

Package: GeneExprLupusMx (via r-universe)

October 31, 2026

Title Datos transcriptómicos de lupus en mujeres mexicanas con lecciones de aprendizaje

Version 0.1.0.9000

Description Paquete de datos con lecciones de aprendizaje bilingües (español e inglés). Incluye los perfiles transcriptómicos (RNA-seq) de monocitos, células dendríticas derivadas de monocitos (moDC) y células dendríticas tolerogénicas (tolDC) de 33 mujeres mexicanas (10 controles sanas y 23 pacientes con lupus eritematoso sistémico; 99 muestras), sus metadatos clínicos y los resultados intermedios de cada análisis. Las lecciones reproducen paso a paso las figuras del artículo original, de la estadística descriptiva a la expresión diferencial, el enriquecimiento funcional y los modelos con interacción, y se acompañan de funciones auxiliares para heatmaps y la reducción de términos GO. Data package with bilingual learning lessons (Spanish and English) on the transcriptomics of lupus in Mexican women, designed for teaching and reproducible research.

License MIT + file LICENSE

Encoding UTF-8

Roxygen list(markdown = TRUE)

URL <https://github.com/EveliaCoss/GeneExprLupusMx>,
<https://eveliacoss.github.io/GeneExprLupusMx/>

BugReports <https://github.com/EveliaCoss/GeneExprLupusMx/issues>

Depends R (>= 4.0)

Imports tools, utils

LazyData true

Language en, es

LazyDataCompression xz

Suggests circlize, ComplexHeatmap, cowplot, DESeq2, dplyr, ggh4x, ggplot2, ggrepel, knitr, MASS, org.Hs.eg.db, rmarkdown, rrvgo, scales, SummarizedExperiment, testthat (>= 3.0.0), tidyverse, withr

Config/testthat/edition 3

VignetteBuilder knitr

Config/roxygen2/version 8.1.0

Repository <https://ropensci-staging.r-universe.dev>

Date/Publication 2026-10-31 18:29:09 UTC

RemoteUrl <https://github.com/EveliaCoss/GeneExprLupusMx>

RemoteRef HEAD

RemoteSha 368a4c6e77def00cac13595326be24a60a9aa8bb

Contents

biotype_summary	3
create_heatmap	3
dc_genes	5
dds	5
de_celltype_results	6
de_interaction_results	7
de_results	8
download_data	9
filtered_txi	10
get_heatmap_mat	11
go_celltype_results	12
go_interaction_results	13
go_results	14
isg_genes	15
metadata_cleaned	15
moDC_genes	17
process_go	18
reduce_go_terms	19
sle_genes	20
tolDC_genes	21
volunteer_info	22
zscore_mat	23

Index	24
--------------	-----------

biotype_summary	<i>Gene biotypes</i>
-----------------	----------------------

Description

Biotype of each gene in filtered_txi, used to annotate differentially expressed genes as protein-coding genes (PCG) or long non-coding RNAs (lncRNA).

Usage

```
biotype_summary
```

Format

A data frame with 21,699 rows and 3 columns:

gene Gene symbol.

gene_biotype Ensembl biotype (e.g., "protein_coding", "lncRNA").

Group_biotype Grouped biotype: PCG, lncRNA, Pseudogene, small ncRNAs or TEC.

Details

The script that generates it is data-row/lesson4_annotations.R.

Source

Ensembl gene annotation.

Examples

```
table(biotype_summary$Group_biotype)
```

create_heatmap	<i>Create a ComplexHeatmap with row annotation</i>
----------------	--

Description

Generates a heatmap using the **ComplexHeatmap** package with a single-color row annotation. Designed to stack several heatmaps (for example, one per cell type) with consistent formatting.

Usage

```
create_heatmap(
  mat,
  color,
  split,
  common_names = rownames(mat),
  use_raster_flag = FALSE
)
```

Arguments

<code>mat</code>	A numeric matrix representing the heatmap values (e.g., Z-scores).
<code>color</code>	A character string specifying the annotation color (e.g., "black", "gold1").
<code>split</code>	A factor, character vector or data frame used to split the columns of the heatmap. NULL for no split.
<code>common_names</code>	A character vector of row names, used to size the annotation. By default, the row names of <code>mat</code> .
<code>use_raster_flag</code>	Logical. If TRUE, rasterization is enabled for faster rendering of large matrices (requires the magick package and a working Cairo/X11 installation). Default FALSE.

Details

Requires the Bioconductor package **ComplexHeatmap**.

Value

A Heatmap object from **ComplexHeatmap**, ready to be combined or drawn.

Examples

```
if (requireNamespace("ComplexHeatmap", quietly = TRUE)) {
  # small simulated Z-score matrix: 50 genes x 20 samples
  # (with the lesson data: zscore_mat <- download_data("zscore_mat"))
  set.seed(1)
  mat <- t(scale(t(matrix(rnorm(50 * 20), nrow = 50))))
  ht <- create_heatmap(mat, color = "black", split = NULL)
  ComplexHeatmap::draw(ht)
}
```

dc_genes

*Genes associated with dendritic cell differentiation and function***Description**

Curated list of genes associated with Gene Ontology terms related to dendritic cells, grouped by biological process. Used to explore canonical dendritic cell differentiation pathways (Figure 5B of the article).

Usage

dc_genes

Format

A data frame with 94 rows and 4 columns:

gene Gene symbol.

process Biological process (e.g., "DC differentiation", "Migration", "TAM signaling").

GOTerm Group of Gene Ontology terms used to select the gene.

info Gene Ontology term associated with the gene.

Details

The script that generates it is data-raw/lesson5_data.R.

Source

Gene Ontology annotations (<https://geneontology.org>).

Examples

```
table(dc_genes$process)
```

dds

*DESeqDataSet checkpoint from the RNA-seq lesson***Description**

DESeqDataSet built in the lesson "De las cuentas de RNA-seq a la exploración de datos" ([intro_RNAseq_es](#)), ready to be used as the starting point of later lessons.

Format

A DESeqDataSet with 13,596 genes and 99 samples. Its colData is metadata_cleaned plus a Volunteer_ID column.

Details

This object is not installed with the package because of its size: get it with `dds <- download_data("dds")` (see [download_data](#)).

Steps: `DESeqDataSetFromTximport()` from `filtered_txi` and `metadata_cleaned` with design `~ Cell_type + Group` (reference levels `monocyte` and `Ctrl`), removal of genes with fewer than 10 reads in at least 10 samples, and `estimateSizeFactors()`.

Requires the **DESeq2** package (Bioconductor) to be used. The script that generates it is `data-raw/lesson2_checkpoints.R`.

Source

Generated from `filtered_txi` and `metadata_cleaned`.

See Also

[zscore_mat](#), [filtered_txi](#), [metadata_cleaned](#)

Examples

```
library(DESeq2)
dds <- download_data("dds")
dds
design(dds)
```

de_celltype_results	<i>Differential expression results: dendritic cells vs monocytes</i>
---------------------	--

Description

DESeq2 results comparing in vitro differentiated dendritic cells (moDC or tolDC) against monocytes, within each group (healthy controls and SLE patients). Used in the lesson about monocyte differentiation ([l5_DEGs_monocytes_diff_es](#); Figures 4 and 5 of the article).

Usage

```
de_celltype_results
```

Format

A named list with four data frames (`Ctrl_moDC`, `Ctrl_tolDC`, `SLE_moDC`, `SLE_tolDC`), each with 21,718 rows (genes) and 7 columns: `gene`, `baseMean`, `log2FoldChange`, `lfcSE`, `stat`, `pvalue` and `padj` (see [de_results](#)).

Details

Designs (all genes of filtered_txi, reference level monocyte):

- Controls: $\sim$ Cell_type.
- SLE patients: $\sim$ SLEDAI_category + Cell_type + Dose_category (adjusted for disease activity and glucocorticoid dose).

Results were extracted with `results(alpha = 0.05, name = "Cell_type_<moDC|tolDC>_vs_monocyte")` (Wald test, Benjamini-Hochberg adjustment).

With $|\log_2\text{FoldChange}| \geq 1$ and $\text{padj} < 0.05$ (up / down): Ctrl moDC 2688 / 2947, Ctrl tolDC 2731 / 2849, SLE moDC 2945 / 3652 and SLE tolDC 2804 / 3596. The article (DESeq2 on R 4.0.2) reports the same numbers for moDC and differs by a few genes in tolDC (2736 / 2851 and 2804 / 3592). The script that generates this object is `data-raw/lesson5_checkpoints.R`.

Source

Generated from [filtered_txi](#) and [metadata_cleaned](#).

See Also

[de_results](#) (SLE vs Ctrl within each cell type).

Examples

```
names(de_celltype_results)
head(de_celltype_results$Ctrl_moDC[order(de_celltype_results$Ctrl_moDC$padj), ])
```

de_interaction_results

Differential expression results: between-group comparison of contrasts

Description

DESeq2 interaction results that test whether the change from monocytes to in vitro differentiated dendritic cells (moDC or tolDC) is different in SLE patients than in healthy controls. Used in the lesson about the between-group comparison of contrasts ([17_DEGs_between_groups_es](#); Figure 7A and 7B of the article).

Usage

```
de_interaction_results
```

Format

A named list with two data frames (moDC and tolDC), each with 21,718 rows (genes) and 7 columns: gene, baseMean, log2FoldChange, lfcSE, stat, pvalue and padj (see [de_results](#)).

Details

Design (all genes of filtered_txi, the 99 samples, reference levels Ctrl and monocyte): ~ Group + Cell_type + Group:Cell_type. Results were extracted with results(alpha = 0.05, name = "GroupSLE.Cell_type<moDC|tolDC>") (Wald test, Benjamini-Hochberg adjustment). The log2FoldChange is a difference of differences: (dendritic cell vs monocyte in SLE) minus (dendritic cell vs monocyte in Ctrl).

With padj < 0.05 (up / down): moDC 1 / 28 and tolDC 3 / 38. The script that generates this object is data-raw/lesson7_checkpoints.R.

Source

Generated from [filtered_txi](#) and [metadata_cleaned](#).

See Also

[de_celltype_results](#) (dendritic cells vs monocytes within each group), [go_interaction_results](#).

Examples

```
names(de_interaction_results)
head(de_interaction_results$tolDC[order(de_interaction_results$tolDC$padj), ])
```

de_results

Differential expression results checkpoint (SLE vs Ctrl by cell type)

Description

DESeq2 results of the comparison between SLE patients and healthy controls within each cell type, generated in the lesson "Genes diferencialmente expresados entre pacientes y controles" ([I3_DEGs_es](#)). Starting point of the functional enrichment lesson.

Usage

```
de_results
```

Format

A named list with three data frames (monocyte, moDC, tolDC), each with 21,718 rows (genes) and 7 columns:

gene Gene symbol.

baseMean Mean normalized expression across samples.

log2FoldChange Log2 fold change, SLE vs Ctrl.

lfcSE Standard error of the log2 fold change.

stat Wald statistic.

pvalue Wald test p-value.

padj Benjamini-Hochberg adjusted p-value (NA for genes removed by independent filtering).

Details

For each cell type: `DESeqDataSetFromTximport()` with all genes of `filtered_txi`, `design ~ Group (reference Ctrl)`, `DESeq()` and `results(name = "Group_SLE_vs_Ctrl")` (Wald test, Benjamini-Hochberg adjustment).

With $|\log_2\text{FoldChange}| \geq 1$ and $\text{padj} < 0.05$: 37 up / 0 down in monocytes, 4 up / 40 down in moDC and 4 up / 12 down in tolDC (94 distinct genes). The script that generates it is `data-raw/lesson3_checkpoints.R`.

Source

Generated from `filtered_txi` and `metadata_cleaned`.

Examples

```
names(de_results)
head(de_results$monocyte[order(de_results$monocyte$padj), ])
```

download_data	<i>Download the large data objects of the package</i>
---------------	---

Description

The three largest objects (`filtered_txi`, `dds` and `zscore_mat`) are not installed with the package, to keep it small. They are stored as files of the GitHub release of the package, and this function downloads them the first time they are used. Later calls read the copy saved on your computer, so no internet connection is needed.

Usage

```
download_data(name = c("filtered_txi", "dds", "zscore_mat"), tag = "v0.1.0")
```

Arguments

name	Name of the object: "filtered_txi", "dds" or "zscore_mat".
tag	GitHub release that contains the files. The default is the release with the data used in the lessons.

Details

Files are saved in `tools::R_user_dir("GeneExprLupusMx", "cache")`. Delete that folder to free space or to download them again. The first download needs about 46 MB in total.

Value

The requested object.

See Also

`filtered_txi`, `dds`, `zscore_mat`

Examples

```
filtered_txi <- download_data("filtered_txi")
dim(filtered_txi$counts)
```

filtered_txi

Gene-level RNA-seq quantification (tximport)

Description

Gene-level RNA-seq quantification of the 99 samples of the study (33 volunteers x 3 cell types: monocytes, moDCs and tolDCs). Transcripts were quantified with **Salmon** and summarized to genes with **tximport**. Genes are identified by their gene symbol; duplicated entries in gene biotypes were removed before creating this object.

Format

A list with the structure returned by `tximport::tximport()`:

abundance Numeric matrix (21,718 genes x 99 samples) of gene abundances in transcripts per million (TPM).

counts Numeric matrix (21,718 genes x 99 samples) of estimated read counts. Values are not integers because Salmon estimates them.

length Numeric matrix (21,718 genes x 99 samples) with the average effective transcript length of each gene in each sample; **DESeq2** uses it as a normalization offset.

countsFromAbundance Character, "no": counts were not derived from abundances.

Details

This object is not installed with the package because of its size: get it with `filtered_txi <- download_data("filtered_txi")` (see [download_data](#)).

Row names are gene symbols (e.g., "A1BG") and column names are sample identifiers (e.g., "QR011_1"), in the same order as the rows of [metadata_cleaned](#). Use it with `DESeq2::DESeqDataSetFromTximport()`.

Source

Hernández-Ledesma AL, Coss-Navarrete EL, et al. (2026), Front. Immunol. 17:1771959 ([doi:10.3389/fimmu.2026.1771959](#)); data and analysis code deposited in Zenodo ([doi:10.5281/zenodo.17742206](#)).

See Also

[metadata_cleaned](#), [biotype_summary](#)

Examples

```

filtered_txi <- download_data("filtered_txi")
names(filtered_txi)
dim(filtered_txi$counts)
filtered_txi$counts[1:4, 1:4]

```

get_heatmap_mat	<i>Generate a heatmap-ready matrix from a DESeq2 object</i>
-----------------	---

Description

Applies the variance stabilizing transformation (VST) to a DESeqDataSet, removes genes with zero variance, converts each gene to Z-scores across samples and orders the columns.

Usage

```
get_heatmap_mat(dds_object, sample_order = colnames(dds_object))
```

Arguments

dds_object	A DESeqDataSet (for example, the samples of a single cell type).
sample_order	Character vector with the sample IDs in the order in which the columns should appear. By default, the order of dds_object.

Details

Requires the Bioconductor packages **DESeq2** and **SummarizedExperiment**.

Value

A numeric matrix of Z-scores (genes in rows, samples in columns).

Examples

```

if (requireNamespace("DESeq2", quietly = TRUE)) {
  # small simulated DESeqDataSet: 1,200 genes x 6 samples
  # (with the lesson data: dds <- download_data("dds"))
  set.seed(1)
  counts <- matrix(rnbinom(1200 * 6, mu = 100, size = 10), ncol = 6,
                  dimnames = list(paste0("gene", 1:1200), paste0("S", 1:6)))
  col_data <- data.frame(Group = factor(rep(c("Ctrl", "SLE"), each = 3)),
                        row.names = colnames(counts))
  sim_dds <- DESeq2::DESeqDataSetFromMatrix(counts, col_data, design = ~ Group)
  sim_mat <- get_heatmap_mat(sim_dds)
  dim(sim_mat)
}

```

go_celltype_results	<i>Functional enrichment checkpoint: dendritic cells vs monocytes (g:Profiler)</i>
---------------------	--

Description

Functional enrichment of the differentially expressed genes (DEG) between dendritic cells and monocytes ([de_celltype_results](#)), obtained with **gprofiler2** in the lesson about monocyte differentiation ([l5_DEGs_monocytes_diff_es](#)).

Usage

```
go_celltype_results
```

Format

A data frame with one row per significant term and the same 10 columns as [go_results](#): query, source, term_id, term_name, term_size, query_size, intersection_size, p_value, parents and intersection.

Details

DEG were defined with $|\log_2\text{FoldChange}| \geq 1$ and $\text{padj} < 0.05$ and queried as eight lists (<Ctrl|SLE>_<UP|DOWN>_<mod> one at a time, with `gost(organism = "hsapiens", correction_method = "fdr", user_threshold = 0.05, sources = c("GO:BP", "KEGG", "REAC"), evcodes = TRUE)`.

The archived g:Profiler version e113_eg59_p19 (2025) was used; it is stored in `attr(go_celltype_results, "gprofiler_version")`. The script that generates it is `data-raw/lesson5_checkpoints.R`.

Source

g:Profiler (<https://biit.cs.ut.ee/gprofiler/>).

See Also

[de_celltype_results](#), [go_results](#)

Examples

```
table(go_celltype_results$query)
```

`go_interaction_results`*Functional enrichment checkpoint: between-group comparison of contrasts (g:Profiler)*

Description

Functional enrichment of the genes whose change from monocytes to dendritic cells differs between SLE patients and controls ([de_interaction_results](#)), obtained with **gprofiler2** in [l7_DEGs_between_groups_es](#).

Usage

`go_interaction_results`

Format

A data frame with one row per significant term and the same 10 columns as [go_results](#): query, source, term_id, term_name, term_size, query_size, intersection_size, p_value, parents and intersection.

Details

Genes with $\text{padj} < 0.05$ were split by the sign of $\log_2\text{FoldChange}$ and queried as four lists (SLEvsCtrl1_<UP|DOWN>_<moDC| with `gost(organism = "hsapiens", correction_method = "fdr", user_threshold = 0.05, sources = c("GO:BP", "KEGG", "REAC"), evcodes = TRUE)`.

The archived g:Profiler version e113_eg59_p19 (2025) was used; it is stored in `attr(go_interaction_results, "gprofiler_version")`. The script that generates it is `data-raw/lesson7_checkpoints.R`.

Source

g:Profiler (<https://biit.cs.ut.ee/gprofiler/>).

See Also

[de_interaction_results](#), [go_celltype_results](#)

Examples

```
table(go_interaction_results$query, go_interaction_results$source)
```

go_results

*Functional enrichment results checkpoint (g:Profiler)***Description**

Functional enrichment of the differentially expressed genes (DEG) of each cell type, obtained with **gprofiler2** in the lesson "Funciones biológicas de los genes diferencialmente expresados" ([14_GOterms_es](#)).

Usage

go_results

Format

A data frame with 943 rows (significant terms) and 10 columns:

query Gene list (cell type and direction of change).

source Database: GO:BP, GO:CC, GO:MF, REAC, KEGG, WP, TF, MIRNA, HP or CORUM.

term_id Term identifier.

term_name Term name.

term_size Number of genes annotated to the term.

query_size Number of genes of the list with annotation in the database.

intersection_size Number of genes of the list annotated to the term.

p_value Adjusted p-value (FDR) of the hypergeometric test.

parents Comma-separated parent terms in the database hierarchy.

intersection Comma-separated genes of the list annotated to the term.

Details

DEG were defined in [de_results](#) with $|\log_2\text{FoldChange}| \geq 1$ and $\text{padj} < 0.05$, and queried as five lists (SLE_UP_monocyte, SLE_UP_moDC, SLE_DOWN_moDC, SLE_UP_to1DC, SLE_DOWN_to1DC) with `gost(organism = "hsapiens", correction_method = "fdr", user_threshold = 0.05, evcodes = TRUE)`.

g:Profiler updates its databases periodically. To reproduce the article, the archived version e113_eg59_p19 (June 2025) was used; it is stored in `attr(go_results, "gprofiler_version")`. The script that generates it is `data-raw/lesson4_checkpoints.R`.

Source

g:Profiler (<https://biit.cs.ut.ee/gprofiler/>).

Examples

```
table(go_results$query)
table(go_results$source)
```

isg_genes	<i>Interferon-stimulated genes (ISGs)</i>
-----------	---

Description

List of 423 interferon-stimulated genes, used to identify differentially expressed ISGs.

Usage

isg_genes

Format

A data frame with 423 rows and 1 column:

genes Gene symbol.

Details

The script that generates it is data-raw/lesson4_annotations.R.

Source

Yu et al. (2025), Table S3: "The list of interferon-stimulated genes" ([doi:10.6084/m9.figshare.28353289](https://doi.org/10.6084/m9.figshare.28353289)).

Examples

```
head(isg_genes)
"IFI27" %in% isg_genes$genes
```

metadata_cleaned	<i>Sample metadata: group, cell type and clinical information</i>
------------------	---

Description

Metadata of the 99 RNA-seq samples: 33 volunteers (10 healthy controls and 23 women with systemic lupus erythematosus, SLE), each contributing three cell types: monocytes (monocyte), monocyte-derived dendritic cells (moDC) and tolerogenic dendritic cells (to1DC). Clinical variables are repeated in the three samples of each volunteer.

Usage

metadata_cleaned

Format

A data frame with 99 rows (samples) and 14 columns. All columns are factors.

sample_ID Sample identifier: volunteer ID plus cell type suffix (`_0` = monocyte, `_1` = moDC, `_3` = tolDC; e.g., "QR011_1"). Also used as row names.

Group Ctrl (healthy control) or SLE (patient).

GC Glucocorticoid treatment: No or Si.

Dose Glucocorticoid dose in mg (0, 2.5, 3, 5 or 8); 0 for controls and untreated patients.

Dose_category Glucocorticoid treatment coded as 0 (no) or 1 (yes); equivalent to GC.

Age Age in years.

Cell_type Cell type: monocyte, moDC or tolDC.

Group_age Age group: less30 (younger than 30 years) or more30 (30 years or older).

Nephritis Lupus nephritis: No or Si; NA in controls.

Other_tx Treatment other than glucocorticoids (e.g., antimalarials, immunosuppressants): No or Si; NA in controls.

Sledai_score SLEDAI disease activity score (0 to 62); NA in controls.

Sledai_group SLEDAI activity group: Without_activity, Low, Moderate, High or Very_high; NA in controls.

SLEDAI_Md Yes/no indicator from the original clinical records (No, Si or Sí); NA in controls.

SLEDAI_category Disease activity category: Zero (SLEDAI score 0, and all controls), No_activity (score 1-3) or Activity (score 4 or more).

Details

- Rows are in the same order as the columns of the count matrices of [filtered_txi](#).
- Age, Dose and Sledai_score are stored as factors; convert them with `as.numeric(as.character(x))` before doing calculations.
- The volunteer identifier (e.g., "QR011") is the prefix of sample_ID, `sub("_.*", "", sample_ID)`; it links this table with [volunteer_info](#).
- In SLEDAI_Md, the "yes" level appears as Si and Sí; combine them before counting, for example with `sub("Sí", "Si", SLEDAI_Md)`.
- Sledai_score, Sledai_group and SLEDAI_category are groupings of disease activity used in the exploratory analyses. To describe the volunteers as in Table 1 of the article (age, treatments and active disease, SLEDAI 6 or more), use [volunteer_info](#), which has one row per volunteer, as in the lessons.

Source

Hernández-Ledesma AL, Coss-Navarrete EL, et al. (2026), Front. Immunol. 17:1771959 ([doi:10.3389/fimmu.2026.1771959](#)); data and analysis code deposited in Zenodo ([doi:10.5281/zenodo.17742206](#)).

See Also

[filtered_txi](#), [volunteer_info](#)

Examples

```
head(metadata_cleaned)
table(metadata_cleaned$Group, metadata_cleaned$Cell_type)

# Age as a number
summary(as.numeric(as.character(metadata_cleaned$Age)))
```

moDC_genes

Genes described in the differentiation of monocytes into moDCs

Description

Curated list of genes reported in the literature as up- or down-regulated when monocytes differentiate in vitro into monocyte-derived dendritic cells (moDCs), grouped by biological process. Used to check the expected transcriptional changes (Figure 4C of the article).

Usage

```
moDC_genes
```

Format

A data frame with 90 rows and 7 columns:

gene Gene symbol.

Gene_name Name used in the source publication (e.g., "IL-1beta").

process Biological process (e.g., "Maturation", "Cytokines").

Associated_pathways Pathway or description reported in the source.

Expression Reported direction: Upregulated or Downregulated.

link Link to the source publication.

contrast Comparison in which the change was reported.

Details

The script that generates it is data-raw/lesson5_data.R. Text was converted to ASCII (Greek letters spelled out) and the spreadsheet artifact "GBP 2.00" was corrected to "GBP2". Eleven entries use aliases or old names (CD31, CD54, IL8, IRF7A, ERF2, CD32, TNFR2, IP3KB, HSP27, GRP78) or genes absent from the data (CXCL8); they do not match the gene symbols of filtered_tx1 and, as in the original analysis, are not shown.

Source

See the link column.

Examples

```
table(moDC_genes$process)
```

process_go

*Process GO Enrichment Result from a Single File***Description**

Load and analyze a single .RData file containing GO enrichment results from either UP or DOWN regulated genes. Includes validation, semantic similarity reduction, merging metadata, and plotting.

Usage

```
process_go(f, direction = c("UP", "DOWN"), outdir, figdir)
```

Arguments

f	Character. Path to the .RData file (must include full or relative path). The file must contain exactly one data frame with the columns go_id, term and p.adjust.
direction	Character. Either "UP" or "DOWN" to specify the type of enrichment.
outdir	Character. Directory to save .RData output files.
figdir	Character. Directory to save PNG plots.

Details

Requires the Bioconductor packages **rrvgo** and **org.Hs.eg.db**. Datasets with fewer than 40 unique GO terms are skipped.

Value

Invisibly, NULL. Saves processed files and figures in outdir and figdir.

See Also

[reduce_go_terms](#) for batch processing of multiple enrichment files.

Examples

```
if (all(vapply(c("rrvgo", "org.Hs.eg.db", "dplyr", "ggplot2"),
               requireNamespace, logical(1), quietly = TRUE))) {
  # Enrichment table with the columns go_id, term and p.adjust
  go_bp <- go_celltype_results[go_celltype_results$query == "SLE_UP_moDC" &
                              go_celltype_results$source == "GO:BP", ]
  go_bp <- head(go_bp[order(go_bp$p_value), ], 60)
  go_table <- data.frame(go_id = go_bp$term_id, term = go_bp$term_name,
                        p.adjust = go_bp$p_value)

  go_dir <- file.path(tempdir(), "process_go_example")
  dir.create(go_dir, showWarnings = FALSE)
  f <- file.path(go_dir, "UP_GO_SLE_moDC.RData")
}
```

```

    save(go_table, file = f)

    process_go(f, direction = "UP", outdir = go_dir, figdir = go_dir)
    list.files(go_dir)
  }

```

reduce_go_terms

Batch Reduce and Visualize GO Terms Across Multiple Datasets

Description

Applies semantic similarity reduction to GO enrichment results from a batch of .RData files (both UP and DOWN regulated genes). For each dataset, it invokes [process_go](#) to reduce redundancy, merge metadata, and generate visualizations.

Usage

```
reduce_go_terms(indir, outdir, figdir)
```

Arguments

indir	Character. Input directory containing GO enrichment .RData files. Expected filenames follow the pattern UP_GO_<name>.RData and DOWN_GO_<name>.RData.
outdir	Character. Output directory to save reduced GO term .RData files.
figdir	Character. Directory to save output plots (treemaps and bar plots).

Details

This wrapper function automates analysis across multiple files following the naming pattern UP_GO_<name>.RData or DOWN_GO_<name>.RData, typically produced by GO enrichment pipelines.

For each file, the function:

- Loads the GO enrichment result.
- Filters invalid, duplicated, or missing GO IDs.
- Skips datasets with fewer than 40 unique GO terms.
- Computes semantic similarity with `rrvgo::calculateSimMatrix()` using the "Rel" method.
- Reduces terms via `rrvgo::reduceSimMatrix()` with a similarity threshold of 0.7.
- Saves both the reduced term object and merged gene-GO mapping.
- Generates treemap and bar plots showing parent GO categories.

Internal errors in similarity computation are caught with `tryCatch()` to allow uninterrupted batch processing across datasets.

Requires the Bioconductor packages **rrvgo** and **org.Hs.eg.db**.

Value

Invisibly, NULL. It generates and saves .RData and .png files in the specified output directories.

See Also

[process_go](#) for analyzing individual files.

Examples

```
if (all(vapply(c("rrvgo", "org.Hs.eg.db", "dplyr", "ggplot2"),
               requireNamespace, logical(1), quietly = TRUE))) {
  indir <- file.path(tempdir(), "reduce_go_terms_example")
  dir.create(indir, showWarnings = FALSE)

  # One file per enrichment result, named UP_GO_<name>.RData or DOWN_GO_<name>.RData
  for (q in c("SLE_UP_moDC", "SLE_DOWN_moDC")) {
    go_bp <- go_celltype_results[go_celltype_results$query == q &
                                go_celltype_results$source == "GO:BP", ]
    go_bp <- head(go_bp[order(go_bp$p_value), ], 60)
    go_table <- data.frame(go_id = go_bp$term_id, term = go_bp$term_name,
                          p.adjust = go_bp$p_value)
    direction <- if (grepl("_UP_", q, fixed = TRUE)) "UP" else "DOWN"
    save(go_table, file = file.path(indir, paste0(direction, "_GO_moDC.RData")))
  }

  reduce_go_terms(indir = indir, outdir = tempdir(), figdir = tempdir())
}
```

sle_genes	<i>Genes associated with systemic lupus erythematosus (SLE)</i>
-----------	---

Description

Genes reported as associated with SLE in the literature, used to identify SLE-related genes among the differentially expressed genes.

Usage

sle_genes

Format

A data frame with 219 rows (213 unique genes; some genes are reported by more than one study) and 4 columns:

Gene_name Gene symbol.

Associated_pathways Pathway or function associated with the gene (NA if not reported).

Reference Study that reports the association.
Link Link to the study.

Details

Compiled from Shen et al. (2022), Deng and Tsao (2017), Bing et al. (2016) and Ortiz-Fernandez et al. (2023). The script that generates it is data-raw/lesson4_annotations.R.

Source

See the Link column.

Examples

```
table(sle_genes$Reference)
length(unique(sle_genes$Gene_name))
```

tolDC_genes	<i>Transcriptional signature of tolerogenic dendritic cells (tolDCs)</i>
-------------	--

Description

Genes reported as activated (up-regulated) or repressed (down-regulated) in tolerogenic dendritic cells by Robertson et al. (2021). Used to evaluate whether the tolDCs of the study acquire the expected tolerogenic signature (Figure 6 of the article; **l6_DEGs_tolDcs_es**).

Usage

```
tolDC_genes
```

Format

A data frame with 86 rows and 4 columns:

- gene** Gene symbol.
- tolDC_signature** Category reported by Robertson et al.: direction (Activated or Repressed) and set in which the gene was found: tolDCs generated with IL-10 (IL10), with IL-10 plus dexamethasone (IL10+DXM) or the consensus tolDC signature (tolDC_signature). Genes found in several sets list all of them, separated by commas.
- biotype** Gene biotype (PCG or lncRNA).
- link** Link to the source publication.

Details

The script that generates it is data-raw/lesson6_annotations.R.

Source

Robertson H, et al. (2021). Transcriptomic analysis identifies a tolerogenic dendritic cell signature. Front Immunol 12:733231. doi:10.3389/fimmu.2021.733231

Examples

```
table(tolDC_genes$tolDC_signature)
```

volunteer_info	<i>Clinical information per volunteer</i>
----------------	---

Description

Clinical and recruitment information with one row per volunteer (10 healthy controls and 23 women with systemic lupus erythematosus). Each volunteer contributed three samples (monocyte, moDC and tolDC) to metadata_cleaned; both tables are linked by Volunteer_ID, the prefix of metadata_cleaned\$sample_ID (e.g. "QR011_0" -> "QR011").

Usage

```
volunteer_info
```

Format

A data frame with 33 rows and 16 columns:

Volunteer_ID Volunteer identifier (e.g. "QR011").

Volunteer Volunteer number, in order of recruitment.

Group Ctrl (healthy control) or SLE (patient).

Age_years Age in years.

Years_with_diagnosis Years since SLE diagnosis (NA in controls).

Age_at_diagnosis_years Age at SLE diagnosis (NA in controls).

SLEDAI SLEDAI disease activity: 1 = active (SLEDAI >= 6 points), 0 = no or low activity (0-5 points); NA in controls.

Nephritis Lupus nephritis: No or Yes.

Treatment_GCC Receives glucocorticoids (TRUE/FALSE).

GCC_drug Glucocorticoid as recorded: PDN or PDS.

GCC_dose_mg Glucocorticoid dose in mg (NA if not recorded).

Treatment_AM Receives antimalarials.

Treatment_MMF Receives mycophenolate mofetil.

Treatment_AZT Receives azathioprine.

Treatment_MTT Receives methotrexate.

Time_of_recruitment Month of recruitment and sample collection (Date, first day of the month).

Details

Treatment columns are NA for controls (not applicable). The cleaning script is in data-raw/volunteer_info.R.

Source

Hernández-Ledesma AL, Coss-Navarrete EL, et al. (2026), Front. Immunol. 17:1771959 (doi:10.3389/fimmu.2026.1771959); data and analysis code deposited in Zenodo (doi:10.5281/zenodo.17742206).

Examples

```
data(volunteer_info)
head(volunteer_info)
table(volunteer_info$Group, format(volunteer_info$Time_of_recruitment, "%Y-%m"))
```

zscore_mat	<i>Z-score matrix checkpoint from the RNA-seq lesson</i>
------------	--

Description

Gene-wise z-scores of the variance-stabilized expression (VST) computed in the lesson "De las cuentas de RNA-seq a la exploración de datos" (intro_RNAseq_es). Useful for heatmaps and clustering.

Format

A numeric matrix with 13,596 genes (rows) and 99 samples (columns).

Details

This object is not installed with the package because of its size: get it with `zscore_mat <- download_data("zscore_mat")` (see [download_data](#)).

Steps: `vst(dds, blind = FALSE)`, removal of genes with zero variance, and `t(scale(t(...)))` so that each gene has mean 0 and standard deviation 1 across samples.

The script that generates it is data-raw/lesson2_checkpoints.R.

Source

Generated from [dds](#).

See Also

[dds](#)

Examples

```
zscore_mat <- download_data("zscore_mat")
dim(zscore_mat)
zscore_mat[1:4, 1:4]
```

Index

* datasets

- biotype_summary, [3](#)
- dc_genes, [5](#)
- dds, [5](#)
- de_celltype_results, [6](#)
- de_interaction_results, [7](#)
- de_results, [8](#)
- filtered_txi, [10](#)
- go_celltype_results, [12](#)
- go_interaction_results, [13](#)
- go_results, [14](#)
- isg_genes, [15](#)
- metadata_cleaned, [15](#)
- moDC_genes, [17](#)
- sle_genes, [20](#)
- tolDC_genes, [21](#)
- volunteer_info, [22](#)
- zscore_mat, [23](#)

process_go, [18](#), [19](#), [20](#)

reduce_go_terms, [18](#), [19](#)

sle_genes, [20](#)

tolDC_genes, [21](#)

volunteer_info, [16](#), [22](#)

zscore_mat, [6](#), [9](#), [23](#)

biotype_summary, [3](#), [10](#)

create_heatmap, [3](#)

dc_genes, [5](#)

dds, [5](#), [9](#), [23](#)

de_celltype_results, [6](#), [8](#), [12](#)

de_interaction_results, [7](#), [13](#)

de_results, [6](#), [7](#), [8](#), [14](#)

download_data, [6](#), [9](#), [10](#), [23](#)

filtered_txi, [6–9](#), [10](#), [16](#)

get_heatmap_mat, [11](#)

go_celltype_results, [12](#), [13](#)

go_interaction_results, [8](#), [13](#)

go_results, [12](#), [13](#), [14](#)

isg_genes, [15](#)

metadata_cleaned, [6–10](#), [15](#)

moDC_genes, [17](#)