

# Package ‘scMappR’

November 30, 2020

**Title** Single Cell Mapper

**Version** 0.1.5

**Description** Description of scMappR R package adapted from pre-print. The single cell mapper (scMappR) R package contains a suite of bioinformatic tools that provide experimentally relevant cell-type specific information to a list of differentially expressed genes (DEG). The function ```scMappR_and_pathway_analysis``` reranks DEGs to generate cell-type specificity scores called cell-weighted fold-changes. Users input a list of DEGs, normalized counts, and a signature matrix into this function. scMappR then re-weights bulk DEGs by cell-type specific expression from the signature matrix, cell-type proportions from RNA-seq deconvolution and the ratio of cell-type proportions between the two conditions to account for changes in cell-type proportion. With ```cwFold-changes``` calculated, scMappR uses two approaches to utilize ```cwFold-changes``` to complete cell-type specific pathway analysis. The ```process_dgTMatrix_lists``` function in the scMappR package contains an automated scRNA-seq processing pipeline where users input scRNA-seq count data, which is made compatible for scMappR and other R packages that analyze scRNA-seq data. We further used this to store hundreds up regularly updating signature matrices. The functions ```tissue_by_celltype_enrichment```, ```tissue_scMappR_internal```, and ```tissue_scMappR_custom``` combine these consistently processed scRNAseq count data with gene-set enrichment tools to allow for cell-type marker enrichment of a generic gene list (e.g. GWAS hits). Reference: Sokolowski,D.J., Faykoo-Martinez,M., Erdman,L., Hou,H., Chan,C., Zhu,H., Holmes,M.M., Goldenberg,A. and Wilson,M.D. (2020) Single-cell mapper (scMappR): using scRNA-seq to infer cell-type specificities of differentially expressed genes. BioRxiv, 10.1101/2020.08.24.265298.

**Depends** R (>= 4.0.0)

**Imports** ggplot2, pheatmap, graphics, Seurat, GSVA, stats, utils, downloader, pcaMethods, grDevices, gProfileR, limSolve, gprofiler2, pbapply, ADAPTS, reshape,

**License** GPL-3

**URL**

**Encoding** UTF-8

**LazyData** true

**RoxygenNote** 7.1.1

**Suggests** testthat, knitr, rmarkdown

**VignetteBuilder** knitr

**NeedsCompilation** no

**Author** Dustin Sokolowski [aut, cre],  
 Mariela Faykoo-Martinez [aut],  
 Lauren Erdman [aut],  
 Houyun Hou [aut],  
 Cadia Chan [aut],  
 Helen Zhu [aut],  
 Melissa Holmes [aut],  
 Anna Goldenberg [aut],  
 Michael Wilson [aut]

**Maintainer** Dustin Sokolowski <dustin.sokolowski@sickkids.ca>

**Repository** CRAN

**Date/Publication** 2020-11-30 18:00:02 UTC

## R topics documented:

cellmarker_enrich . . . . .	3
coEnrich . . . . .	4
compare_deconvolution_methods . . . . .	5
cwFoldChange_evaluate . . . . .	7
DeconRNAseq_CRAN . . . . .	9
deconvolute_and_contextualize . . . . .	10
extract_genes_cell . . . . .	12
genes_to_heatmap . . . . .	13
get_gene_symbol . . . . .	15
get_signature_matrices . . . . .	16
gmt . . . . .	16
gProfiler_cellWeighted_Foldchange . . . . .	17
gsva_cellIdentify . . . . .	18
heatmap_generation . . . . .	20
human_mouse_ct_marker_enrich . . . . .	21
make_TF_barplot . . . . .	22
pathway_enrich_internal . . . . .	24
PBMC_example . . . . .	25
plotBP . . . . .	26
POA_example . . . . .	27
process_dgTMatrix_lists . . . . .	28
process_from_count . . . . .	30
scMappR_and_pathway_analysis . . . . .	31
scMappR_tissues . . . . .	34
seurat_to_genes . . . . .	35
single_gene_preferences . . . . .	36
sm . . . . .	37
tissue_by_celltype_enrichment . . . . .	38
tissue_scMappR_custom . . . . .	39

<i>cellmarker_enrich</i>	3
tissue_scMappR_internal . . . . .	40
tochr . . . . .	42
toNum . . . . .	43
topgenes_extract . . . . .	44
two_method_pathway_enrichment . . . . .	45
<b>Index</b>	<b>47</b>

---

cellmarker_enrich	<i>Fisher's Exact Cell-Type Identification.</i>
-------------------	---

---

### Description

This function uses the CellMarker and Panglao datasets to identify cell-type differentially expressed genes.

### Usage

```
cellmarker_enrich(
  gene_list,
  p_thresh,
  gmt = "cellmarker_list.Rdata",
  fixed_length = 13000,
  min_genes = 5,
  max_genes = 3000,
  isect_size = 3
)
```

### Arguments

gene_list	A character vector of gene symbols with the same designation (e.g. mouse symbol - mouse, human symbol - human) as the gene set database.
p_thresh	The Fisher's test cutoff for a cell-marker to be enriched.
gmt	Either a path to an rda file containing an object called "gmt", which is a named list where each element of the list is a vector of gene symbols website for more detail on the file type ( <a href="https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_format">https://software.broadinstitute.org/cancer/software/gsea/wiki/index.php/Data_format</a> ). The gmt list may also be inputted.
fixed_length	Estimated number of genes in your background.
min_genes	Minimum number of genes in the cell-type markers.
max_genes	Maximum number of genes in the cell-type markers.
isect_size	Number of genes in your list and the cell-type.

### Details

Complete a Fisher's exact test of an input list of genes against a gene set saved in an \*.RData object. The RData object is storing a named list of genes called "gmt".

**Value**

cellmarker\_enrich Gene set enrichment of cell-types on your inputted gene list.

**Examples**

```
data(POA_example)
POA_genes <- POA_example$POA_genes
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
Signature <- POA_Rank_signature
rowname <- get_gene_symbol(Signature)
rownames(Signature) <- rowname$rowname
genes <- rownames(Signature)[1:100]
data(gmt)
enriched <- cellmarker_enrich(gene_list = genes, p_thresh = 0.05, gmt = gmt)
```

---

coEnrich

*Identify co-expressed cell-types*


---

**Description**

This function identifies genes with similar cell-type markers and if those markers are driving enrichment.

**Usage**

```
coEnrich(
  sig,
  gene_list_heatmap,
  background_heatmap,
  study_name,
  outDir,
  toSave = FALSE,
  path = NULL
)
```

**Arguments**

**sig** A The number of combinations of significant cell-types to enrich.

**gene\_list\_heatmap** Signature matrix of inputted genes in heatmap and the cell-type preferences – output of heatmap generation.

background_heatmap	Signature matrix of background matrix in heatmap and cell-type preferences – output of heatmap generation.
study_name	Name of the outputted table.
outDir	Name of the directory this table will be printed in.
toSave	Allow scMappR to write files in the current directory (T/F).
path	If toSave == TRUE, path to the directory where files will be saved.

### Details

This function takes significantly enriched cell-types from the single CT\_enrich before testing to see if the genes driving their enrichment are overlapping to a significant proportion using Fisher's exact test. To save computational time and to not complete this with an incredible number of permutations, scMappR stops at overlapping 5 cell-types.

### Value

coEnrich Enrichment of cell-types that are expressed by the same genes, up to 4 sets of cell-types.

### Examples

```
# load in signature matrices
data(POA_example)
POA_genes <- POA_example$POA_genes
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
sig <- get_gene_symbol(POA_Rank_signature)
Signature <- POA_Rank_signature
rownames(Signature) <- sig$rowname
genes <- rownames(Signature)[1:60]
heatmap_test <- tissue_scMappR_custom( genes, signature_matrix = Signature,
output_directory = "scMappR_test", toSave = FALSE)
group_preferences <- heatmap_test$group_celltype_preferences
```

---

compare\_deconvolution\_methods

*compare\_deconvolution\_methods*

---

### Description

This function calculates cell-type proportions of an inputted bulk sample using DeconRNA-seq, WGCNA, and DCQ methods. Outputted cell-type proportions are then compared.

**Usage**

```
compare_deconvolution_methods(
  count_file,
  signature_matrix,
  print_plot = FALSE,
  order_celltype = NULL
)
```

**Arguments**

count_file	Normalized (CPM, TPM, RPKM) RNA-seq count matrix where rows are gene symbols and columns are individuals. Either the object itself or the path of a .tsv file.
signature_matrix	Signature matrix (odds ratios) of cell-type specificity of genes. Either the object itself or a pathway to an .RData file containing an object named "wilcoxon_rank_mat_or" - generally internal.
print_plot	print the barplot of estimated cell-type proportions from each method into the R console (logical: TRUE/FALSE)
order_celltype	Specify the order that cell-type are placed on the barplot. NULL = alphabetical, otherwise a character vector of cell-type labels (i.e. column names of the signature matrix).

**Value**

List with the following elements:

cellWeighted_Foldchange	data frame of cellweightedFold-changes for each gene.
cellType_Proportions	data frame of cell-type proportions from DeconRNA-seq.
leave_one_out_proportions	data frame of average cell-type proportions for case and control when gene is removed.
processed_signature_matrix	signature matrix used in final analysis.

**Examples**

```
data(PBMC_example)
norm_counts <- PBMC_example$bulk_normalized
signature <- PBMC_example$odds_ratio_in
tst <- compare_deconvolution_methods(norm_counts, signature, FALSE,
  order_celltype = c("I_mono", "C_mono", "CD8_CM", "CD8_TE",
    "B_SM", "B_NSM", "B_naive"))
```

---

cwFoldChange\_evaluate *Measure cell-type specificity of cell-weighted Fold-changes*

---

## Description

This function normalizes cwFold-changes by each gene to help visualize the cell-type specificity of DEGs. It then tests if a cell-type has a large change in correlation from bulk DEGs. Finally, it identifies genes that may be specific to each cell-type.

## Usage

```
cwFoldChange_evaluate(  
  cwFC,  
  celltype_prop,  
  DEG_list,  
  gene_cutoff = NULL,  
  sd_cutoff = 3  
)
```

## Arguments

cwFC	A matrix or data frame of cell-weighted fold-changes of DEGs. Rows are DEGs and columns are cell-types.
celltype_prop	A matrix or data frame of cell-type proportions. Rows are different cell-types and columns are different samples. These cell-type proportions can come from any source (not just scMappR).
DEG_list	An object with the first column as gene symbols within the bulk dataset (doesn't have to be in signature matrix), second column is the adjusted p-value, and the third the log2FC path to a .tsv file containing this info is also acceptable.
gene_cutoff	Additional cut-off of normalized cwFold-change to see if a gene is cut-off.
sd_cutoff	Number of standard deviations or median absolute deviations to calculate outliers.

## Details

cwFold-changes are re-normalized and re-processed to interrogate cell-type specificity at the level of the cell-type and at the level of the gene. At the level of the cell-type, cwFold-changes are correlated to bulk DEGs. The difference in rank between bulk DEGs and cwFold-changes are also compared. At the level of the gene, cwFold-changes are re-normalized so that each gene sums to 1. Normalization of their distributions are tested with a Shapiro test. Then, outlier cell-types for each gene are measured by testing for 'sd\_cutoff's mad or sd's greater than the median or mean depending on if the cwFold-change is non-normally or normally distributed respectively. Cell-types considered outliers are then further filtered so their normalized cwFold-changes are greater than the cell-type proportions of that gene and 'gene\_cutoff' if the user sets it.

**Value**

List with the following elements:

- gene\_level\_investigation  
data frame of genes showing the Euclidian distances between cwFold-change and null vector as well as if cwFold-changes are distributed.
- celltype\_level\_investigation  
data frame of Spearman's and Pearson's correlation between bulk DEGs and cwFold-changes.
- cwFoldchange\_vs\_bulk\_rank\_change  
data frame of the change in rank of DEG between the bulk fold-change and cwFold-change.
- cwFoldChange\_normalized  
cwFold-change normalized such that each gene sums to 1.
- cwFoldchange\_gene\_assigned  
List of cell-types where genes are designated to cell-type specific differential expression.

**Examples**

```

data(PBMC_example)
bulk_DE_cors <- PBMC_example$bulk_DE_cors
bulk_normalized <- PBMC_example$bulk_normalized
odds_ratio_in <- PBMC_example$odds_ratio_in
case_grep <- "_female"
control_grep <- "_male"
max_proportion_change <- 10
print_plots <- FALSE
theSpecies <- "human"
toOut <- scMappR_and_pathway_analysis(bulk_normalized, odds_ratio_in,
                                     bulk_DE_cors, case_grep = case_grep,
                                     control_grep = control_grep, rda_path = "",
                                     max_proportion_change = 10, print_plots = TRUE,
                                     plot_names = "tst1", theSpecies = "human",
                                     output_directory = "tester",
                                     sig_matrix_size = 3000, up_and_downregulated = FALSE,
                                     internet = FALSE)

cwFC1 <- toOut$cellWeighted_Foldchange
prop1 <- toOut$cellType_Proportions
DE <- bulk_DE_cors
eval_test <- cwFoldChange_evaluate(cwFC1, prop1, DE)

```



---

DeconRNAseq\_CRAN      *DeconRNASeq CRAN compatible*

---

### Description

This function runs DeconRNAseq with default parameters such that it is compatible with CRAN and scMappR

### Usage

```
DeconRNAseq_CRAN(  
  datasets,  
  signatures,  
  proportions = NULL,  
  checksig = FALSE,  
  known.prop = FALSE,  
  use.scale = TRUE,  
  fig = FALSE  
)
```

### Arguments

datasets	Normalized RNA-seq dataset
signatures	Signature matrix of odds ratios
proportions	If cell-type proportion is already inputted - always NULL for scMappR
checksig	Check to see if plotting is significant - always false for scMappR
known.prop	If proportions were known - always false for scMappR
use.scale	Scale and center value - always TRUE for scMappR
fig	Make figures - always FALSE for scMappR

### Details

This is the exact same function as the primary function in the Bioconductor package, DeconRNAseq (PMID: 23428642) except it is now compatible with CRAN packages.

### Value

DeconRNAseq\_CRAN Estimated cell-type proportions with DeconRNAseq.

## Examples

```
data(PBMC_example)
bulk_DE_cors <- PBMC_example$bulk_DE_cors
bulk_normalized <- PBMC_example$bulk_normalized
odds_ratio_in <- PBMC_example$odds_ratio_in
out <- DeconRNAseq_CRAN(as.data.frame(bulk_normalized), as.data.frame(odds_ratio_in))
```

---

deconvolute\_and\_contextualize

*Generate cell weighted Fold-Changes (cwFold-changes)*

---

## Description

This function takes a count matrix, signature matrix, and differentially expressed genes (DEGs) before generating cwFold-changes for each cell-type.

## Usage

```
deconvolute_and_contextualize(
  count_file,
  signature_matrix,
  DEG_list,
  case_grep,
  control_grep,
  max_proportion_change = -9,
  print_plots = T,
  plot_names = "scMappR",
  theSpecies = "human",
  FC_coef = T,
  sig_matrix_size = 3000,
  drop_unknown_celltype = TRUE,
  toSave = FALSE,
  path = NULL,
  deconMethod = "DeconRNASeq"
)
```

## Arguments

**count\_file** Normalized (e.g. CPM, TPM, RPKM) RNA-seq count matrix where rows are gene symbols and columns are individuals. Either the matrix itself of class "matrix" or "data.frame" or a path to a tsv file containing these DEGs. The gene symbols in the count file, signature matrix, and DEG list must match.

signature_matrix	Signature matrix (fold-change ratios) of cell-type specificity of genes. Either the object itself or a pathway to an .RData file containing an object named "wilcoxon_rank_mat_or". We strongly recommend inputting the signature matrix directly.
DEG_list	An object with the first column as gene symbols within the bulk dataset (doesn't have to be in signature matrix), second column is the adjusted P-value, and the third the log2FC. Path to a tsv file containing this info is also acceptable.
case_grep	Tag in the column name for cases (i.e. samples representing upregulated) OR an index of cases.
control_grep	Tag in the column name for control (i.e. samples representing downregulated) OR an index of cases.
max_proportion_change	Maximum cell-type proportion change. May be useful if a cell-type does not exist in one condition, thus preventing infinite values.
print_plots	Whether boxplots of the estimated CT proportion for the leave-one-out method of CT deconvolution should be printed (T/F).
plot_names	If plots are being printed, the pre-fix of their .pdf files.
theSpecies	internal species designation to be passed from 'scMappR_and_pathway_analysis'. It only impacts this function if data are taken directly from the PanglaoDB database (i.e. not reprocessed by scMappR or the user).
FC_coef	Making cwFold-changes based on fold-change (TRUE) or rank := (-log10(Pval)) (FALSE) rank. After testing, we strongly recommend to keep true (T/F).
sig_matrix_size	Number of genes in signature matrix for cell-type deconvolution.
drop_unknown_celltype	Whether or not to remove "unknown" cell-types from the signature matrix (T/F).
toSave	Allow scMappR to write files in the current directory (T/F).
path	If toSave == TRUE, path to the directory where files will be saved.
deconMethod	Which RNA-seq deconvolution method to use to estimate cell-type proportions. Options are "WGCNA", "DCQ", or "DeconRNAseq"

## Details

This function completes the pre-processing, normalization, and scaling steps in the scMappR algorithm before calculating cwFold-changes. cwFold-changes scales bulk fold-changes by the cell-type specificity of the gene, cell-type gene-normalized cell-type proportions, and the reciprocal ratio of cell-type proportions between the two conditions. cwFold-changes are generated for genes that are in both the count matrix and in the list of DEGs. It does not have to also be in the signature matrix. First, this function will estimate cell-type proportions with all genes included before estimating changes in cell-type proportion between case/control using a t-test. Then, it takes a leave-one-out approach to cell-type deconvolution such that estimated cell-type proportions are computed for every inputted DEG. Optionally, the differences between cell-type proportions before and after a gene is removed is plotted in boxplots. Then, for every gene, cwFold-changes are computed with the following formula (the example for upregulated genes)  $val <- cell\_preferences * cell\_type\_proportion * cell\_type\_proportion\_fold\_change * sign * 2^{abs(gene\_DE\$log2fc)}$ . A matrix of cwFold-changes for all DEGs are returned.

**Value**

List with the following elements:

cellWeighted\_Foldchange  
                                   data frame of cellweightedFold changes for each gene.

cellType\_Proportions  
                                   data frame of cell-type proportions from DeconRNA-seq.

leave\_one\_out\_proportions  
                                   data frame of average cell-type proportions for case and control when gene is removed.

processed\_signature\_matrix  
                                   signature matrix used in final analysis.

**Examples**

```
data(PBMC_example)
bulk_DE_cors <- PBMC_example$bulk_DE_cors
bulk_normalized <- PBMC_example$bulk_normalized
odds_ratio_in <- PBMC_example$odds_ratio_in
case_grep <- "_female"
control_grep <- "_male"
max_proportion_change <- 10
print_plots <- FALSE
theSpecies <- "human"
norm <- deconvolute_and_contextualize(bulk_normalized, odds_ratio_in, bulk_DE_cors,
                                     case_grep = case_grep, control_grep = control_grep,
                                     max_proportion_change = max_proportion_change,
                                     print_plots = print_plots,
                                     theSpecies = theSpecies, toSave = FALSE)
```

---

extract\_genes\_cell      *Extract Markers*

---

**Description**

Extracting cell-type markers from a signature matrix.

**Usage**

```
extract_genes_cell(
  geneHeat,
  cellTypes = "ALL",
  val = 1,
  isMax = FALSE,
  isPvalue = FALSE
)
```

**Arguments**

geneHeat	The heatmap of ranks from your scRNA-seq dataset with your genes subsetted.
cellTypes	The cell-types that you're interested in extracting. They need to be colnames (not case sensitive).
val	How associated a gene is with a particular cell type to include in your list - default is slightly associated.
isMax	If you are taking the single best CT marker (T/F) – TRUE not recommended.
isPvalue	If the signature matrix is raw p-value (T/F) – TRUE not recommended.

**Details**

This function takes a signature matrix and extracts cell-type markers above a p-value or fold-change threshold.

**Value**

extract\_genes\_cell A vector of genes above the threshold for each sample.

**Examples**

```
data(POA_example)
Signature <- POA_example$POA_Rank_signature
RowName <- get_gene_symbol(Signature)
rownames(Signature) <- RowName$rowname
# extract genes with a -log10(Padj > 1)
Signat <- extract_genes_cell(Signature)
```

---

genes\_to\_heatmap      *Generate signature matrix*

---

**Description**

Convert a list of cell-type markers from FindMarkers in Seurat to a signature matrix defined by odds ratio and rank.

**Usage**

```
genes_to_heatmap(
  genes,
  species = "human",
  naming_preference = -9,
  rda_path = "",
  make_names = TRUE,
  internal = FALSE
)
```

**Arguments**

genes	A list of cell-type markers with fold-changes and p-values (FindMarkers output in Seurat).
species	The species of gene symbols, if not internal, "human" or "mouse".
naming_preference	Likely cell-types given tissues (to be passed into human_mouse_ct_marker_enrich).
rda_path	Path to output directory, if toSave is true.
make_names	Identify names of cell-type markers using the Fisher's exact test method (T/F).
internal	If this function is pre-processing from Panglao (T/F).

**Details**

Take a list of compiled differentially expressed genes from different cell-types, identify what the cell-types are using the Fisher's exact test, and then convert into a signature matrix for both the adjusted p-value and odds ratio.

**Value**

List with the following elements:

pVal	A dataframe containing the signature matrix of ranks ( $-\log_{10}(\text{Padj}) * \text{sign}(\text{fold-change})$ ).
OR	A dataframe containing the signature matrix of odds ratios.
cellname	A vector of the cell-labels returned from the GSVA method.
topGenes	the top 30 most expressed genes in each cell-type.

**Examples**

```
data(POA_example)
POA_genes <- POA_example$POA_genes
signature <- genes_to_heatmap(POA_genes, species = -9, make_names = FALSE)
```

---

get_gene_symbol	<i>Internal – get gene symbol from Panglao.db assigned gene-names (symbol-ensembl).</i>
-----------------	---

---

## Description

Internal – removes Ensembl signature appended to signature matrix from Panglao and figure out species by pre-fix Ensembl of the Ensembl ID that is appended to gene names.

## Usage

```
get_gene_symbol(wilcoxon_rank_mat_t)
```

## Arguments

wilcoxon\_rank\_mat\_t  
Matrix where row names are "GeneSymbol-Ensembl" (human or mouse).

## Details

Internal: This function removes the ENGMUS/ENGS tag from Panglao created gene names (symbol-ENGS). From the ENSG/ENSMUS, this function determines if the species is mouse/human and returns the gene symbols.

## Value

List with the following elements:

rowname	Genes in the signature matrix excluding the ensemble name.
species	"mouse" or "human" depending on appended ensembl symbols.

## Examples

```
# load signature
data(POA_example)
POA_OR_signature <- POA_example$POA_OR_signature
symbols <- get_gene_symbol(POA_OR_signature)
```

---

`get_signature_matrices`*Get signature matrices.*

---

**Description**

This function downloads and returns signature matrices and associated cell-type labels from the scMappR\_data repo.

**Usage**

```
get_signature_matrices(type = "all")
```

**Arguments**

`type` a character vector that can be 'all', 'pVal', or 'OR'

**Value**

`get_signature_matrices` Returns the signature matrices currently stored in scMappR\_Data. Associated cell-type labels from different methods for each signature matrix is also provided.

**Examples**

```
signatures <- get_signature_matrices(type = "all")
```

---

`gmt`*gmt\_example*

---

**Description**

Markers of 5 glial cell-types

**Usage**

```
data(gmt)
```



**Format**

A list with 5 character vectors, each containing genes.

**Astrocytes\_panglao** astrocyte markers identified by panglao

**Schwann\_panglao** Schwann markers identified by panglao

**Bergmann glia\_panglao** Bergmann glia markers identified by panglao

**Kupffer\_panglao** Kupffer markers identified by panglao

**Oligodendrocyte progenitor\_panglao** Oligodendrocyte progenitor markers identified by panglao

**Details**

A named list containing the cell-type markers of 5 glial cell types. Used for testing cell-type naming functions.

**Examples**

```
data(gmt)
```

---

```
gProfiler_cellWeighted_Foldchange
```

*Pathway enrichment for cwFold-changes*

---

**Description**

This function runs through each list of cell weighted Fold changes (cwFold-changes) and completes both pathway and transcription factor (TF) enrichment.

**Usage**

```
gProfiler_cellWeighted_Foldchange(  
  cellWeighted_Foldchange_matrix,  
  species,  
  background,  
  gene_cut,  
  newGprofiler  
)
```

**Arguments**

cellWeighted_Foldchange_matrix	Matrix of cell weighted Fold changes from the deconvolute_and_contextualize functions.
species	Human, mouse, or a name that is compatible with gProfileR (e.g. "mmusculus").
background	A list of background genes to test against.
gene_cut	The top number of genes in pathway analysis.
newGprofiler	Using gProfileR or gprofiler2, (T/F).

**Details**

This function takes a matrix of `cellWeighted_Foldchange` and a species (human, mouse, or a character directly compatible with `g:ProfileR`). Before completing pathway analysis with `g:ProfileR`. Enriched pathways are stored in a list and returned.

**Value**

List with the following elements:

BP	gprofiler enrichment of biological pathways for each cell-type
TF	gprofiler enrichment of transcription factors for each cell-type.

**Examples**

```
data(PBMC_example)

bulk_DE_cors <- PBMC_example$bulk_DE_cors
bulk_normalized <- PBMC_example$bulk_normalized
odds_ratio_in <- PBMC_example$odds_ratio_in

case_grep <- "_female"
control_grep <- "_male"
max_proportion_change <- 10
print_plots <- FALSE
theSpecies <- "human"
norm <- deconvolute_and_contextualize(bulk_normalized, odds_ratio_in,
                                     bulk_DE_cors, case_grep = case_grep,
                                     control_grep = control_grep,
                                     max_proportion_change = max_proportion_change,
                                     print_plots = print_plots,
                                     theSpecies = theSpecies)

background = rownames(bulk_normalized)
STVs <- gProfiler_cellWeighted_Foldchange(norm$cellWeighted_Foldchange, theSpecies,
    background, gene_cut = -9, newGprofiler = FALSE)
```

---

gsva\_cellIdentify      *Cell-type naming with GSVA*

---

**Description**

This function computes the mean expression of every cell-type before predicting the most likely cell-type using the GSVA method.

**Usage**

```
gsva_cellIdentify(  
  pbmc,  
  theSpecies,  
  naming_preference = -9,  
  rda_path = "",  
  toSave = FALSE  
)
```

**Arguments**

pbmc	Processed Seurat object without named cells.
theSpecies	"human" or "mouse" – it will determine which species cell-type markers will originate from.
naming_preference	Once top cell-type markers are identified, naming_preferences will then extract CT markers within a more appropriate tissue type.
rda_path	Path to pre-computed cell-type .gmt files (rda objects).
toSave	If scMappR is allowed to write files and directories.

**Details**

This function inputs a Seurat object and uses the average normalized expression of each gene in each cluster to identify cell-types using the GSVA method.

**Value**

List with the following elements:

cellMarker	Most likely cell-types predicted from CellMarker database.
panglao	Most likely cell-types predicted from Panglao database.
avg_expression	Average expression of each gene in each cell-type.

**Examples**

```
data(sm)  
toProcess <- list(example = sm)  
tst1 <- process_from_count(toProcess, "testProcess")  
cellnames <- gsva_cellIdentify(tst1, theSpecies = "mouse",  
  naming_preference = "brain", rda_path = "")
```

---

heatmap\_generation      *Generate Heatmap*


---

### Description

This function takes an inputted signature matrix as well as a list of genes and overlaps them. Then, if there is overlap, it prints a heatmap or barplot (depending on the number of overlapping genes). Then, for every cell-type, genes considered over-represented are saved in a list.

### Usage

```
heatmap_generation(
  genesIn,
  comp,
  reference,
  cex = 0.8,
  rd_path = "",
  cellTypes = "ALL",
  pVal = 0.01,
  isPval = TRUE,
  isMax = FALSE,
  isBackground = FALSE,
  which_species = "human",
  toSave = FALSE,
  path = NULL
)
```

### Arguments

genesIn	A list of gene symbols (all caps) to have their cell type enrichment.
comp	The name of the comparison.
reference	Path to signature matrix or the signature matrix itself.
cex	The size of the genes in the column label for the heatmap.
rd_path	The directory to RData files – if they are not in this directory, then the files will be downloaded.
cellTypes	Colnames of the cell-types you will extract (passed to <code>extract_genes_cell</code> ).
pVal	The level of association a gene has within a cell type (passed to <code>extract_genes_cell</code> ).
isPval	If the signature matrix is raw p-value (T/F) – TRUE not recommended
isMax	If you are taking the single best CT marker (T/F) – TRUE not recommended
isBackground	If the heatmap is from the entire signature matrix or just the inputted gene list (T/F). <code>isBackground == TRUE</code> is used for internal.
which_species	Species of gene symbols – "human" or "mouse" .
toSave	Allow <code>scMappR</code> to write files in the path directory (T/F).
path	If <code>toSave == TRUE</code> , path to the directory where files will be saved.

**Value**

List with the following elements:

genesIn	Vector of genes intersecting gene list and signature matrix.
genesNoIn	Vector of inputted genes not in signature matrix.
geneHeat	Signature matrix subsetted by inputted gene list
preferences	Cell-markers mapping to cell-types.

**Examples**

```
# load in signature matrices
data(POA_example)
POA_genes <- POA_example$POA_genes
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
Signature <- POA_Rank_signature
rowname <- get_gene_symbol(Signature)
rownames(Signature) <- rowname$rowname
genes <- rownames(Signature)[1:100]
heatmap_test <- heatmap_generation(genesIn = genes, "scMappR_test",
                                  reference = Signature, which_species = "mouse")
```

---

```
human_mouse_ct_marker_enrich
      Consensus cell-type naming (Fisher's Exact)
```

---

**Description**

This function completes the Fisher's exact test cell-type naming for all cell-types.

**Usage**

```
human_mouse_ct_marker_enrich(
  gene_lists,
  theSpecies = "human",
  cell_marker_path = "",
  naming_preference = -9
)
```

**Arguments**

gene_lists	A named list of vectors containing cell-type markers (mouse or human gene-symbols) which will be named as a cell-type via the Fisher's exact test method.
theSpecies	The species of the gene symbols: "human" or "mouse".

cell\_marker\_path

If local, path to cell-type marker rda files, otherwise, we will try to download data files.

naming\_preference

Either -9 if there is no expected cell-type or one of the categories from `get_naming_preference_options()`. This is useful if you previously have an idea of which cell-type you were going to enrich.

## Details

Fisher's exact test method of cell-type identification using the Panglao and CellMarker databases. It extracts significant pathways ( $pFDR < 0.05$ ). Then, if `naming_preference != -9`, it will extract the enriched cell-types within the cell-types identified with the naming preferences option. Generally, this method seems to be biased to cell-types with a greater number of markers.

## Value

List with the following elements:

`cellTypes`        most likely marker for each cell-type from each database.  
`marker_sets`     all enriched cell-types for each cluster from each dataset.

## Examples

```
data(POA_example)
POA_geneses <- POA_example$POA_geneses
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
Signature <- POA_Rank_signature
rowname <- get_gene_symbol(Signature)
rownames(Signature) <- rowname$rowname
genes <- rownames(Signature)[1:100]
enriched <- human_mouse_ct_marker_enrich(gene_lists = genes, theSpecies = "mouse",
                                         cell_marker_path = "", naming_preference = "brain")
```

---

make\_TF\_barplot        *Plot g:profileR Barplot (TF)*

---

## Description

Make a barplot of the top transcription factors enriched by `gprofileR`.

## Usage

```
make_TF_barplot(ordered_back_all_tf, top_tf = 5)
```

**Arguments**

ordered\_back\_all\_tf  
Output of the g:profileR function.

top\_tf  
The number of transcription factors to be plotted.

**Details**

This function takes a gprofileR output and prints the top "top\_tfs" most significantly enriched fdr adjusted p-values before plotting the rank of their p-values.

**Value**

make\_TF\_barplot A barplot of the number of "top\_tf" tf names (not motifs), ranked by  $-\log_{10}(\text{Pfd})$ .

**Examples**

```
data(POA_example)
POA_generes <- POA_example$POA_generes
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
Signature <- as.data.frame(POA_Rank_signature)
rowname <- get_gene_symbol(Signature)
rownames(Signature) <- rowname$rowname
ordered_back_all <- gprofiler2::gost(query = rowname$rowname[1:100], organism = "mmusculus",
ordered_query = TRUE, significant = TRUE, exclude_iea = FALSE, multi_query = FALSE,
measure_underrepresentation = FALSE, evcodes = FALSE, user_threshold = 0.05,
correction_method = "fdr", numeric_ns = "", sources = c("GO:BP", "KEGG", "REAC"))
ordered_back_all <- ordered_back_all$result
ordered_back_all <- ordered_back_all[ordered_back_all$term_size > 15 &
ordered_back_all$term_size < 2000 & ordered_back_all$intersection_size > 2,]
ordered_back_all_tf <- gprofiler2::gost(query = rowname$rowname[1:150], organism = "mmusculus",
ordered_query = TRUE, significant = TRUE, exclude_iea = FALSE, multi_query = FALSE,
measure_underrepresentation = FALSE, evcodes = FALSE, user_threshold = 0.05,
correction_method = "fdr", numeric_ns = "", sources = c("TF"))
ordered_back_all_tf <- ordered_back_all_tf$result
ordered_back_all_tf <- ordered_back_all_tf[ordered_back_all_tf$term_size > 15
& ordered_back_all_tf$term_size < 5000 & ordered_back_all_tf$intersection_size > 2,]
TF = ordered_back_all_tf
BP <- ordered_back_all
bp <- plotBP(BP)
tf <- make_TF_barplot(TF)
```

---

pathway\_enrich\_internal

*Internal - Pathway enrichment for cellWeighted\_Foldchanges and bulk gene list*

---

### Description

This function completes pathway enrichment of cellWeighted\_Foldchanges and bulk gene list.

### Usage

```
pathway_enrich_internal(
  DEGs,
  theSpecies,
  scMappR_vals,
  background_genes,
  output_directory,
  plot_names,
  number_genes = -9,
  newGprofiler = FALSE,
  toSave = FALSE,
  path = NULL
)
```

### Arguments

DEGs	Differentially expressed genes (gene_name, padj, log2fc).
theSpecies	Human, mouse, or a character that is compatible with g:ProfileR.
scMappR_vals	cell weighted Fold-changes of differentially expressed genes.
background_genes	A list of background genes to test against.
output_directory	Path to the directory where files will be saved.
plot_names	Names of output.
number_genes	Number of genes to if there are many, many DEGs.
newGprofiler	Whether to use g:ProfileR or gprofiler2 (T/F).
toSave	Allow scMappR to write files in the current directory (T/F).
path	If toSave == TRUE, path to the directory where files will be saved.

### Details

Internal: Pathway analysis of differentially expressed genes (DEGs) and cell weighted Fold-changes (cellWeighted\_Foldchanges) for each cell-type. Returns .RData objects of differential analysis as well as plots of the top bulk pathways. It is a wrapper for making barplots, bulk pathway analysis, and gProfiler\_cellWeighted\_Foldchange.



**Value**

List with the following elements:

BPs                    Enriched biological pathways for each cell-type.  
 TFs                    Enriched transcription factors for each cell-type.

**Examples**

```
data(PBMC_example)
bulk_DE_cors <- PBMC_example$bulk_DE_cors
bulk_normalized <- PBMC_example$bulk_normalized
odds_ratio_in <- PBMC_example$odds_ratio_in
case_grep <- "_female"
control_grep <- "_male"
max_proportion_change <- 10
print_plots <- FALSE
theSpecies <- "human"
toOut <- scMappR_and_pathway_analysis(bulk_normalized, odds_ratio_in,
                                     bulk_DE_cors, case_grep = case_grep,
                                     control_grep = control_grep, rda_path = "",
                                     max_proportion_change = 10, print_plots = TRUE,
                                     plot_names = "tst1", theSpecies = "human",
                                     output_directory = "tester",
                                     sig_matrix_size = 3000, up_and_downregulated = FALSE,
                                     internet = FALSE)
```

---

 PBMC\_example

*PBMC\_scMappR*


---

**Description**

Toy example of data where cell-weighted fold-changes and related downstream analyses can be completed.

**Usage**

```
data(PBMC_example)
```

**Format**

A list containing three data frames, normalized count data, a signature matrix, and a list of differentially expressed genes.

**bulk\_normalized** A 3231 x 9 matrix where rows are genes, columns are samples, and the matrix is filled with CPM normalized counts.

**odds\_ratio\_in** A 2336 x 7 matrix where rows are genes, columns are cell-types and matrix is filled with the odds-ratio that a gene is in each cell-type.

**bulk\_DE\_cors** A 59 x 3 matrix of sex-specific genes found between male and female PBMC samples (female biased = upregulated). row and rownames are genes, columns are gene name, FDR adjusted p-value, and log2 fold-change. DEGs were computed with DESeq2 and genes with a log2FC > 1 were kept.

### Details

A named list called "PBMC\_example" containing the count data, signature matrix, and DEGs. The count data and signature matrix are shortened to fit the size of the package and do not reflect biologically relevant data.

### Examples

```
data(PBMC_example)
```

---

plotBP

*Plot gProfileR Barplot*

---

### Description

Make a barplot of the top biological factors enriched by g:ProfileR.

### Usage

```
plotBP(ordered_back_all, top_bp = 10)
```

### Arguments

`ordered_back_all` Output of the g:ProfileR function.

`top_bp` The number of pathways you want to plot.

### Details

This function takes a gProfileR output and prints the top "top\_bp" most significantly enriched FDR adjusted p-values before plotting the rank of their p-values.

### Value

plotBP A barplot of the number of "top\_bp" pathways, ranked by  $-\log_{10}(P_{fdr})$ .

## Examples

```

data(POA_example)
POA_generes <- POA_example$POA_generes
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
Signature <- as.data.frame(POA_Rank_signature)
rowname <- get_gene_symbol(Signature)
rownames(Signature) <- rowname$rowname
ordered_back_all <- gprofiler2::gost(query = rowname$rowname[1:100], organism = "mmusculus",
  ordered_query = TRUE, significant = TRUE, exclude_iea = FALSE, multi_query = FALSE,
  measure_underrepresentation = FALSE, ecodes = FALSE, user_threshold = 0.05,
  correction_method = "fdr", numeric_ns = "", sources = c("GO:BP", "KEGG", "REAC"))
ordered_back_all <- ordered_back_all$result
ordered_back_all <- ordered_back_all[ordered_back_all$term_size > 15
  & ordered_back_all$term_size < 2000 & ordered_back_all$intersection_size > 2,]
ordered_back_all_tf <- gprofiler2::gost(query = rowname$rowname[1:150], organism = "mmusculus",
  ordered_query = TRUE, significant = TRUE, exclude_iea = FALSE, multi_query = FALSE,
  measure_underrepresentation = FALSE, ecodes = FALSE, user_threshold = 0.05,
  correction_method = "fdr", numeric_ns = "", sources = c("TF"))
ordered_back_all_tf <- ordered_back_all_tf$result
ordered_back_all_tf <- ordered_back_all_tf[ordered_back_all_tf$term_size > 15
  & ordered_back_all_tf$term_size < 5000 & ordered_back_all_tf$intersection_size > 2,]
TF = ordered_back_all_tf
BP <- ordered_back_all
bp <- plotBP(BP)
tf <- make_TF_barplot(TF)

```

---

POA\_example

*Preoptic\_Area*

---

## Description

Toy data for tissue\_scMappR\_custom, tissue\_scMappR\_internal, generes\_to\_heatmap.

## Usage

```
data(POA_example)
```

## Format

A list containing three objects: summary statistics of cell-type markers, a signature matrix of odds ratios, and a signature matrix of ranks.

**POA\_generes** A list of 27 data frames containing (up to 30) cell-type markers. Each element of the list is a dataframe where rows are genes, and columns are p-value, log<sub>2</sub>FC, percentage of cells expressing gene in cell-type, percentage of cells expressing gene in other cell-types, and FDR adjusted p-value.

**POA\_OR\_signature** A 266 x 27 matrix where rows are genes, columns are cell-types and matrix is filled with the odds-ratio that a gene is in each cell-type.

**POA\_Rank\_signature** A 266 x 27 matrix of matrix where rows are genes, columns are cell-types and matrix is filled with the rank :=  $-\log_{10}(P_{\text{fdr}})$  that a gene is in each cell-type.

### Details

A named list called POA\_example (pre-optic area example) containing three objects, POA\_genes: a list of truncated dataframes containing summary statistics for each cell-type marker, POA\_OR\_signature a truncated signature matrix of odds ratio's for cell-types in the POA, and POA\_Rank\_signature a truncated signature matrix of  $-\log_{10}(P_{\text{adj}})$  for cell-type markers in the POA.

### Examples

```
data(POA_example)
```

---

```
process_dgTMatrix_lists
      Count Matrix To Signature Matrix
```

---

### Description

This function takes a list of count matrices, processes them, calls cell-types, and generates signature matrices.

### Usage

```
process_dgTMatrix_lists(
  dgTMatrix_list,
  name,
  species_name,
  naming_preference = -9,
  rda_path = "",
  panglao_set = FALSE,
  haveUMAP = FALSE,
  saveSCObject = FALSE,
  internal = FALSE,
  toSave = FALSE,
  path = NULL,
  use_sctransform = FALSE,
  test_ctname = "wilcox",
  genes_integrate = 2000,
  genes_include = FALSE
)
```

**Arguments**

dgTMatrix_list	A list of matrices in the class of dgTMatrix object – sparse object – compatible with Seurat rownames should be of the same species for each.
name	The name of the outputted signature matrices, cell-type preferences, and Seurat objects if you choose to save them.
species_name	Mouse or human symbols, -9 if internal as Panglao objects have gene symbol and ensembl combined.
naming_preference	For cell-type naming, see if cell-types given the inputted tissues are more likely to be named within one of the categories. These categories are: "brain", "epithelial", "endothelial", "blood", "connective", "eye", "epidermis", "Digestive", "Immune", "pancreas", "liver", "reproductive", "kidney", "respiratory".
rda_path	If saved, directory to where data from scMappR_data is downloaded.
panglao_set	If the inputted matrices are from Panglao (i.e. if they're internal).
haveUMAP	Save the UMAPs - requires additional packages (see Seurat for details).
saveSCObject	Save the Seurat object as an RData object (T/F).
internal	Was this used as part of the internal processing of Panglao datasets (T/F).
toSave	Allow scMappR to write files in the current directory (T/F)
path	If toSave == TRUE, path to the directory where files will be saved.
use_sctransform	If you should use sctransform or the Normalize/VariableFeatures/ScaleData pipeline (T/F).
test_ctname	statistical test for calling CT markers – must be in Seurat
genes_integrate	The number of genes to include in the integration anchors feature when combining datasets.
genes_include	TRUE or FALSE – include 2000 genes in signature matrix or all matrix.

**Details**

This function is a one line wrapper to process count matrices into a signature matrix. It combines process\_from\_count, two methods of identifying cell-type identities (GSVA and Fisher's test). Then, it takes the output of cell-type markers and converts it into a signature matrix of p-value ranks and odds ratios. It saves the Seurat object (if chosen with saveSCObject), cell-type identities from GSVA (its own object), and the signature matrices. Cell-type marker outputs are also saved in the genes.RData list. This is a list of cell-types containing all of the cell-type markers found with the FindMarkers function. Names of the genes lists and the signature matrices are kept.

**Value**

List with the following elements:

wilcoxon_rank_mat_t	A dataframe containing the signature matrix of ranks ( $-\log_{10}(\text{Padj}) * \text{sign}(\text{fold-change})$ ).
---------------------	---

wilcoxon\_rank\_mat\_or  
A dataframe containing the signature matrix of odds-ratios.

genes  
All cell-type markers for each cell-type with p-value and fold changes.

## Examples

```
data(sm)
toProcess <- list(example = sm)
tst1 <- process_dgTMatrix_lists(toProcess, name = "testPropcess", species_name = -9,
  naming_preference = "eye", rda_path = "")
```

---

process\_from\_count      *Count Matrix To Seurat Object*

---

## Description

This function processes a list of count matrices (same species/gene symbols in each list) and converts them to a Seurat object.

## Usage

```
process_from_count(
  countmat_list,
  name,
  theSpecies = -9,
  haveUmap = FALSE,
  saveALL = FALSE,
  panglao_set = FALSE,
  toSave = FALSE,
  path = NULL,
  use_sctransform = FALSE,
  genes_integrate = 2000,
  genes_include = FALSE
)
```

## Arguments

countmat\_list      A list of count matrices that will be integrated using the IntegrationAnchors features they should have the same rownames. A dgCMatix or matrix object is also acceptable, and no samples will be integrated.

name                The output of the normalized and fused Seurat object if you choose to keep it.

theSpecies         Gene symbols for human, mouse, or -9 if internal. If your species is not human or mouse gene symbols, make sure that you have "MT-" before your mitochondrial gene names then pick "human".

haveUmap	Write a UMAP (T/F).
saveALL	Save the Seurat object generated (T/F).
panglao_set	If the function is being used from internal (T/F).
toSave	Allows scMappR to print files and make directories locally (T/F).
path	If toSave == TRUE, path to the directory where files will be saved.
use_sctransform	If you should use sctransform or the Normalize/VariableFeatures/ScaleData pipeline (T/F).
genes_integrate	The number of genes to include in the integration anchors feature when combining datasets
genes_include	TRUE or FALSE – include 2000 genes in signature matrix or all matrix.

### Details

This function takes a list of count matrices and returns a Seurat object of the count matrices integrated using Seurat v3 (and IntegrationAnchors feature). Different normalization features such as the SCTransform pipeline are also available in this function. Different options are used when the function is being ran internally (i.e. reprocessing count matrices from PanglaoDB) or if it is running from custom scRNA-seq data. Larger scRNA-seq datasets can take considerable amounts of memory and run-time. See Seurat for details.

### Value

process\_from\_count A processed and integrated Seurat object that has been scaled and clustered. It can be returned as an internal object or also stored as an RData object if necessary.

### Examples

```
data(sm)
toProcess <- list(example = sm)
tst1 <- process_from_count(toProcess, "testProcess")
```

---

scMappR\_and\_pathway\_analysis

*Generate cellWeighted\_Foldchanges, visualize, and enrich.*

---

### Description

This function generates cell weighted Fold-changes (cellWeighted\_Foldchange), visualizes them in a heatmap, and completes pathway enrichment of cellWeighted\_Foldchanges and the bulk gene list using g:ProfileR.

**Usage**

```

scMappR_and_pathway_analysis(
  count_file,
  signature_matrix,
  DEG_list,
  case_grep,
  control_grep,
  rda_path = "",
  max_proportion_change = -9,
  print_plots = T,
  plot_names = "scMappR",
  theSpecies = "human",
  output_directory = "scMappR_analysis",
  sig_matrix_size = 3000,
  drop_unknown_celltype = TRUE,
  internet = TRUE,
  up_and_downregulated = FALSE,
  gene_label_size = 0.4,
  number_genes = -9,
  toSave = FALSE,
  newGprofiler = FALSE,
  path = NULL,
  deconMethod = "DeconRNASeq"
)

```

**Arguments**

<code>count_file</code>	Normalized (i.e. TPM, RPKM, CPM) RNA-seq count matrix where rows are gene symbols and columns are individuals. Inputted data should be a data.frame or matrix. A character vector to a tsv file where this data can be loaded is also acceptable. Gene symbols from the count file, signature matrix, and DEG list should all match (case sensitive, gene symbol or ensembl, etc.)
<code>signature_matrix</code>	Signature matrix: a gene by cell-type matrix populated with the fold-change of gene expression in cell-type marker "i" vs all other cell-types. Object should be a data.frame or matrix.
<code>DEG_list</code>	An object with the first column as gene symbols within the bulk dataset (doesn't have to be in signature matrix), second column is the adjusted p-value, and the third the log2FC path to a .tsv file containing this info is also acceptable.
<code>case_grep</code>	A character representing what designates the "cases" (i.e. upregulated is 'case' biased) in the columns of the count file. A numeric vector of the index of "cases" is also acceptable. Tag in the column name for cases (i.e. samples representing upregulated) OR an index of cases.
<code>control_grep</code>	A character representing what designates the "control" (i.e. downregulated is 'control biased) in the columns of the count file. A numeric vector of the index of "control" is also acceptable. Tag in the column name for cases (i.e. samples representing upregulated) OR an index of cases.



rda_path	If downloaded, path to where data from scMappR_data is stored.
max_proportion_change	Maximum cell-type proportion change – may be useful if there are many rare cell-type. Alternatively, if a cell-type is only present in one condition but not the other, it will prevent possible infinite or 0 cwFold-changes.
print_plots	Whether boxplots of the estimated CT proportion for the leave-one-out method of CT deconvolution should be printed. The same name of the plots will be completed for top pathways.
plot_names	The prefix of plot pdf files.
theSpecies	human, mouse, or a species directly compatible with gProfileR (i.e. g:ProfileR).
output_directory	The name of the directory that will contain output of the analysis.
sig_matrix_size	Maximum number of genes in signature matrix for cell-type deconvolution.
drop_unknown_celltype	Whether or not to remove "unknown" cell-types from the signature matrix.
internet	Whether you have stable Wifi (T/F).
up_and_downregulated	Whether you are additionally splitting up/downregulated genes (T/F).
gene_label_size	The size of the gene label on the plot.
number_genes	The number of genes to cut-off for pathway analysis (good with many DEGs).
toSave	Allow scMappR to write files in the current directory (T/F).
newGprofiler	Whether to use gProfileR or gprofiler2 (T/F).
path	If toSave == TRUE, path to the directory where files will be saved.
deconMethod	Which RNA-seq deconvolution method to use to estimate cell-type proportions. Options are "WGCNA", "DCQ", or "DeconRNAseq"

## Details

This function generates cellWeighted\_Foldchanges for every cell-type (see deconvolute\_and\_contextualize), as well as accompanying data such as cell-type proportions with the DeconRNA-seq, WGCNA, or DCQ methods. Then, it generates heatmaps of all cellWeighted\_Foldchanges, cellWeighted\_Foldchanges overlapping with the signature matrix, the entire signature matrix, the cell-type preference values from the signature matrix that overlap with inputted differentially expressed genes. Then, assuming there is available internet, it will complete gProfileR of the reordered cellWeighted\_Foldchanges as well as a the ordered list of genes. This function is a wrapper for deconvolute\_and\_contextualize and pathway\_enrich\_internal and the primary function within the package.

## Value

List with the following elements:

cellWeighted_Foldchanges	Cellweighted Fold-changes for all differentially expressed genes.
paths	Enriched biological pathways for each cell-type.
TFs	Enriched TFs for each cell-type.

## Examples

```

data(PBMC_example)
bulk_DE_cors <- PBMC_example$bulk_DE_cors
bulk_normalized <- PBMC_example$bulk_normalized
odds_ratio_in <- PBMC_example$odds_ratio_in
case_grep <- "_female"
control_grep <- "_male"
max_proportion_change <- 10
print_plots <- FALSE
theSpecies <- "human"
toOut <- scMappR_and_pathway_analysis(bulk_normalized, odds_ratio_in,
                                     bulk_DE_cors, case_grep = case_grep,
                                     control_grep = control_grep, rda_path = "",
                                     max_proportion_change = 10, print_plots = TRUE,
                                     plot_names = "tst1", theSpecies = "human",
                                     output_directory = "tester",
                                     sig_matrix_size = 3000, up_and_downregulated = FALSE,
                                     internet = FALSE)

```

---

scMappR\_tissues

*scMappR\_tissues*

---

## Description

Tissues available in scMappR.

## Usage

```
data(scMappR_tissues)
```

## Format

A vector of tissue names available for tissue\_scMappR\_internal or to download and use in scMappR\_and\_pathway\_analysis.

**scMappR\_tissues** A list of 174 tissue names from PanglaoDB.

## Details

A vector of tissues available in scMappR.

## Examples

```
data(scMappR_tissues)
```

---

seurat_to_genes	<i>Identify all cell-type markers</i>
-----------------	---------------------------------------

---

**Description**

Takes processed Seurat matrix and identifies cell-type markers with FindMarkers in Seurat.

**Usage**

```
seurat_to_genes(pbmc, test = "wilcox")
```

**Arguments**

pbmc	Processed Seurat object.
test	statistical test for calling CT markers – must be in Seurat.

**Details**

Internal: This function runs the FindMarkers function from Seurat in a loop, will use the Seurat v2 or Seurat v3 object after identifying which Seurat object is inputted. It then takes the output of the FindMarkers and puts it in a list, returning it.

**Value**

seurat\_to\_genes A list of genes where their over-representation in the i'th cell-type is computed. Each element contains the gene name, adjusted p-value, and the log2Fold-Change of each gene being present in that cell-type.

**Examples**

```
data(sm)
toProcess <- list(example = sm)
tst1 <- process_from_count(toProcess, "testProcess")
genes <- seurat_to_genes(tst1)
```

---

`single_gene_preferences`*Single cell-type gene preferences*

---

**Description**

Measure enrichment of individual cell-types in a signature matrix.

Internal function as part of `tissue_scMappR_internal()`. This function takes genes preferentially expressed within a gene list, each cell-type and the background (i.e. all genes within the signature matrix) before completing the cell-type specific enrichment of the inputted gene list on each cell type. This function then returns a table describing the cell-type enrichments (p-value and odds ratio) of each cell-type.

**Usage**

```
single_gene_preferences(  
  hg_short,  
  hg_full,  
  study_name,  
  outDir,  
  toSave = FALSE,  
  path = NULL  
)
```

**Arguments**

<code>hg_short</code>	A list with two objects: a "preferences" and a "genesIn". Preferences is a list of gene symbols over-represented in each cell-type and genesIn were all the inputted genes.
<code>hg_full</code>	The same as <code>hg_short</code> but for every gene in the signature matrix.
<code>study_name</code>	Name of output table.
<code>outDir</code>	Directory where table is outputted.
<code>toSave</code>	Allow <code>scMappR</code> to write files in the current directory (T/F).
<code>path</code>	If <code>toSave == TRUE</code> , path to the directory where files will be saved.

**Value**

`single_gene_preferences` A gene-set enrichment table of individual cell-type enrichment.

**Examples**

```
# load in signature matrices  
data(POA_example)
```

```
POA_generes <- POA_example$POA_generes
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
sig <- get_gene_symbol(POA_Rank_signature)
Signature <- POA_Rank_signature
rownames(Signature) <- sig$rowname
genes <- rownames(Signature)[1:60]
heatmap_test <- tissue_scMappR_custom( genes, signature_matrix = Signature,
                                     output_directory = "scMappR_test", toSave = FALSE)
single_preferences <- heatmap_test$single_celltype_preferences
```

---

sm

*single\_cell\_process*

---

### Description

Example data for processing scRNA-seq count data with Seurat.

### Usage

```
data(sm)
```

### Format

A 752 x 236 matrix of class dgCMatix where rows are genes and columns are cells. Data matrix is filled with counts detected from scRNAseq.

**TCTCTAACACAGGCCT** Barcode of one of the sequenced cells present. Each column is the count from a scRNA-seq dataset reprocessed by PanglaoDB.

### Details

A dgCMatix object containing count data for scRNA-seq processing.

### Examples

```
data(sm)
```

---

```
tissue_by_celltype_enrichment
      tissue_by_celltype_enrichment
```

---

## Description

This function uses a Fisher's-exact-test to rank gene-set enrichment.

## Usage

```
tissue_by_celltype_enrichment(
  gene_list,
  species,
  name = "CT_Tissue_example",
  p_thresh = 0.05,
  rda_path = "",
  isect_size = 3,
  return_gmt = FALSE
)
```

## Arguments

gene_list	A character vector of gene symbols with the same designation (e.g. mouse symbol - mouse, human symbol - human) as the gene set database.
species	Species of cell-type marker to use ('human' or 'mouse').
name	Name of the pdf to be printed.
p_thresh	The Fisher's test cut-off for a cell-marker to be enriched.
rda_path	Path to a .rda file containing an object called "gmt". Either human or mouse cell-type markers split by experiment. If the correct file isn't present they will be downloaded from <a href="https://github.com/wilsonlabgroup/scMappR_Data">https://github.com/wilsonlabgroup/scMappR_Data</a> .
isect_size	Number of genes in your list and the cell-type.
return_gmt	Return .gmt file – recommended if downloading from online as it may have updated (T/F).

## Details

Complete a Fisher's-exact test of an input list of genes against one of the two curated tissue by cell-type marker datasets from scMappR.

## Value

List with the following elements:

enriched	Data frame of enriched cell-types from tissues.
gmt	Cell-markers in enriched cell-types from tissues.

## Examples

```

data(POA_example)
POA_geneses <- POA_example$POA_geneses
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
Signature <- POA_Rank_signature
rowname <- get_gene_symbol(Signature)
rownames(Signature) <- rowname$rowname
genes <- rownames(Signature)[1:100]

enriched <- tissue_by_celltype_enrichment(gene_list = genes,
species = "mouse",p_thresh = 0.05, isect_size = 3)

```

---

tissue\_scMappR\_custom *Gene List Visualization and Enrichment with Custom Signature Matrix*

---

## Description

This function visualizes signature matrix, clusters subsetted genes, completes enrichment of individual cell-types and co-enrichment.

## Usage

```

tissue_scMappR_custom(
  gene_list,
  signature_matrix,
  output_directory = "custom_test",
  toSave = FALSE,
  path = NULL,
  gene_cutoff = 1,
  is_pvalue = TRUE
)

```

## Arguments

gene_list	A list of gene symbols matching that of the signature_matrix. Any gene symbol is acceptable.
signature_matrix	Pre-computed signature matrix with matching gene names.
output_directory	Directory made containing output of functions.
toSave	Allow scMappR to write files in the current directory (T/F).

path	If toSave == TRUE, path to the directory where files will be saved.
gene_cutoff	Value cut-off (generally rank := log10(Padj)) for a gene to be considered a marker.
is_pvalue	If signature matrix is p-value before rank is applied (not recommended) (T/F).

### Details

This function is roughly the same as `tissue_scMappR_internal`, however now there is a custom signature matrix. It generates a heatmap of the signature matrix and your inputted gene list, as well as single cell-type and co-celltype enrichment.

### Value

List with the following elements:

background_heatmap	Data frame of the entire gene by cell-type signature matrix inputted.
gene_list_heatmap	Data frame of inputted signature matrix subsetted by input genes.
single_celltype_preferences	Data frame of enriched cell-types.
group_celtype_preference	Data frame of groups of cell-types enriched by the same genes.

### Examples

```
# load in signature matrices
data(POA_example)
POA_genes <- POA_example$POA_genes
POA_OR_signature <- POA_example$POA_OR_signature
POA_Rank_signature <- POA_example$POA_Rank_signature
sig <- get_gene_symbol(POA_Rank_signature)
Signature <- POA_Rank_signature
rownames(Signature) <- sig$rowname
genes <- rownames(Signature)[1:60]
heatmap_test <- tissue_scMappR_custom( genes, signature_matrix = Signature,
                                     output_directory = "scMappR_test", toSave = FALSE)
```

---

tissue\_scMappR\_internal

*Gene List Visualization and Enrichment (Internal)*

---

### Description

This function loops through every signature matrix in a particular tissue and generates heatmaps, cell-type preferences, and co-enrichment.



**Usage**

```
tissue_scMappR_internal(
  gene_list,
  species,
  output_directory,
  tissue,
  rda_path = "",
  cluster = "Pval",
  genececx = 0.01,
  raw_pval = FALSE,
  path = NULL,
  toSave = FALSE,
  drop_unkown_celltype = FALSE
)
```

**Arguments**

gene_list	A list of gene symbols, mouse or human.
species	"mouse", "human" or "-9" if using a precomputed signature matrix.
output_directory	If toSave = TRUE, the name of the output directory that would be built.
tissue	Name of the tissue in "get_tissues".
rda_path	Path to the .rda file containing all of the signature matrices.
cluster	'Pval' or 'OR' depending on if you want to cluster odds ratios or p-values of cell-type preferences.
genececx	The size of the gene names of the rows in the heatmap.
raw_pval	If the inputted signature matrix are raw (untransformed) p-values – recommended to generate rank first (T/F).
path	If toSave == TRUE, path to the directory where files will be saved.
toSave	Allow scMappR to write files in the current directory (T/F).
drop_unkown_celltype	Whether or not to remove "unknown" cell-types from the signature matrix (T/F).

**Details**

This function takes a list of genes and a tissue that is contained in current signature matrices before and generating heatmaps of cell-type preferences. It then completes cell-type enrichment of each individual cell-type, then, if more than two cell-types are significantly enriched, co-enrichment. of those enriched cell-types is then computed.

**Value**

List with the following elements:

background_heatmap	Data frame of the entire gene by cell-type signature matrix inputted.
--------------------	---

gene\_list\_heatmap  
                                   Data frame of inputted signature matrix subsetted by input genes.

single\_celltype\_preferences  
                                   Data frame of enriched cell-types.

group\_celtype\_preference  
                                   Data frame of groups of cell-types enriched by the same genes.

## Examples

```
data(POA_example) # region to preoptic area
Signature <- POA_example$POA_Rank_signature # signature matrix
rowname <- get_gene_symbol(Signature) # get signature
rownames(Signature) <- rowname$rowname
genes <- rownames(Signature)[1:60]
rda_path1 = "" # data directory (if it exists)

# set toSave = TRUE and path = output directory of your choice
internal <- tissue_scMappR_internal(genes, "mouse", output_directory = "scMappR_TesInternal",
                                   tissue = "hypothalamus", toSave = FALSE)
```

---

tochr	<i>To Character.</i>
-------	----------------------

---

## Description

This function checks if your vector is not a character and if not, will convert it to a character.

## Usage

```
tochr(x)
```

## Arguments

x                            A character, factor or numeric vector.

## Value

tochr Returns a character vector.

**Examples**

```
# vector of factors
fact <- factor(c("a", "b", "c", "d"))
# convert to character
char <- tochr(fact)
```

---

toNum

*To Numeric.*

---

**Description**

This function checks if your vector is not a character and if it is, then converts it to a numeric.

**Usage**

```
toNum(x)
```

**Arguments**

x                    A character, factor, or numeric vector.

**Value**

toNum Returns a numeric vector.

**Examples**

```
# vector of factors
fact <- factor(c("1", "2", "3", "4"))
# convert to numeric
num <- toNum(fact)
```

---

topgenes_extract	<i>Extract Top Markers</i>
------------------	----------------------------

---

**Description**

Internal – Extracts strongest cell-type markers from a Seurat object.

**Usage**

```
topgenes_extract(generes, padj = 0.05, FC = 1.5, topNum = 30)
```

**Arguments**

generes	A list of cell-type markers with fold-changes and p-values (FindMarkers output in Seurat).
padj	The p-value (FDR) cutoff.
FC	The fold-change cutoff.
topNum	The number of genes to extract.

**Details**

Internal, this function runs through a list of outputs from FindMarkers objects in Seurat and will extract genes past a padj and fold-change threshold. Then it extracts the topNum number of genes. if you have not used the FindMarkers function, then a list of summary statistics with fold-change designated by avg\_logFC and p-val by p\_val\_adj.

**Value**

topgenes\_extract Returns a list of character vectors with the top (topNum) of gene markers for each cell-type.

**Examples**

```
# load generes object
data(POA_example)
topGenes <- topgenes_extract(POA_example$POA_generes)
```

---

```
two_method_pathway_enrichment
    two_method_pathway_enrichment
```

---

## Description

Pathway analysis of each cell-type based on cell-type specificity and rank improvement by scMappR.

## Usage

```
two_method_pathway_enrichment(
  DEG_list,
  theSpecies,
  scMappR_vals,
  background_genes = NULL,
  output_directory = "output",
  plot_names = "reweighted",
  number_genes = -9,
  newGprofiler = FALSE,
  toSave = FALSE,
  path = NULL
)
```

## Arguments

DEG_list	Differentially expressed genes (gene_name, padj, log2fc).
theSpecies	Human, mouse, or a character that is compatible with g:ProfileR.
scMappR_vals	cell weighted Fold-changes of differentially expressed genes.
background_genes	A list of background genes to test against. NULL assumes all genes in g:profileR gene set databases.
output_directory	Path to the directory where files will be saved.
plot_names	Names of output.
number_genes	Number of genes to if there are many, many DEGs.
newGprofiler	Whether to use g:ProfileR or gprofiler2 (T/F).
toSave	Allow scMappR to write files in the current directory (T/F).
path	If toSave == TRUE, path to the directory where files will be saved.

## Details

This function re-ranks cwFoldChanges based on their absolute cell-type specificity scores (per-celltype) as well as their rank increase in cell-type specificity before completing an ordered pathway analysis. In the second method, only genes with a rank increase in cell-type specificity were included.

**Value**

List with the following elements:

`rank_increase` A list containing the degree of rank change between bulk DE genes and cwFold-changes. Pathway enrichment and TF enrichment of these reranked genes.

`non_rank_increase` list of DFs containing the pathway and TF enrichment of cwFold-changes.

**Examples**

```
# load data for scMappR
data(PBMC_example)
bulk_DE_cors <- PBMC_example$bulk_DE_cors
bulk_normalized <- PBMC_example$bulk_normalized
odds_ratio_in <- PBMC_example$odds_ratio_in
case_grep <- "_female"
control_grep <- "_male"
max_proportion_change <- 10
print_plots <- FALSE
theSpecies <- "human"

# calculate cwFold-changes
toOut <- scMappR_and_pathway_analysis(bulk_normalized, odds_ratio_in,
                                     bulk_DE_cors, case_grep = case_grep,
                                     control_grep = control_grep, rda_path = "",
                                     max_proportion_change = 10, print_plots = TRUE,
                                     plot_names = "tst1", theSpecies = "human",
                                     output_directory = "tester",
                                     sig_matrix_size = 3000, up_and_downregulated = FALSE,
                                     internet = FALSE)

# complete pathway enrichment using both methods
twoOutFiles <- two_method_pathway_enrichment(bulk_DE_cors, "human",
scMappR_vals = toOut$cellWeighted_Foldchange, background_genes = rownames(bulk_normalized),
output_directory = "newfun_test", plot_names = "nonreranked_", toSave = FALSE)
```

# Index

## \* datasets

- gmt, [16](#)
- PBMC\_example, [25](#)
- POA\_example, [27](#)
- scMappR\_tissues, [34](#)
- sm, [37](#)

- cellmarker\_enrich, [3](#)
- coEnrich, [4](#)
- compare\_deconvolution\_methods, [5](#)
- cwFoldChange\_evaluate, [7](#)

- DeconRNAseq\_CRAN, [9](#)
- deconvolute\_and\_contextualize, [10](#)

- extract\_genes\_cell, [12](#)

- generes\_to\_heatmap, [13](#)
- get\_gene\_symbol, [15](#)
- get\_signature\_matrices, [16](#)
- gmt, [16](#)
- gProfiler\_cellWeighted\_Foldchange, [17](#)
- gsva\_cellIdentify, [18](#)

- heatmap\_generation, [20](#)
- human\_mouse\_ct\_marker\_enrich, [21](#)

- make\_TF\_barplot, [22](#)

- pathway\_enrich\_internal, [24](#)
- PBMC\_example, [25](#)
- plotBP, [26](#)
- POA\_example, [27](#)
- process\_dgTMatrix\_lists, [28](#)
- process\_from\_count, [30](#)

- scMappR\_and\_pathway\_analysis, [31](#)
- scMappR\_tissues, [34](#)
- seurat\_to\_generes, [35](#)
- single\_gene\_preferences, [36](#)
- sm, [37](#)

- tissue\_by\_celltype\_enrichment, [38](#)
- tissue\_scMappR\_custom, [39](#)
- tissue\_scMappR\_internal, [40](#)
- tochr, [42](#)
- toNum, [43](#)
- topgenes\_extract, [44](#)
- two\_method\_pathway\_enrichment, [45](#)