Package ‘EMMA’

June 23, 2026

Title EMMA: Enrichment Methods Matter for enabling fully reproducible and provenance-aware pathway analysis

Version 0.99.4

Description EMMA is a package that provides a provenance-aware execution framework for functional enrichment analyses. It allows users to run enrichment analyses using existing tools and workflows while explicitly capturing analytical parameters during runtime, and returns the native enrichment results output together with structured metadata.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

RoxygenNote 8.0.0

Imports AnnotationDbi, cli

Suggests knitr, rmarkdown, BiocStyle, macrophage, DESeq2, apeglm, clusterProfiler, gprofiler2, org.Hs.eg.db, GO.db, mosdef, topGO, renv, DeeDeeExperiment, testthat (>= 3.0.0)

VignetteBuilder knitr

URL `https://github.com/imbeimainz/EMMA`

BugReports `https://github.com/imbeimainz/EMMA/issues`

biocViews Software, Pathways, GO, KEGG, GeneSetEnrichment, ImmunoOncology, Transcriptomics, SingleCell, GeneExpression, DifferentialExpression

Config/testthat/edition 3

git_url https://git.bioconductor.org/packages/EMMA

git_branch devel

git_last_commit 7ac847e

git_last_commit_date 2026-06-10

Repository Bioconductor 3.24

Date/Publication 2026-06-23

Author Najla Abassi [aut, cre] (ORCID: <`https://orcid.org/0000-0001-8357-0938`>), Annekathrin Nedwed [aut] (ORCID: <`https://orcid.org/0000-0002-2475-4945`>), Federico Marini [aut] (ORCID: <`https://orcid.org/0000-0003-3252-7758`>)

Maintainer Najla Abassi <`abassi.nejla96@gmail.com`>

Contents

`de_res_IFNg_vs_naive` *A sample* `data.frame` *containing Differential Expression Analysis, generated with* `DESeq2`

Description

A sample `data.frame` containing Differential Expression Analysis, generated with `DESeq2`

Format

A `data.frame` object

Details

This `data.frame` object contains the results of a Differential Expression Analysis performed on data from the `macrophage` package, more precisely contrasting the counts from naive macrophage to those associated with IFNg.

The code to create said object can be found in the folder `/inst/scripts` in the EMMA package, the file is called `create_datasets_examples.R`.

Value

A sample `data.frame` object, extracted from `DESeq2` results

References

Alasoo, et al. "Shared genetic effects on chromatin and gene expression indicate a role for enhancer priming in immune response", Nature Genetics, January 2018 doi: 10.1038/s41588-018-0046-7.

`EMMA-pkg` *EMMA*

Description

EMMA stands for Enrichment Methods Matter. And EMMA stands to help you in realizing it.

Author(s)

Maintainer: Najla Abassi <`abassi.nejla96@gmail.com`> (ORCID)

Authors:

- Najla Abassi <`abassi.nejla96@gmail.com`> (ORCID)
- Annekathrin Nedwed <`anneludt@uni-mainz.de`> (ORCID)
- Federico Marini <`marinif@uni-mainz.de`> (ORCID)

See Also

Useful links:

- `https://github.com/imbeimainz/EMMA`
- Report bugs at `https://github.com/imbeimainz/EMMA/issues`

`EMMA_add_custom_metadata` *EMMA_add_custom_metadata*

Description

Append or replace the `extra` field in the `EMMA_record` attribute of a result object returned by `EMMA_run()`. This allows users to manually provide additional annotation or contextual information that could not be captured automatically

Usage

```
EMMA_add_custom_metadata(res, extra = list())
```

Arguments

`res`	A functional enrichment analysis results object as returned by `EMMA_run()`
`extra`	A named list of user-defined metadata elements to store in the `extra` field

Value

The input result object with updated `EMMA_record` attribute

Examples

```
data("fea_res", package = "EMMA")
fea_res <- EMMA_add_custom_metadata(fea_res, extra =
  list(note = "The background gene set list was all expressed genes in the assay"))
```

`EMMA_explain` *EMMA_explain*

Description

This function generates a human-readable description of the FEA, similar to a Materials and Methods section of a paper, by summarizing the executed call, the parameters, software context, and reference databases used.

Usage

```
EMMA_explain(res, get_citation = TRUE)
```

Arguments

`res`	A functional enrichment analysis results object as returned by `EMMA_run()`. Its attributes contain `EMMA_record`, which contains all provenance information of the performed FEA
`get_citation`	Logical indicating whether to display the citations of the packages used in the FEA. It only prints the citations in an interactive session (e.g console). Defaults to `TRUE`

Value

A character string describing how the FEA was performed using the recorded metadata

Examples

```
data("fea_res", package = "EMMA")
EMMA_explain(fea_res)
```

`EMMA_freeze` *EMMA_freeze*

Description

This function records the R environment during analysis runtime and generates a lockfile that can be used with tools such as `renv`. By default, all currently loaded namespaces are recorded.

Usage

```
EMMA_freeze(
  project = getwd(),
  file = "renv.lock",
  pkgs = loadedNamespaces(),
  prompt = interactive(),
  force = TRUE
)
```

Arguments

`project`	Character string corresponding to the path to the project directory where the lockfile should be written. If the directory does not exist, it will be created. It defaults to the current working directory
`file`	Character string referring to the name of the lockfile to generate. It defaults to "renv.lock"
`pkgs`	Character vector of package names to snapshot. It defaults to all currently loaded namespaces via `loadedNamespaces()`
`prompt`	Logical indicating whether to prompt before taking actions. Defaults to `interactive()`
`force`	Logical indicating whether to force creation of the lockfile. Defaults to `TRUE`

Details

This function calls `renv::snapshot()` with the specified packages. The resulting lockfile can later be restored with `renv::restore()` to recreate the same package environment.

By default, the lockfile is created with `force` = `TRUE`, allowing snapshot creation even if inconsistencies are detected in the environment.

Value

Invisibly returns the path to the generated lockfile. The lockfile is written in JSON format and can be used with `renv::restore()` to recreate the package environment

See Also

`snapshot`, `restore`

Examples

```
# create a lockfile
if (requireNamespace("renv", quietly = TRUE)) {
tmp <- tempfile("emma_env")
dir.create(tmp)

EMMA_freeze(project = tmp)

# inspect generated files
list.files(tmp)

# to restore the environment later
# renv::restore(project = tmp)
}
```

`EMMA_get_record` *EMMA_get_record*

Description

EMMA_get_record

Usage

```
EMMA_get_record(res)
```

Arguments

`res`	Functional Enrichment Analysis results (enrichResult, gseaResult ...) generated by `EMMA_run()`

Value

list of metadata recorded during FEA runtime

See Also

`enrichGO()`, `groupGO()`, `gseGO()`, `gseKEGG()`, `enrichKEGG()`, `gost()`, `run_cluPro()`, `run_topGO()`, `run_goseq()`

Examples

```
data("fea_res", package = "EMMA")
EMMA_get_record(fea_res)
```

`EMMA_run` *EMMA_run*

Description

This function executes any functional enrichment analysis function and attaches a provenance record (`EMMA_record`) describing the analysis. The captured record includes the original call, metadata derived from the call and its arguments, runtime information, and optionally the current session information.

Usage

```
EMMA_run(
  expr,
  envir = parent.frame(),
  store_session_info = TRUE,
  args_form = c("evaluated", "unevaluated")
)
```

Arguments

`expr`	A function call that performs functional enrichment analysis. The call is captured and executed by EMMA to record analysis parameters and provenance information. Both bare calls (`enrichGO(...)`) and namespace-qualified calls (`clusterProfiler::enrichGO(...)`) are supported.
`envir`	An environment in which to evaluate `expr`
`store_session_info`	Logical, indicating whether to store the output of `sessionInfo()` or not. If `TRUE` (default), the session is stored in the provenance record
`args_form`	A character string indicating whether to store the evaluated or the unevaluated arguments in the provenance record. It default to `"evaluated"`

Details

`EMMA_run()` accepts both direct calls to known enrichment functions (`enrichGO()`, `GSEA()`, `fgsea()` ...) and calls to wrapper functions that internally invoke a know enrichment function.

Value

The result object returned by the enrichment function in `expr`, in standard format, with an additional `EMMA_record` attribute containing the provenance information. Use `EMMA_get_record()` to retrieve this record

Examples

```
data("de_res_IFNg_vs_naive", package = "EMMA")
data("gene_universe", package = "EMMA")
library(gprofiler2)

EMMA_run(gost(query = de_res_IFNg_vs_naive$SYMBOL, organism = "hsapiens",
  correction_method = "fdr", custom_bg = gene_universe, sources = "GO:BP"),
  store_session_info = FALSE, args_form = "unevaluated")
```

`EMMA_show` *EMMA_show*

Description

This function displays a human-readable summary of the `EMMA_record` attribute attached to a result object produced by `EMMA_run()`

Usage

```
EMMA_show(res)
```

Arguments

`res`	A functional enrichment analysis results object as returned by `EMMA_run()`

Value

`base::invisible()`

Examples

```
data("fea_res", package = "EMMA")
EMMA_show(fea_res)
```

`fea_res` *A sample* `list` *containing Functional Enrichment Analysis results, generated with* `gprofiler2`

Description

A sample `list` containing Functional Enrichment Analysis results, generated with `gprofiler2`

Format

A `list`

Details

This `list` object contains the result table and metadata of the functional enrichment analysis (FEA) performed on the `macrophage` data, specifically using the `gost()` function from the `gprofiler2` package, and wrapped in `EMMA_run()`

The code to create said object can be found in the folder `/inst/scripts` in the EMMA package, the file is called `create_datasets_examples.R`.

Value

A sample `list` containing the FEA results `result` and `metadata`. This results object has the `EMMA_record` attribute.

References

Alasoo, et al. "Shared genetic effects on chromatin and gene expression indicate a role for enhancer priming in immune response", Nature Genetics, January 2018 doi: 10.1038/s41588-018-0046-7.

`gene_universe` *A sample* `character vector` *containing the background gene list used to perform FEA on the* `macrophage` *dataset*

Description

A sample `character vector` containing the background gene list used to perform FEA on the `macrophage` dataset

Format

A `character vector`

Details

This `character vector` object contains the assay's `rownames` of the `macrophage` data

The code to create said object can be found in the folder `/inst/scripts` in the EMMA package, the file is called `create_datasets_examples.R`.

Value

A sample `character vector` containing the assay's `rownames` of the `macrophage` data

References

Alasoo, et al. "Shared genetic effects on chromatin and gene expression indicate a role for enhancer priming in immune response", Nature Genetics, January 2018 doi: 10.1038/s41588-018-0046-7.

Index