code>verbose</code>	Logical. Whether to print progress messages. Default TRUE.

Value

A data frame with the following columns:

gene Gene identifier

probe Methylation probe identifier

correlation Correlation coefficient between methylation and expression

p_value Raw p-value of the correlation test

adj_p_value Adjusted p-value (Benjamini-Hochberg)

n_samples Number of samples used in the correlation

regulation "Negative" if methylation is inversely correlated with expression (expected for promoter methylation), "Positive" otherwise

significant Logical, whether the pair passes both p-value and correlation cutoffs

Examples

```
if (interactive()) {
# Load multi-omics data
brca <- load_cancer_data("brca_tcga",
                        data_types = c("methylation", "expression"))

# Integrate methylation and expression
integrated <- integrate_meth_expr(brca)

# Filter significant negative correlations
epi_genes <- integrated[integrated$significant &
                        integrated$regulation == "Negative", ]
}
```

load_cancer_data

Load Multi-Omics Cancer Data

Description

Loads multi-omics data for any cancer type from cBioPortal, TCGA, or user-provided datasets. The function returns a `MultiAssayExperiment` object that can be used by all downstream analysis functions in the package.

Usage

```
load_cancer_data(
  cancer_type,
  data_types = c("methylation", "expression", "mirna"),
  source = "cBioPortal",
  genes = NULL,
  custom_data = NULL,
  verbose = TRUE
)
```

Arguments

cancer_type	Character. Cancer study identifier. For cBioPortal, use study IDs such as "brca_tcga", "luad_tcga", "coadread_tcga". For a full list, see <code>cBioPortalData::getStudies()</code> .
data_types	Character vector. Omics layers to load. Options include "methylation", "expression", and "mirna". Default loads all three.
source	Character. Data source. One of "cBioPortal" (default), "custom", or "TCGA".
genes	Optional character vector of gene symbols to restrict the download (e.g., <code>c("TP53", "BRCA1", "EGFR")</code>). Only used when <code>source = "cBioPortal"</code> . If NULL (default), all available genes are downloaded, which can be very slow for large studies.
custom_data	Optional. A named list of matrices for custom data. Names must match <code>data_types</code> . Only used when <code>source = "custom"</code> .
verbose	Logical. Whether to print progress messages. Default TRUE.

Value

A `MultiAssayExperiment` object containing the requested omics layers and associated clinical data.

Examples

```
if (interactive()) {
  # Load breast cancer data from cBioPortal (all genes)
  brca <- load_cancer_data("brca_tcga")

  # Load only selected genes (faster)
  brca <- load_cancer_data(
    "brca_tcga",
    data_types = c("methylation", "expression"),
    genes = c("TP53", "BRCA1", "BRCA2", "EGFR")
  )

  # Load custom data
  custom <- load_cancer_data("my_study",
    source = "custom",
    custom_data = list(methylation = meth_matrix,
      expression = expr_matrix))
}
```

```
}
```

plot_cancer

Visualization Functions for Pan-Cancer Multi-Omics Analysis

Description

A collection of publication-quality plotting functions for visualizing results from differential methylation, differential expression, multi-omics integration, and survival analysis. All plots are based on ggplot2 and return ggplot objects that can be further customized.

Usage

```
plot_cancer(
  data = NULL,
  plot_type = c("volcano", "heatmap", "correlation", "survival"),
  title = NULL,
  p_cutoff = 0.05,
  fc_cutoff = NULL,
  color_up = "firebrick",
  color_down = "steelblue",
  color_ns = "grey70",
  max_labels = 10,
  surv_result = NULL,
  verbose = TRUE
)
```

Arguments

data	A data frame returned by analyze_methylation , analyze_expression , or integrate_meth_expr .
plot_type	Character. Type of plot to generate. One of "volcano", "heatmap", "correlation", or "survival".
title	Optional. Plot title. If NULL, a default title is used.
p_cutoff	Numeric. Adjusted p-value threshold for highlighting significant features. Default 0.05.
fc_cutoff	Numeric. Log fold change or delta-beta threshold for highlighting significant features. Default 1 for expression and 0.1 for methylation. If NULL, uses the significant column in the data.
color_up	Character. Color for upregulated/hypermethylated features. Default "firebrick".
color_down	Character. Color for downregulated/hypomethylated features. Default "steelblue".
color_ns	Character. Color for non-significant features. Default "grey70".
max_labels	Integer. Maximum number of labels to display on volcano plots. Default 10.
surv_result	Optional. A list returned by analyze_survival for survival plots.
verbose	Logical. Whether to print progress messages. Default TRUE.

Value

A ggplot object.

Examples

```
if (interactive()) {  
  # Volcano plot of differential expression  
  deg <- analyze_expression(brca)  
  plot_cancer(deg, plot_type = "volcano")  
  
  # Volcano plot of differential methylation  
  dmr <- analyze_methylation(brca)  
  plot_cancer(dmr, plot_type = "volcano")  
  
  # Survival plot  
  surv <- analyze_survival(brca, feature_name = "TP53")  
  plot_cancer(plot_type = "survival", surv_result = surv)  
}
```

validate_proteins	<i>Validate Genes at Protein Level</i>
-------------------	--

Description

Validates genes using UniProt (protein existence).

Usage

```
validate_proteins(genes, use_uniprot = TRUE, timeout = 30)
```

Arguments

genes	Character vector of gene symbols.
use_uniprot	Logical. Query UniProt? Default TRUE.
timeout	Numeric. HTTP timeout in seconds. Default 30.

Value

A data frame with columns: gene, protein_exists, protein_name.

Examples

```
# Source: Mock data for offline demonstration.  
genes <- c("BRCA1", "TP53", "CD274")  
res <- validate_proteins(genes, use_uniprot = FALSE)  
head(res)
```

Index

analyze_expression, [2](#), [14](#)
analyze_methylation, [4](#), [14](#)
analyze_survival, [6](#), [14](#)

bootstrap_validate_ims, [7](#)
build_ims, [8](#)

compare_ims_clinical, [9](#)

estimate_purity, [10](#)

integrate_meth_expr, [11](#), [14](#)

load_cancer_data, [3](#), [4](#), [6](#), [11](#), [12](#)

plot_cancer, [14](#)

validate_proteins, [15](#)