

Package: MacrophageFramework (via r-universe)

October 26, 2026

Type Package

Title Multidimensional Macrophage Profiling

Version 0.99.0

Description Provides a multidimensional computational framework for macrophage profiling based on an ontology of biological programs. The framework independently detects macrophage populations and characterizes their lineage-associated and functional-state programs using program-level expression scores, detection coverage, gene-program coherence, cluster profiles, and profile similarity. The framework is designed to capture continuous and combinatorial macrophage states without imposing binary activation categories.

License GPL-3

URL <https://github.com/AlessandroPalma-hub/MacrophageFramework>

BugReports <https://github.com/AlessandroPalma-hub/MacrophageFramework/issues>

Encoding UTF-8

Depends R (>= 4.5.0)

Imports ggplot2, Matrix, patchwork, Seurat, SeuratObject

Suggests testthat (>= 3.0.0), knitr, rmarkdown

VignetteBuilder knitr

Config/testthat/edition 3

biocViews GeneExpression, ImmunoOncology, SingleCell, Software, Visualization

RoxygenNote 8.0.0

Config/pak/sysreqs cmake libglpk-dev make libicu-dev libpng-dev libuv1-dev libxml2-dev libssl-dev python3 zlib1g-dev

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

Date/Publication 2026-10-26 15:15:27 UTC

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

RemoteRef HEAD

RemoteSha 34082803510846696ed9fb860eee3dd83ccdd629

Contents

analyze_macrophage_profile	2
calculate_detection_coverage	4
calculate_profile_similarity	5
calculate_program_coherence	6
load_ontology	8
plot_macrophage_profile	9
plot_macrophage_similarity	10
plot_macrophage_umap	11
score_programs	13
summarize_cluster_coherence	14
summarize_macrophage_profile	15

Index	17
--------------	-----------

analyze_macrophage_profile

Analyze a macrophage expression profile

Description

Runs the independent macrophage detector and applies the Macrophage Framework to clusters passing the detector gate. The detector establishes macrophage identity independently from the Framework ontology, whereas the Framework characterizes lineage-associated and functional-state programs.

Usage

```
analyze_macrophage_profile(
  expression = NULL,
  clusters = NULL,
  ontology = NULL,
  object = NULL,
  assay = "RNA",
  layer = NULL,
  reduction = NULL,
  detect_macrophages = TRUE,
  macrophage_clusters = NULL,
  min_core_score = 0.1,
  species = "Human",
  output_dir = "MacrophageFramework_output",
  save_outputs = TRUE
)
```

Arguments

expression	A gene-by-cell expression matrix. Ignored when object is supplied.
clusters	A vector assigning each cell to a cluster. Ignored when object is supplied.
ontology	Optional Framework ontology. If NULL, the default ontology is loaded with load_ontology .
object	Optional Seurat object. If supplied, expression data and cluster identities are extracted automatically.
assay	Assay to use when object is a Seurat object.
layer	Expression layer to use when object is a Seurat object. If NULL, available count or data layers are selected automatically.
reduction	Reduction used to obtain the embedding from a Seurat object. If NULL, an available embedding is selected automatically.
detect_macrophages	Logical indicating whether the independent macrophage detector should be run.
macrophage_clusters	Optional vector specifying clusters to analyze directly. If NULL, clusters are selected using the macrophage detector.
min_core_score	Minimum cluster-level macrophage-core score required for inclusion when macrophage clusters are selected by the detector.
species	Species used for ontology gene mapping. Supported values include "Human" and "Mouse".
output_dir	Directory in which analysis outputs are saved when save_outputs = TRUE.
save_outputs	Logical indicating whether analysis results and visualization outputs should be written to output_dir.

Details

The function accepts either a gene-by-cell expression matrix together with cluster assignments or a Seurat object. For Seurat objects, expression data and cluster identities are extracted automatically. The Framework ontology is version 0.4 by default.

Value

A list containing macrophage detection results, Framework profiles, similarity, hierarchy, program scores, coverage and coherence diagnostics, ontology, annotations, embedding information, and the analyzed cells and clusters. When save_outputs = TRUE, the list also contains paths to the generated output files.

Examples

```
ontology <- load_ontology("v0.4")

genes <- unique(
  ontology$Gene[
    !is.na(ontology$Gene) & nzchar(ontology$Gene)
  ]
)
```

```

)

set.seed(1)
expression <- matrix(
  sample(1:5, length(genes) * 6, replace = TRUE),
  nrow = length(genes),
  dimnames = list(
    genes,
    paste0("Cell", seq_len(6))
  )
)

clusters <- rep(c("Cluster1", "Cluster2"), each = 3)

result <- analyze_macrophage_profile(
  expression = expression,
  clusters = clusters,
  ontology = ontology,
  detect_macrophages = FALSE,
  macrophage_clusters = unique(clusters),
  species = "Human",
  save_outputs = FALSE
)

head(result$analyzed_cells)

```

calculate_detection_coverage

Calculate program detection coverage

Description

Calculates the fraction of genes belonging to each Macrophage Framework program that are detected in each cell. A gene is considered detected when its expression value is greater than zero.

Usage

```
calculate_detection_coverage(expression, ontology = NULL)
```

Arguments

expression	A gene-by-cell expression matrix. Dense matrices and sparse matrices from the Matrix package are supported. Gene names must be provided as row names.
ontology	Optional Framework ontology data frame. If NULL, the default ontology is loaded with load_ontology .

Details

Coverage is calculated independently for each program and cell using only ontology genes that are present in the supplied expression matrix. It therefore describes the extent to which the genes defining a program are detected in a cell and should be interpreted as a diagnostic measure rather than as a probability, confidence score, or measure of program activity.

Value

A program-by-cell numeric matrix containing the proportion of available ontology genes detected in each cell. Rows correspond to Framework programs and columns correspond to cells.

Examples

```
ontology <- load_ontology()

set.seed(1)
genes <- unique(ontology$Gene)
expression <- matrix(
  rpois(length(genes) * 5, lambda = 1),
  nrow = length(genes),
  dimnames = list(
    genes,
    paste0("Cell", seq_len(5))
  )
)

coverage <- calculate_detection_coverage(
  expression = expression,
  ontology = ontology
)

head(coverage)
```

calculate_profile_similarity

Calculate similarity between macrophage profiles

Description

Calculates the cosine similarity between two Macrophage Framework profiles based on their program scores. Only biological programs shared by both profiles and with non-missing scores in both profiles are used.

Usage

```
calculate_profile_similarity(profile_1, profile_2)
```

Arguments

- profile_1 A Macrophage Framework profile list containing a scores matrix.
- profile_2 A second Macrophage Framework profile list containing a scores matrix.

Details

Cosine similarity measures the similarity in the overall pattern of program scores independently of the absolute magnitude of the two profiles. Values closer to 1 indicate more similar program profiles, whereas values closer to 0 indicate less similar profiles.

Value

A numeric value representing the cosine similarity between the two profiles. NA is returned when fewer than two programs can be compared or when one of the profiles has zero magnitude.

Examples

```
profile_1 <- list(
  scores = matrix(
    c(1, 0.5, 0.2),
    ncol = 1,
    dimnames = list(
      c("Program_A", "Program_B", "Program_C"),
      "Profile_1"
    )
  )
)

profile_2 <- list(
  scores = matrix(
    c(0.8, 0.6, 0.1),
    ncol = 1,
    dimnames = list(
      c("Program_A", "Program_B", "Program_C"),
      "Profile_2"
    )
  )
)

calculate_profile_similarity(profile_1, profile_2)
```

calculate_program_coherence

Calculate program coherence

Description

Quantifies the internal coherence of each Macrophage Framework program across individual cells. Coherence is calculated from pairwise similarity among the genes belonging to each program that are detected in a given cell.

Usage

```
calculate_program_coherence(expression, ontology = NULL, scale_factor = 10000)
```

Arguments

expression	A gene-by-cell expression matrix. Dense matrices and sparse matrices from the Matrix package are supported. Gene names must be provided as row names.
ontology	Optional Framework ontology data frame. If NULL, the default ontology is loaded with load_ontology . The ontology must contain Program, Gene, and Weight columns.
scale_factor	Numeric scaling factor used for library-size normalization. Defaults to 10000.

Details

Expression values are normalized using transcriptome-wide library size, scaled by `scale_factor`, and log-transformed before coherence is calculated. Genes with zero expression in a cell are treated as unobserved and are excluded from the pairwise calculation.

For each program and cell, pairwise relative expression differences are calculated among observed program genes and combined using the product of their ontology weights. The resulting coherence score is bounded between 0 and 1, with higher values indicating greater internal agreement among the observed genes of a program.

Coherence is a diagnostic measure of internal program consistency and should not be interpreted as a probability, confidence score, or direct measure of program activity.

Value

A program-by-cell numeric matrix containing coherence scores. Rows correspond to Framework programs and columns correspond to cells. Values range from 0 to 1. NA values indicate that fewer than two program genes were detected in the corresponding cell.

Examples

```
ontology <- load_ontology()

set.seed(1)
genes <- unique(ontology$Gene)
expression <- matrix(
  rpois(length(genes) * 5, lambda = 1),
  nrow = length(genes),
  dimnames = list(
    genes,
    paste0("Cell", seq_len(5))
  )
)
```

```

)

coherence <- calculate_program_coherence(
  expression = expression,
  ontology = ontology
)

head(coherence)

```

load_ontology	<i>Load the macrophage ontology</i>
---------------	-------------------------------------

Description

Loads a predefined version of the Macrophage Framework ontology from the package's installed inst/extdata directory.

Usage

```
load_ontology(version = "v0.4")
```

Arguments

version	Character string specifying the ontology version to load. Defaults to "v0.4".
---------	---

Details

The current ontology version is "v0.4". In this version, macrophage identity is established independently by the macrophage detector and is therefore not represented as an ontology dimension. The ontology defines lineage-associated and functional-state programs used for Framework scoring and annotation.

Value

A data frame containing the selected Macrophage Framework ontology.

Examples

```
ontology <- load_ontology()
head(ontology)
```

`plot_macrophage_profile`*Plot multidimensional macrophage profiles*

Description

Visualizes the relative intensity of Macrophage Framework programs across analyzed macrophage clusters. Program scores are normalized independently for each cluster to unit Euclidean norm for visualization, allowing the relative composition of multidimensional profiles to be compared.

Usage

```
plot_macrophage_profile(result)
```

Arguments

`result` Output list returned by [analyze_macrophage_profile](#).

Details

This normalization is applied only for visualization and does not modify the original program scores stored in the analysis result.

Value

A ggplot object showing normalized program-intensity profiles for the analyzed clusters.

Examples

```
ontology <- load_ontology("v0.4")

genes <- unique(
  ontology$Gene[
    !is.na(ontology$Gene) & nzchar(ontology$Gene)
  ]
)

set.seed(1)
expression <- matrix(
  sample(1:5, length(genes) * 6, replace = TRUE),
  nrow = length(genes),
  dimnames = list(
    genes,
    paste0("Cell", seq_len(6))
  )
)

clusters <- rep(c("Cluster1", "Cluster2"), each = 3)
```

```

result <- analyze_macrophage_profile(
  expression = expression,
  clusters = clusters,
  ontology = ontology,
  detect_macrophages = FALSE,
  macrophage_clusters = unique(clusters),
  species = "Human",
  save_outputs = FALSE
)

plot_macrophage_profile(result)

```

plot_macrophage_similarity

Plot macrophage profile similarity

Description

Visualizes pairwise similarity between macrophage profiles as a heatmap. Similarity values are obtained from the profile similarity matrix generated by [analyze_macrophage_profile](#) and represent cosine similarity between multidimensional program-score profiles.

Usage

```
plot_macrophage_similarity(result)
```

Arguments

`result` Output list returned by [analyze_macrophage_profile](#).

Details

Higher values indicate greater similarity between profiles. The diagonal represents the similarity of each profile with itself.

Value

A ggplot object containing a heatmap of pairwise macrophage profile similarity.

Examples

```

ontology <- load_ontology("v0.4")

genes <- unique(
  ontology$Gene[
    !is.na(ontology$Gene) & nzchar(ontology$Gene)
  ]
)

```

```

set.seed(1)
expression <- matrix(
  sample(1:5, length(genes) * 6, replace = TRUE),
  nrow = length(genes),
  dimnames = list(
    genes,
    paste0("Cell", seq_len(6))
  )
)

clusters <- rep(c("Cluster1", "Cluster2"), each = 3)

result <- analyze_macrophage_profile(
  expression = expression,
  clusters = clusters,
  ontology = ontology,
  detect_macrophages = FALSE,
  macrophage_clusters = unique(clusters),
  species = "Human",
  save_outputs = FALSE
)

plot_macrophage_similarity(result)

```

plot_macrophage_umap *Plot macrophage Framework annotations on a dimensional reduction*

Description

Visualizes Macrophage Framework profile annotations on a dimensional reduction stored in a Seurat object. The function produces two plots: the original computational cluster assignments and the corresponding multidimensional Macrophage Framework profile annotations.

Usage

```
plot_macrophage_umap(result, object, clusters, cell_names, reduction = "umap")
```

Arguments

result	Output list returned by analyze_macrophage_profile .
object	Seurat object containing the dimensional reduction to plot.
clusters	A vector of cell-level cluster assignments corresponding to cell_names.
cell_names	Character vector of cell barcodes corresponding to the elements of clusters.
reduction	Name of the Seurat dimensional reduction to use. Defaults to "umap".

Details

Cells are matched between the dimensional reduction and the supplied cluster assignments using their cell barcodes. Framework profile labels are obtained from [summarize_macrophage_profile](#).

Although the function is named `plot_macrophage_umap`, any Seurat dimensional reduction containing at least two dimensions can be used.

Value

A patchwork object containing two ggplot panels: the original computational clusters and the corresponding Macrophage Framework profile annotations.

Examples

```
ontology <- load_ontology("v0.4")

genes <- unique(
  ontology$Gene[
    !is.na(ontology$Gene) & nzchar(ontology$Gene)
  ]
)

set.seed(1)
cells <- paste0("Cell", seq_len(6))

expression <- matrix(
  sample(1:5, length(genes) * length(cells), replace = TRUE),
  nrow = length(genes),
  dimnames = list(genes, cells)
)

clusters <- rep(c("Cluster1", "Cluster2"), each = 3)

result <- analyze_macrophage_profile(
  expression = expression,
  clusters = clusters,
  ontology = ontology,
  detect_macrophages = FALSE,
  macrophage_clusters = unique(clusters),
  species = "Human",
  save_outputs = FALSE
)

seurat_object <- SeuratObject::CreateSeuratObject(
  counts = expression
)

embeddings <- matrix(
  rnorm(length(cells) * 2),
  nrow = length(cells),
  dimnames = list(cells, c("UMAP_1", "UMAP_2"))
)
```

```

seurat_object[["umap"]] <- SeuratObject::CreateDimReducObject(
  embeddings = embeddings,
  key = "UMAP_",
  assay = "RNA"
)

plot_macrophage_umap(
  result = result,
  object = seurat_object,
  clusters = clusters,
  cell_names = cells,
  reduction = "umap"
)

```

score_programs

Score macrophage biological programs

Description

Calculates weighted expression scores for the biological programs defined in the Macrophage Framework ontology. Expression values are normalized using the transcriptome-wide library size of each cell, scaled by `scale_factor`, and log-transformed before program scores are calculated.

Usage

```

score_programs(
  expression,
  ontology = NULL,
  scale_factor = 10000,
  library_size = NULL
)

```

Arguments

<code>expression</code>	A gene-by-cell expression matrix. Dense matrices and sparse matrices from the Matrix package are supported. Gene names must be provided as row names.
<code>ontology</code>	Optional Framework ontology data frame. If <code>NULL</code> , the default ontology is loaded with load_ontology .
<code>scale_factor</code>	Numeric scaling factor used for library-size normalization. Defaults to 10000.
<code>library_size</code>	Optional vector containing the library size for each cell. If <code>NULL</code> , library sizes are calculated as the column sums of expression.

Details

Only ontology genes detected in the supplied expression matrix are used for scoring. Program scores are calculated as the weighted mean of the log-transformed expression values of the genes belonging to each program. Gene weights are defined by the `Weight` column of the ontology.

Program coverage is calculated as the proportion of ontology genes available in the expression matrix for each program. Coverage therefore describes gene availability and is not used as a probability or confidence measure.

Value

A list containing:

scores A program-by-cell matrix of weighted program scores.

genes_available The number of ontology genes available in the expression matrix for each program.

genes_total The total number of genes defined for each program in the ontology.

coverage The proportion of ontology genes available for each program.

Examples

```
ontology <- load_ontology()

set.seed(1)
expression <- matrix(
  rpois(100, lambda = 2),
  nrow = 10,
  dimnames = list(
    ontology$Gene[seq_len(10)],
    paste0("Cell", seq_len(10))
  )
)

result <- score_programs(
  expression = expression,
  ontology = ontology
)
```

summarize_cluster_coherence

Summarize program coherence by cluster

Description

Calculates the median and interquartile range (IQR) of each program's cell-level coherence within each cluster. Non-finite values are excluded from the calculations.

Usage

```
summarize_cluster_coherence(cell_coherence, cluster_labels)
```

Arguments

cell_coherence Numeric matrix of program-by-cell coherence values, with programs in rows and cells in columns.

cluster_labels Vector assigning one cluster label to each cell, in the same order as the columns of cell_coherence.

Value

A list containing two matrices, median and IQR. Both matrices have programs in rows and clusters in columns. Entries remain NA when a program-cluster combination has no finite coherence values.

```
summarize_macrophage_profile
```

Summarize macrophage profiles in a human-readable table

Description

Converts the results of a Macrophage Framework analysis into a data frame containing cluster-level identity, lineage, functional state, and other profile information.

Usage

```
summarize_macrophage_profile(result, clusters = NULL)
```

Arguments

result Output from `analyze_macrophage_profile()`.

clusters Optional vector of cell-level cluster assignments, used to calculate the number of cells in each cluster.

Value

A data.frame summarizing the multidimensional macrophage profiles, including cluster identity, lineage, functional state, profile labels, and program-level scores and diagnostics.

Examples

```
ontology <- load_ontology("v0.4")

genes <- unique(
  ontology$Gene[
    !is.na(ontology$Gene) & nzchar(ontology$Gene)
  ]
)
```

```
)

set.seed(1)
expression <- matrix(
  sample(1:5, length(genes) * 6, replace = TRUE),
  nrow = length(genes),
  dimnames = list(
    genes,
    paste0("Cell", seq_len(6))
  )
)

clusters <- rep(c("Cluster1", "Cluster2"), each = 3)

result <- analyze_macrophage_profile(
  expression = expression,
  clusters = clusters,
  ontology = ontology,
  detect_macrophages = FALSE,
  macrophage_clusters = unique(clusters),
  species = "Human",
  save_outputs = FALSE
)

summary <- summarize_macrophage_profile(
  result = result,
  clusters = clusters
)

head(summary)
```

Index

`analyze_macrophage_profile`, [2](#), [9–11](#)

`calculate_detection_coverage`, [4](#)
`calculate_profile_similarity`, [5](#)
`calculate_program_coherence`, [6](#)

`load_ontology`, [3](#), [4](#), [7](#), [8](#), [13](#)

`plot_macrophage_profile`, [9](#)
`plot_macrophage_similarity`, [10](#)
`plot_macrophage_umap`, [11](#)

`score_programs`, [13](#)
`summarize_cluster_coherence`, [14](#)
`summarize_macrophage_profile`, [12](#), [15](#)