Package: SlimR (via r-universe)

October 16, 2026

Version 1.2.0

Title Adaptive Machine Learning Powered, Context Matching Tool for Single-Cell and Spatial Transcriptomics Annotation

Description Annotates single-cell and spatial-transcriptomic (ST) data using context-matching marker datasets. It creates a unified marker list (`Markers_list`) from multiple sources: built-in curated databases ('Cellmarker2', 'PanglaoDB', 'ScType', 'CellTypist', 'scBaseCount', 'scIBD', 'TCellSI', 'PCTIT', 'PCTAM'), Seurat objects with cell labels, or user-provided Excel tables. SlimR first uses adaptive machine learning for parameter optimization, and then offers two automated annotation approaches: 'cluster-based' and 'per-cell'. Cluster-based annotation assigns one label per cluster, expression-based probability calculation, and AUC validation. Per-cell annotation assigns labels to individual cells using three scoring methods with adaptive thresholds and ratio-based confidence filtering, plus optional UMAP spatial smoothing, making it ideal for heterogeneous clusters and rare cell types. The package also supports semi-automated workflows with heatmaps, feature plots, and combined visualizations for manual annotation. For more information, see the package documentation at <https://github.com/zhaoqing-wang/SlimR>.

License MIT + file LICENSE

URL https://github.com/zhaoqing-wang/SlimR

BugReports https://github.com/zhaoqing-wang/SlimR/issues

Depends R (>= 4.1.0)

Imports cowplot, dplyr, ggplot2, patchwork, pheatmap, readxl, scales, Seurat, tidyr, tools, tibble, gtools, RColorBrewer, methods

Suggests crayon, RANN, testthat (>= 3.0.0), WeightedTreemaps

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

Date 2026-10-16

Config/roxygen2/version 8.1.1

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

Repository https://zhaoqing-wang.r-universe.dev

Date/Publication 2026-10-16 03:10:50 UTC

RemoteUrl https://github.com/zhaoqing-wang/slimr

RemoteRef HEAD

RemoteSha 59dfb7f36d1cd6f3655830583fc7f6077159ff21

Contents

ArchRPalettes . . . 3
calculate_cluster_variability . . . 4
calculate_expression . . . 4
calculate_expression_skewness . . . 5
calculate_probability . . . 6
Cellmarker2 . . . 7
Cellmarker2_raw . . . 7
Cellmarker2_table . . . 8
Celltype_Annotation . . . 9
Celltype_annotation_Cellmarker2 . . . 10
Celltype_Annotation_Combined . . . 12
Celltype_annotation_Excel . . . 13
Celltype_Annotation_Features . . . 15
Celltype_Annotation_Heatmap . . . 17
Celltype_annotation_PanglaoDB . . . 19
Celltype_Annotation_PerCell . . . 20
Celltype_annotation_Seurat . . . 22
Celltype_Calculate . . . 23
Celltype_Calculate_PerCell . . . 26
Celltype_Compare . . . 29
Celltype_Verification . . . 31
Celltype_Verification_PerCell . . . 32
CellTypist . . . 33
compute_adaptive_parameters . . . 34
Compute_Gene_AUC_ROC . . . 35
Compute_Gene_PRC . . . 37
estimate_batch_effect . . . 40
extract_dataset_features . . . 40
Markers_filter_Cellmarker2 . . . 41
Markers_filter_PanglaoDB . . . 42
Markers_filter_ScType . . . 43
Markers_list_PCTAM . . . 44
Markers_list_PCTIT . . . 45

Markers_list_scIBD 46
Markers_list_TCellSI 46
PanglaoDB 47
PanglaoDB_raw 48
PanglaoDB_table 48
Parameter_Calculate 49
percell_workflow 50
plot.pheatmap 55
Plot_Group_Gene_Heatmap 56
Plot_Hierarchy_Proportion 58
Plot_Voronoi_diagram 60
Read_excel_markers 63
Read_seurat_markers 64
scBaseCount 65
ScType 66
ScType_raw 67
ScType_table 68

Index **69**

`ArchRPalettes` *ArchR colour palettes*

Description

A list of colour vectors reproducing the palettes used by the **ArchR** package. Names correspond to the `set` argument of `paletteDiscrete()`.

Usage

```
ArchRPalettes
```

Format

A named list of character vectors.

Source

https://github.com/GreenleafLab/ArchR

calculate_cluster_variability *Calculate Cluster Variability (Use in package)*

Description

Measures the degree of separation between different cell clusters based on expression patterns.

Usage

```
calculate_cluster_variability(data.features, features)
```

Arguments

data.features	Data frame containing expression data and cluster labels
features	Feature names to include in analysis

Value

Numeric value representing cluster separation strength

See Also

Other Section_1_Functions_Use_in_Package: calculate_expression(), calculate_expression_skewness(), calculate_probability(), compute_adaptive_parameters(), estimate_batch_effect(), extract_dataset_featu

calculate_expression *Counts average expression of gene set (Use in package)*

Description

Counts average expression of gene set (Use in package)

Usage

```
calculate_expression(
  object,
  features,
  assay = NULL,
  cluster_col = NULL,
  colour_low = "white",
  colour_high = "navy"
)
```

Arguments

object	Enter a Seurat object.
features	Enter one or a set of markers.
assay	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = NULL".
cluster_col	Enter the meta.data column in the Seurat object to be annotated, such as "seurat_cluster". Default parameters use "cluster_col = NULL".
colour_low	Color for lowest expression level. (default = "white")
colour_high	Color for highest expression level. (default = "black")

Value

Average expression genes and relatied informations in the input "Seurat" object given "cluster_col" and given "features".

See Also

Other Section_1_Functions_Use_in_Package: calculate_cluster_variability(), calculate_expression_skewness(
calculate_probability(), compute_adaptive_parameters(), estimate_batch_effect(), extract_dataset_featur

---

calculate_expression_skewness

Calculate Expression Distribution Skewness (Use in package)

Description

Computes the average skewness of gene expression distributions across all features.

Usage

```
calculate_expression_skewness(expression_matrix)
```

Arguments

expression_matrix	Matrix of expression values

Value

Mean absolute skewness across all genes

See Also

Other Section_1_Functions_Use_in_Package: calculate_cluster_variability(), calculate_expression(),
calculate_probability(), compute_adaptive_parameters(), estimate_batch_effect(), extract_dataset_featur

`calculate_probability` *Calculate gene set expression and infer probabilities with control datasets (Use in package)*

Description

Calculate gene set expression and infer probabilities with control datasets (Use in package)

Usage

```
calculate_probability(
  object,
  features,
  assay = NULL,
  cluster_col = NULL,
  min_expression = 0.1,
  specificity_weight = 3
)
```

Arguments

`object`	Enter a Seurat object.
`features`	Enter one or a set of markers.
`assay`	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = NULL".
`cluster_col`	Enter the meta.data column in the Seurat object to be annotated, such as "seurat_cluster". Default parameters use "cluster_col = NULL".
`min_expression`	The min_expression parameter defines a threshold value to determine whether a cell's expression of a feature is considered "expressed" or not. It is used to filter out low-expression cells that may contribute noise to the analysis. Default parameters use "min_expression = 0.1".
`specificity_weight`	The specificity_weight parameter controls how much the expression variability (standard deviation) of a feature within a cluster contributes to its "specificity score." It amplifies or suppresses the impact of variability in the final score calculation.Default parameters use "specificity_weight = 3".

Value

Average expression of genes in the input "Seurat" object given "cluster_col" and given "features".

See Also

Other Section_1_Functions_Use_in_Package: `calculate_cluster_variability()`, `calculate_expression()`, `calculate_expression_skewness()`, `compute_adaptive_parameters()`, `estimate_batch_effect()`, `extract_dataset_features()`

Cellmarker2 *Cellmarker2 dataset*

Description

A dataset containing marker genes for different cell types from Cellmarker2

Usage

```
Cellmarker2
```

Format

A data frame with 8 columns:

Details

This dataset is used to filter and create a standardized marker list. The dataset can be filtered based on species, tissue class, tissue type, cancer type, and cell type to generate a list of marker genes for specific cell types.

Source

http://117.50.127.228/CellMarker/

See Also

Other Section_0_Database: CellTypist, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

Cellmarker2_raw *Cellmarker2 raw dataset*

Description

A dataset containing marker genes for different cell types from Cellmarker2

Usage

```
Cellmarker2_raw
```

Format

A data frame with 20 columns contined in the Cellmarker2 database:

Details

This dataset is used to filter and create a standardized marker list. The dataset can be filtered based on species, tissue class, tissue type, cancer type, and cell type to generate a list of marker genes for specific cell types.

Source

http://117.50.127.228/CellMarker/

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

---

Cellmarker2_table *Cellmarker2 table*

Description

A dataset containing marker genes for different cell types from Cellmarker2

Usage

```
Cellmarker2_table
```

Format

A list contain different types like species, tissue_class, tissue_type, cancer_type, cell_type

Details

This list is used to choose filters for creation of standardized marker list.

Source

http://117.50.127.228/CellMarker/

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

`Celltype_Annotation` *Annotate Seurat Object with SlimR Cell Type Predictions*

Description

This function assigns SlimR predicted cell types to a Seurat object based on cluster annotations, and stores the results in the meta.data slot.

Usage

```
Celltype_Annotation(
  seurat_obj,
  cluster_col,
  SlimR_anno_result,
  plot_UMAP = TRUE,
  annotation_col = "Cell_type_SlimR"
)
```

Arguments

`seurat_obj`	A Seurat object containing cluster information in meta.data.
`cluster_col`	Character string indicating the column name in meta.data that contains cluster IDs.
`SlimR_anno_result`	List generated by function Celltype_Calculate() which containing a data.frame in $Prediction_results with: 1.cluster_col (Cluster identifiers (should match cluster_col in meta.data)) 2.Predicted_cell_type (Predicted cell types for each cluster).
`plot_UMAP`	logical(1); if TRUE, plot the UMAP with cell type annotations.
`annotation_col`	The location to write in 'meta.data' that contains the predicted cell type. (default = "Cell_type_SlimR")

Value

A Seurat object with updated meta.data containing the predicted cell types.

Note

If plot_UMAP = TRUE, this function will print a UMAP plot as a side effect.

See Also

Other Section_3_Automated_Annotation: `Celltype_Annotation_PerCell()`, `Celltype_Calculate()`, `Celltype_Calculate_PerCell()`, `Celltype_Verification()`, `Celltype_Verification_PerCell()`, `Parameter_Calculate()`, `percell_workflow`

Examples

```
## Not run:
sce <- Celltype_Annotation(seurat_obj = sce,
    cluster_col = "seurat_clusters",
    SlimR_anno_result = SlimR_anno_result,
    plot_UMAP = TRUE,
    annotation_col = "Cell_type_SlimR"
    )

## End(Not run)
```

Celltype_annotation_Cellmarker2

Uses "marker_list" from Cellmarker2 for cell annotation

Description

Uses "marker_list" from Cellmarker2 for cell annotation

Usage

```
Celltype_annotation_Cellmarker2(
  seurat_obj,
  gene_list,
  species,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  save_path = NULL,
  min_counts = 1,
  colour_low = "white",
  colour_high = "navy",
  colour_low_mertic = "white",
  colour_high_mertic = "navy"
)
```

Arguments

seurat_obj	Enter the Seurat object with annotation columns such as "seurat_cluster" in meta.data to be annotated.
gene_list	Enter the standard "Marker_list" generated by the Cellmarker2 database for the SlimR package, generated by the "Markers_filter_Cellmarker2 ()" function.
species	This parameter selects the species "Human" or "Mouse" for standard gene format correction of markers entered by "Marker_list".
cluster_col	Enter annotation columns such as "seurat_cluster" in meta.data of the Seurat object to be annotated. Default parameters use "cluster_col = 'seurat_clusters'".

assay — Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = "RNA"".

save_path — The output path of the cell annotation picture. Example parameters use "save_path = './SlimR/Celltype_annotation_Cellmarker2/'".

min_counts — The minimum number of counts of genes in "Marker_list" entered. This number represents the number of the same gene in the same species and the same location in the Cellmarker2 database used for annotation of this cell type. Default parameters use "min_counts = 1".

colour_low — Color for lowest expression level. (default = "white")

colour_high — Color for highest expression level. (default = "navy")

colour_low_mertic — Color for lowest mertic level. (default = "white")

colour_high_mertic — Color for highest mertic level. (default = "navy")

Value

The cell annotation picture is saved in "save_path".

See Also

Other Section_5_Other_Functions_Provided: Celltype_Compare(), Celltype_annotation_Excel(), Celltype_annotation_PanglaoDB(), Celltype_annotation_Seurat(), Compute_Gene_AUC_ROC(), Compute_Gene_PRC(), Plot_Group_Gene_Heatmap(), Plot_Hierarchy_Proportion(), Plot_Voronoi_diagram(), paletteDiscrete()

Examples

```
## Not run:
Celltype_annotation_Cellmarker2(seurat_obj = sce,
    gene_list = Markers_list_Cellmarker2,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_Cellmarker2")
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

## End(Not run)
```

`Celltype_Annotation_Combined` *Uses "marker_list" to generate combined plot for cell annotation*

Description

Uses "marker_list" to generate combined plot for cell annotation

Usage

```
Celltype_Annotation_Combined(
  seurat_obj,
  gene_list,
  species,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  save_path = NULL,
  colour_low = "white",
  colour_high = "navy"
)
```

Arguments

`seurat_obj`	Enter the Seurat object with annotation columns such as "seurat_cluster" in meta.data to be annotated.
`gene_list`	A list of cells and corresponding gene controls, the name of the list is cell type, and the first column of the list corresponds to markers. Lists can be generated using functions such as "Markers_filter_Cellmarker2 ()", "Markers_filter_PanglaoDB ()", "read_excel_markers ()", "read_seurat_markers ()", etc.
`species`	This parameter selects the species "Human" or "Mouse" for standard gene format correction of markers entered by "Marker_list".
`cluster_col`	Enter annotation columns such as "seurat_cluster" in meta.data of the Seurat object to be annotated. Default parameters use "cluster_col = 'seurat_clusters'".
`assay`	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = 'RNA'".
`save_path`	The output path of the cell annotation picture. Example parameters use "save_path = './SlimR/Celltype_annotation_Bar/'".
`colour_low`	Color for lowest expression level. (default = "white")
`colour_high`	Color for highest expression level. (default = "navy")

Value

The cell annotation picture is saved in "save_path".

See Also

Other Section_4_Semi_Automated_Annotation: Celltype_Annotation_Features(), Celltype_Annotation_Heatmap()

Examples

```
## Not run:
Celltype_Annotation_Combined(seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_Annotation_Combined"),
    colour_low = "white",
    colour_high = "navy"
    )

## End(Not run)
```

Celltype_annotation_Excel

Uses "marker_list" from Excel input for cell annotation

Description

Uses "marker_list" from Excel input for cell annotation

Usage

```
Celltype_annotation_Excel(
  seurat_obj,
  gene_list,
  species,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  save_path = NULL,
  metric_names = NULL,
  colour_low = "white",
  colour_high = "navy",
  colour_low_mertic = "white",
  colour_high_mertic = "navy"
)
```

Arguments

seurat_obj	Enter the Seurat object with annotation columns such as "seurat_cluster" in meta.data to be annotated.

gene_list	Enter the standard "Marker_list" generated by the Excel files database for the SlimR package, generated by the "read_excel_markers()" function.
species	This parameter selects the species "Human" or "Mouse" for standard gene format correction of markers entered by "Marker_list".
cluster_col	Enter annotation columns such as "seurat_cluster" in meta.data of the Seurat object to be annotated. Default parameters use "cluster_col = "seurat_clusters"".
assay	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = 'RNA'".
save_path	The output path of the cell annotation picture. Example parameters use "save_path = './SlimR/Celltype_annotation_Excel/'".
metric_names	Change the row name for the input mertics, not recommended unless necessary. (NULL is used as default parameter)
colour_low	Color for lowest expression level. (default = "white")
colour_high	Color for highest expression level. (default = "navy")
colour_low_mertic	Color for lowest mertic level. (default = "white")
colour_high_mertic	Color for highest mertic level. (default = "navy")

Value

The cell annotation picture is saved in "save_path".

See Also

Other Section_5_Other_Functions_Provided: Celltype_Compare(), Celltype_annotation_Cellmarker2(), Celltype_annotation_PanglaoDB(), Celltype_annotation_Seurat(), Compute_Gene_AUC_ROC(), Compute_Gene_PRC(), Plot_Group_Gene_Heatmap(), Plot_Hierarchy_Proportion(), Plot_Voronoi_diagram(), paletteDiscrete()

Examples

```
## Not run:
Celltype_annotation_Excel(seurat_obj = sce,
    gene_list = Markers_list_Excel,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_Excel")
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

## End(Not run)
```

`Celltype_Annotation_Features` *Annotate cell types using features plot with different marker databases*

Description

This function dynamically selects the appropriate annotation method based on the `gene_list_type` parameter. It supports marker databases from Cellmarker2, PanglaoDB, Seurat (via `FindAllMarkers`), or Excel files.

Usage

```
Celltype_Annotation_Features(
  seurat_obj,
  gene_list,
  gene_list_type = "Default",
  species = NULL,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  save_path = NULL,
  min_counts = 1,
  metric_names = NULL,
  colour_low = "white",
  colour_high = "navy",
  colour_low_mertic = "white",
  colour_high_mertic = "navy",
  ...
)
```

Arguments

`seurat_obj`	A valid Seurat object with cluster annotations in `meta.data`.
`gene_list`	A list of data frames containing marker genes and metrics. Format depends on `gene_list_type`: • **Cellmarker2**: Generated by `Markers_filter_Cellmarker2()`. • **PanglaoDB**: Generated by `Markers_filter_PanglaoDB()`. • **Seurat**: Generated by `read_seurat_markers()`. • **Excel**: Generated by `read_excel_markers()`.
`gene_list_type`	Type of marker database to use. Be one of: `"Cellmarker2"`, `"PanglaoDB"`, `"Seurat"`, or `"Excel"`.
`species`	Species of the dataset: `"Human"` or `"Mouse"` for gene name standardization.
`cluster_col`	Column name in `meta.data` defining clusters (default: `"seurat_clusters"`).
`assay`	Assay layer in the Seurat object (default: `"RNA"`).
`save_path`	Directory to save output PNGs. Must be explicitly specified.

min_counts	Minimum number of counts for Cellmarker2 annotations (default: 1).
metric_names	Optional. Change the row name for the input mertics, not recommended unless necessary. (NULL is used as default parameter; used in "Seurat"/"Excel").
colour_low	Color for lowest expression level. (default = "white")
colour_high	Color for highest expression level. (default = "navy")
colour_low_mertic	Color for lowest mertic level. (default = "white")
colour_high_mertic	Color for highest mertic level. (default = "navy")
...	Additional parameters passed to the specific annotation function.

Value

Saves cell type annotation PNGs in save_path. Returns invisibly.

See Also

Other Section_4_Semi_Automated_Annotation: Celltype_Annotation_Combined(), Celltype_Annotation_Heatmap()

Examples

```
## Not run:
# Example for Cellmarker2
Celltype_Annotation_Features(seurat_obj = sce,
    gene_list = Markers_list_Cellmarker2,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_Cellmarker2"),
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

# Example for PanglaoDB
Celltype_Annotation_Features(seurat_obj = sce,
    gene_list = Markers_list_panglaoDB,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_PanglaoDB")
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

# Example for Seurat marker list
Celltype_Annotation_Features(seurat_obj = sce,
```

```
    gene_list = Markers_list_Seurat,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_Seurat")
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

# Example for Excel marker list
Celltype_Annotation_Features(seurat_obj = sce,
    gene_list = Markers_list_Excel,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_Excel")
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

## End(Not run)
```

`Celltype_Annotation_Heatmap` *Uses "marker_list" to generate heatmap for cell annotation*

Description

Uses "marker_list" to generate heatmap for cell annotation

Usage

```
Celltype_Annotation_Heatmap(
  seurat_obj,
  gene_list,
  species,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  min_expression = 0.1,
  specificity_weight = 3,
  colour_low = "navy",
  colour_high = "firebrick3"
)
```

Arguments

seurat_obj	Enter the Seurat object with annotation columns such as "seurat_cluster" in meta.data to be annotated.
gene_list	A list of cells and corresponding gene controls, the name of the list is cell type, and the first column of the list corresponds to markers. Lists can be generated using functions such as "Markers_filter_Cellmarker2 ()", "Markers_filter_PanglaoDB ()", "read_excel_markers ()", "read_seurat_markers ()", etc.
species	This parameter selects the species "Human" or "Mouse" for standard gene format correction of markers entered by "Marker_list".
cluster_col	Enter annotation columns such as "seurat_cluster" in meta.data of the Seurat object to be annotated. Default parameters use "cluster_col = 'seurat_clusters'".
assay	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = 'RNA'".
min_expression	The min_expression parameter defines a threshold value to determine whether a cell's expression of a feature is considered "expressed" or not. It is used to filter out low-expression cells that may contribute noise to the analysis. Default parameters use "min_expression = 0.1".
specificity_weight	The specificity_weight parameter controls how much the expression variability (standard deviation) of a feature within a cluster contributes to its "specificity score." It amplifies or suppresses the impact of variability in the final score calculation.Default parameters use "specificity_weight = 3".
colour_low	Color for lowest probability level in Heatmap visualization of probability matrix. (default = "navy")
colour_high	Color for highest probability level Heatmap visualization of probability matrix. (default = "firebrick3")

Value

The heatmap of the comparison between "cluster_col" in the Seurat object and the given gene set "gene_list" needs to be annotated.

See Also

Other Section_4_Semi_Automated_Annotation: Celltype_Annotation_Combined(), Celltype_Annotation_Features(

Examples

```
## Not run:
Celltype_Annotation_Heatmap(seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    min_expression = 0.1,
    specificity_weight = 3,
    colour_low = "navy",
```

```
    colour_high = "firebrick3"
    )

## End(Not run)
```

`Celltype_annotation_PanglaoDB` *Uses "marker_list" from PanglaoDB for cell annotation*

Description

Uses "marker_list" from PanglaoDB for cell annotation

Usage

```
Celltype_annotation_PanglaoDB(
  seurat_obj,
  gene_list,
  species,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  save_path = NULL,
  metric_names = NULL,
  colour_low = "white",
  colour_high = "navy",
  colour_low_mertic = "white",
  colour_high_mertic = "navy"
)
```

Arguments

`seurat_obj`	Enter the Seurat object with annotation columns such as "seurat_cluster" in meta.data to be annotated.
`gene_list`	Enter the standard "Marker_list" generated by the PanglaoDB database for the SlimR package, generated by the "Markers_filter_PanglaoDB ()" function.
`species`	This parameter selects the species "Human" or "Mouse" for standard gene format correction of markers entered by "Marker_list".
`cluster_col`	Enter annotation columns such as "seurat_cluster" in meta.data of the Seurat object to be annotated. Default parameters use "cluster_col = 'seurat_clusters'".
`assay`	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = 'RNA'".
`save_path`	The output path of the cell annotation picture. Example parameters use "save_path = './SlimR/Celltype_annotation_PanglaoDB/'".
`metric_names`	Warning: Do not enter information. This parameter is used to check if "Marker_list" conforms to the PanglaoDB database output.

colour_low Color for lowest expression level. (default = "white")

colour_high Color for highest expression level. (default = "navy")

colour_low_mertic
Color for lowest mertic level. (default = "white")

colour_high_mertic
Color for highest mertic level. (default = "navy")

Value

The cell annotation picture is saved in "save_path".

See Also

Other Section_5_Other_Functions_Provided: Celltype_Compare(), Celltype_annotation_Cellmarker2(), Celltype_annotation_Excel(), Celltype_annotation_Seurat(), Compute_Gene_AUC_ROC(), Compute_Gene_PRC(), Plot_Group_Gene_Heatmap(), Plot_Hierarchy_Proportion(), Plot_Voronoi_diagram(), paletteDiscrete()

Examples

```
## Not run:
Celltype_annotation_PanglaoDB(seurat_obj = sce,
    gene_list = Markers_list_panglaoDB,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_PanglaoDB")
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

## End(Not run)
```

Celltype_Annotation_PerCell

Annotate Seurat Object with Per-Cell SlimR Predictions

Description

This function assigns SlimR per-cell predicted cell types directly to individual cells in a Seurat object's meta.data slot.

Usage

```
Celltype_Annotation_PerCell(
  seurat_obj,
  SlimR_percell_result,
  plot_UMAP = TRUE,
  annotation_col = "Cell_type_PerCell_SlimR",
  plot_confidence = FALSE
)
```

Arguments

seurat_obj	A Seurat object.
SlimR_percell_result	List generated by Celltype_Calculate_PerCell() containing Cell_annotations data.frame with Cell_barcode and Predicted_cell_type columns.
plot_UMAP	Logical; if TRUE, plot the UMAP with cell type annotations. Default: TRUE.
annotation_col	Column name to write in meta.data. Default: "Cell_type_PerCell_SlimR".
plot_confidence	Logical; if TRUE, also plot a UMAP colored by confidence scores. Default: FALSE.

Value

A Seurat object with updated meta.data containing:

- annotation_col: Predicted cell type for each cell
- paste0(annotation_col, "_score"): Max score for each cell
- paste0(annotation_col, "_confidence"): Confidence score for each cell

Note

If plot_UMAP = TRUE, this function will print UMAP plot(s) as a side effect.

See Also

Other Section_3_Automated_Annotation: Celltype_Annotation(), Celltype_Calculate(), Celltype_Calculate_Per(
Celltype_Verification(), Celltype_Verification_PerCell(), Parameter_Calculate(), percell_workflow

Examples

```
## Not run:
# Run per-cell annotation
result <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human"
)
```

```
# Annotate Seurat object
sce <- Celltype_Annotation_PerCell(
    seurat_obj = sce,
    SlimR_percell_result = result,
    plot_UMAP = TRUE,
    annotation_col = "Cell_type_PerCell_SlimR"
)

## End(Not run)
```

`Celltype_annotation_Seurat` *Uses "marker_list" from Seurat object for cell annotation*

Description

Uses "marker_list" from Seurat object for cell annotation

Usage

```
Celltype_annotation_Seurat(
  seurat_obj,
  gene_list,
  species,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  save_path = NULL,
  metric_names = NULL,
  colour_low = "white",
  colour_high = "navy",
  colour_low_mertic = "white",
  colour_high_mertic = "navy"
)
```

Arguments

`seurat_obj`	Enter the Seurat object with annotation columns such as "seurat_cluster" in meta.data to be annotated.
`gene_list`	Enter the standard "Marker_list" generated by the Seurat object database for the SlimR package, generated by the "read_seurat_markers()" function.
`species`	This parameter selects the species "Human" or "Mouse" for standard gene format correction of markers entered by "Marker_list".
`cluster_col`	Enter annotation columns such as "seurat_cluster" in meta.data of the Seurat object to be annotated. Default parameters use "cluster_col = 'seurat_clusters'".
`assay`	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = 'RNA'".

save_path	The output path of the cell annotation picture. Example parameters use "save_path = ’./SlimR/Celltype_annotation_Seurat/’".
metric_names	Change the row name for the input mertics, not recommended unless necessary. (NULL is used as default parameter)
colour_low	Color for lowest expression level. (default = "white")
colour_high	Color for highest expression level. (default = "navy")
colour_low_mertic	Color for lowest mertic level. (default = "white")
colour_high_mertic	Color for highest mertic level. (default = "navy")

Value

The cell annotation picture is saved in "save_path".

See Also

Other Section_5_Other_Functions_Provided: Celltype_Compare(), Celltype_annotation_Cellmarker2(), Celltype_annotation_Excel(), Celltype_annotation_PanglaoDB(), Compute_Gene_AUC_ROC(), Compute_Gene_PRC(), Plot_Group_Gene_Heatmap(), Plot_Hierarchy_Proportion(), Plot_Voronoi_diagram(), paletteDiscrete()

Examples

```
## Not run:
Celltype_annotation_Seurat(seurat_obj = sce,
    gene_list = Markers_list_Seurat,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    save_path = file.path(tempdir(),"SlimR_Celltype_annotation_Seurat")
    colour_low = "white",
    colour_high = "navy",
    colour_low_mertic = "white",
    colour_high_mertic = "navy",
    )

## End(Not run)
```

Celltype_Calculate *Uses "marker_list" to calculate probability, prediction results, AUC and generate heatmap for cell annotation*

Description

Uses "marker_list" to calculate probability, prediction results, AUC and generate heatmap for cell annotation

Usage

```
Celltype_Calculate(
  seurat_obj,
  gene_list,
  species,
  cluster_col = "seurat_clusters",
  assay = "RNA",
  min_expression = 0.1,
  specificity_weight = 3,
  threshold = 0.6,
  compute_AUC = TRUE,
  plot_AUC = TRUE,
  AUC_correction = FALSE,
  colour_low = "navy",
  colour_high = "firebrick3"
)
```

Arguments

seurat_obj	Enter the Seurat object with annotation columns such as "seurat_cluster" in meta.data to be annotated.
gene_list	A list of cells and corresponding gene controls, the name of the list is cell type, and the first column of the list corresponds to markers. Lists can be generated using functions such as "Markers_filter_Cellmarker2 ()", "Markers_filter_PanglaoDB ()", "read_excel_markers ()", "read_seurat_markers ()", etc.
species	This parameter selects the species "Human" or "Mouse" for standard gene format correction of markers entered by "Marker_list".
cluster_col	Enter annotation columns such as "seurat_cluster" in meta.data of the Seurat object to be annotated. Default parameters use "cluster_col = 'seurat_clusters'".
assay	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = 'RNA'".
min_expression	The min_expression parameter defines a threshold value to determine whether a cell's expression of a feature is considered "expressed" or not. It is used to filter out low-expression cells that may contribute noise to the analysis. Default parameters use "min_expression = 0.1".
specificity_weight	The specificity_weight parameter controls how much the expression variability (standard deviation) of a feature within a cluster contributes to its "specificity score." It amplifies or suppresses the impact of variability in the final score calculation.Default parameters use "specificity_weight = 3".
threshold	This parameter refers to the normalized similarity between the "alternative cell type" and the "predicted cell type" in the returned results. (the default parameter is 0.6)
compute_AUC	Logical indicating whether to calculate AUC values for predicted cell types. AUC measures how well the marker genes distinguish the cluster from others. When TRUE, adds an AUC column to the prediction results. (default: TRUE)

plot_AUC	The logic indicates whether to draw an AUC curve for the predicted cell type. When TRUE, add an AUC_plot to result. (default: TRUE)
AUC_correction	Logical value controlling AUC-based correction. (default = FALSE) When set to TRUE: 1.Computes AUC values for candidate cell types. (probability > threshold) 2.Selects the cell type with the highest AUC as the final predicted type. 3.Records the selected type's AUC value in the "AUC" column.
colour_low	Color for lowest probability level in Heatmap visualization of probability matrix. (default = "navy")
colour_high	Color for highest probability level Heatmap visualization of probability matrix. (default = "firebrick3")

Value

A list containing:

- Expression_list: List of expression matrices for each cell type
- Proportion_list: List of proportion of expression for each cell type
- Expression_scores_matrix: Matrix of expression scores
- Probability_matrix: Matrix of normalized probabilities
- Prediction_results: Data frame with cluster annotations including:
  - cluster_col: Cluster identifier
  - Predicted_cell_type: Primary predicted cell type
  - AUC: Area Under the Curve value (when compute_AUC = TRUE)
  - Alternative_cell_types: Semi-colon separated alternative cell types
- Heatmap_plot: Heatmap visualization of probability matrix (pheatmap object). Can be displayed using `print()` or `plot()`
- AUC_plot: AUC visualization of Predicted cell type (ggplot object)
- AUC_list: The resulting list of AUC values calculated for genes in alternative cell types above the approximate threshold

See Also

Other Section_3_Automated_Annotation: `Celltype_Annotation()`, `Celltype_Annotation_PerCell()`, `Celltype_Calculate_PerCell()`, `Celltype_Verification()`, `Celltype_Verification_PerCell()`, `Parameter_Calculate()`, `percell_workflow`

Examples

```
## Not run:
SlimR_anno_result <- Celltype_Calculate(seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    cluster_col = "seurat_clusters",
    assay = "RNA",
    min_expression = 0.1,
    specificity_weight = 3,
```

```
    threshold = 0.6,
    compute_AUC = TRUE,
    plot_AUC = TRUE,
    AUC_correction = FALSE,
    colour_low = "navy",
    colour_high = "firebrick3"
    )

## End(Not run)
```

`Celltype_Calculate_PerCell` *Per-cell annotation using marker expression and optional UMAP spatial smoothing*

Description

Unlike cluster-based annotation, this function assigns cell type labels to each individual cell based on marker gene expression profiles. Optionally uses UMAP coordinates to smooth predictions via k-nearest neighbor voting.

Usage

```
Celltype_Calculate_PerCell(
  seurat_obj,
  gene_list,
  species,
  assay = "RNA",
  method = c("weighted", "mean", "AUCell"),
  min_expression = 0.1,
  use_umap_smoothing = FALSE,
  umap_reduction = "umap",
  k_neighbors = 15,
  smoothing_weight = 0.3,
  min_score = "auto",
  min_confidence = 1.2,
  return_scores = FALSE,
  ncores = 1,
  chunk_size = 5000,
  verbose = TRUE
)
```

Arguments

`seurat_obj`	Seurat object with normalized expression data.
`gene_list`	A standardized marker list (same format as Celltype_Calculate).

species	"Human" or "Mouse" for gene name formatting.
assay	Assay to use (default: "RNA").
method	Scoring method: "AUCell" (rank-based), "mean" (average expression), or "weighted" (expression * detection weighted). Default: "weighted".
min_expression	Minimum expression threshold for detection. Default: 0.1.
use_umap_smoothing	Logical. If TRUE, apply k-NN smoothing using UMAP coordinates to improve annotation consistency. Default: FALSE.
umap_reduction	Name of UMAP reduction in Seurat object. Default: "umap".
k_neighbors	Number of neighbors for UMAP smoothing. Default: 15.
smoothing_weight	Weight for neighbor votes vs cell's own score (0-1). Higher values give more weight to neighbors. Default: 0.3.
min_score	Minimum score threshold to assign a cell type. Cells below this threshold are labeled "Unassigned". Default: "auto" which adaptively sets the threshold based on number of cell types (1.5 / n_celltypes). Set to a numeric value (e.g., 0.1) to use a fixed threshold.
min_confidence	Minimum confidence threshold. Cells with confidence below this value are labeled "Unassigned". Confidence is calculated as the ratio of max score to second-highest score. Default: 1.2 (max must be 20% higher than second). Set to 1.0 to disable confidence filtering.
return_scores	If TRUE, return full score matrix. Default: FALSE.
ncores	Number of cores for parallel processing. Default: 1.
chunk_size	Number of cells to process per chunk (memory optimization). Default: 5000.
verbose	Print progress messages. Default: TRUE.

Details

Scoring Methods:

"weighted" (recommended): Combines normalized expression with detection rate. For each cell and cell type: score = mean(expr_i * weight_i) where weight_i is derived from the marker's specificity across the dataset.

"mean": Simple average of normalized marker expression. Fast but less discriminative for overlapping marker sets.

"AUCell": Rank-based scoring similar to AUCell package. For each cell, genes are ranked by expression, and the score is the proportion of marker genes in the top X% of expressed genes. Robust to technical variation.

UMAP Smoothing:

When `use_umap_smoothing = TRUE`, the function:

1. Computes initial per-cell scores
2. Finds k nearest neighbors in UMAP space for each cell
3. Smooths scores by weighted averaging with neighbors
4. Re-assigns cell types based on smoothed scores

This helps reduce noise and improve consistency of annotations within spatially coherent regions.

Value

A list containing:

- Cell_annotations: Data frame with Cell_barcode, Predicted_cell_type, Max_score, Confidence
- Cell_confidence: Numeric vector of confidence scores per cell
- Summary: Summary table of cell type counts and percentages
- Expression_list: List of mean expression matrices per cell type (for verification)
- Proportion_list: List of detection proportion matrices per cell type
- Prediction_results: Summary data frame with per-cell-type statistics
- Probability_matrix: Full cell × cell_type probability matrix (normalized)
- Raw_score_matrix: Full cell × cell_type raw score matrix (before normalization)
- Parameters: List of parameters used including adaptive thresholds
- Cell_scores: (if return_scores=TRUE) Same as Probability_matrix

See Also

Other Section_3_Automated_Annotation: Celltype_Annotation(), Celltype_Annotation_PerCell(), Celltype_Calculate(), Celltype_Verification(), Celltype_Verification_PerCell(), Parameter_Calculate(), percell_workflow

Examples

```
## Not run:
# Basic per-cell annotation
result <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    method = "weighted"
)

# Add annotations to Seurat object
sce$Cell_type_PerCell <- result$Cell_annotations$Predicted_cell_type

# With UMAP smoothing for more consistent annotations
result_smooth <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    use_umap_smoothing = TRUE,
    k_neighbors = 20,
    smoothing_weight = 0.3
)

## End(Not run)
```

`Celltype_Compare` *Compare cell type labels across two single-cell datasets after aligning cell barcodes*

Description

This function automatically aligns cell barcodes between two Seurat objects using a variety of normalization transformations, then cross-tabulates a cell type label column (from the first object) against a grouping column (from the second object). It returns count tables, proportion tables, a dominant mapping, and a heatmap.

Usage

```
Celltype_Compare(
  sce_label,
  sce,
  label_col = NULL,
  group_col = NULL,
  barcode_col = NULL,
  color_low = "grey70",
  color_high = "navy",
  show_plot = TRUE
)
```

Arguments

`sce_label`	A Seurat object containing the cell type label column.
`sce`	A Seurat object containing the grouping column.
`label_col`	Character. Name of the metadata column in `sce_label` that stores cell type labels (e.g., "Sub_cell_type").
`group_col`	Character. Name of the metadata column in `sce` that stores grouping information (e.g., "SCT_snn_res.0.3").
`barcode_col`	Optional character. Name of a metadata column in both objects that contains the cell barcode identifiers. If `NULL`, the function uses `colnames(sce_label)` and `colnames(sce)`.
`color_low`	Character. Color for low proportion values in the heatmap. Default: "grey70".
`color_high`	Character. Color for high proportion values in the heatmap. Default: "navy".
`show_plot`	Logical. If `TRUE` (default), the heatmap is automatically displayed when the function is called. Set to `FALSE` to suppress automatic plotting (e.g., in non-interactive environments).

Details

Cell barcode alignment: The function automatically tries a set of normalization functions on the cell identifiers (either from `barcode_col` or from column names) to maximise the number of shared barcodes between the two objects. Transformations include: `identity`, `drop_numeric_suffix` (removes e.g., "-1-2"), `drop_suffix` (removes "-1"), and several prefix removals. The transformation pair yielding the highest number of shared identifiers is selected.

Proportion calculation: Proportions are computed **within each** `group_col` **level** (column-wise), i.e. for each group, the sum of proportions across all cell types equals 1.

Plot: The heatmap uses `ggplot2::geom_tile()` with a fixed coordinate ratio and a colour gradient from `color_low` to `color_high`.

Value

A list with five components:

`count_table`	A data frame (wide format) with rows = unique `label_col` values and columns = unique `group_col` values; cell values are raw counts of shared cells.
`prop_table`	Same shape as `count_table`; each cell shows the proportion of cells within a `group_col` column (column-wise sum = 1).
`main_to_sub`	A data frame mapping each `group_col` value to the most frequent `label_col` value among shared cells.
`plot`	A ggplot2 heatmap object visualizing the proportion table.
`match_info`	A tibble with columns `label_transform`, `sce_transform`, `shared_n` – the transformations used to align barcodes and the number of shared cells after alignment.

See Also

Other Section_5_Other_Functions_Provided: `Celltype_annotation_Cellmarker2()`, `Celltype_annotation_Excel()`, `Celltype_annotation_PanglaoDB()`, `Celltype_annotation_Seurat()`, `Compute_Gene_AUC_ROC()`, `Compute_Gene_PRC()`, `Plot_Group_Gene_Heatmap()`, `Plot_Hierarchy_Proportion()`, `Plot_Voronoi_diagram()`, `paletteDiscrete()`

Examples

```
## Not run:
# Basic usage with two Seurat objects and default barcode alignment
result <- Celltype_Compare(
  sce_label = seurat_obj1,
  sce = seurat_obj2,
  label_col = "cell_type",
  group_col = "cluster"
)

# Access the proportion table
head(result$prop_table)

# View the dominant mapping
```

```
print(result$main_to_sub)

# Display the heatmap
print(result$plot)

# Use a custom barcode column
result2 <- Celltype_Compare(
  sce_label = seurat_obj1,
  sce = seurat_obj2,
  label_col = "cell_type",
  group_col = "cluster",
  barcode_col = "cell_barcode"
)

## End(Not run)
```

`Celltype_Verification` *Perform cell type verification and generate the validation dotplot*

Description

This function performs verification of predicted cell types by selecting high log2FC and high expression proportion genes and generates and generate the validation dotplot.

Usage

```
Celltype_Verification(
  seurat_obj,
  SlimR_anno_result,
  assay = "RNA",
  gene_number = 5,
  colour_low = "white",
  colour_high = "navy",
  annotation_col = "Cell_type_SlimR"
)
```

Arguments

`seurat_obj`	A Seurat object containing single-cell data.
`SlimR_anno_result`	A list containing SlimR annotation results with: Expression_list - List of expression matrices for each cell type. Prediction_results - Data frame with cluster annotations.
`assay`	Enter the assay used by the Seurat object, such as "RNA". Default parameters use "assay = 'RNA'".
`gene_number`	Integer specifying number of top genes to select per cell type.

colour_low	Color for lowest expression level. (default = "white")
colour_high	Color for highest expression level. (default = "navy")
annotation_col	Character string specifying the column in meta.data to use for grouping.

Value

A ggplot object showing expression of top variable genes.

See Also

Other Section_3_Automated_Annotation: Celltype_Annotation(), Celltype_Annotation_PerCell(), Celltype_Calculate(), Celltype_Calculate_PerCell(), Celltype_Verification_PerCell(), Parameter_Calculate(), percell_workflow

Examples

```
## Not run:
Celltype_Verification(seurat_obj = sce,
    SlimR_anno_result = SlimR_anno_result,
    assay = "RNA",
    gene_number = 5,
    colour_low = "white",
    colour_high = "navy",
    annotation_col = "Cell_type_SlimR"
    )

## End(Not run)
```

Celltype_Verification_PerCell

Verify per-cell annotations with marker expression dotplot

Description

This function verifies per-cell SlimR annotations by generating a dotplot showing marker gene expression across predicted cell types.

Usage

```
Celltype_Verification_PerCell(
  seurat_obj,
  SlimR_percell_result,
  assay = "RNA",
  gene_number = 5,
  colour_low = "white",
  colour_high = "navy",
  annotation_col = "Cell_type_PerCell_SlimR",
  min_cells = 10
)
```

Arguments

seurat_obj	A Seurat object with per-cell annotations.
SlimR_percell_result	A list from Celltype_Calculate_PerCell() containing Expression_list with marker genes per cell type.
assay	Assay to use. Default: "RNA".
gene_number	Number of top genes to show per cell type. Default: 5.
colour_low	Color for lowest expression. Default: "white".
colour_high	Color for highest expression. Default: "navy".
annotation_col	Column in meta.data with cell type annotations. Default: "Cell_type_PerCell_SlimR".
min_cells	Minimum number of cells required for a cell type to be included in the plot. Default: 10.

Value

A ggplot object showing marker gene expression dotplot.

See Also

Other Section_3_Automated_Annotation: Celltype_Annotation(), Celltype_Annotation_PerCell(), Celltype_Calculate(), Celltype_Calculate_PerCell(), Celltype_Verification(), Parameter_Calculate(), percell_workflow

Examples

```
## Not run:
# After running Celltype_Calculate_PerCell and Celltype_Annotation_PerCell
dotplot <- Celltype_Verification_PerCell(
    seurat_obj = sce,
    SlimR_percell_result = result,
    gene_number = 5,
    annotation_col = "Cell_type_PerCell_SlimR"
)
print(dotplot)

## End(Not run)
```

CellTypist *List of cell type markers from the CellTypist organ atlas*

Description

A dataset containing marker genes for human cell types across 12 organs, derived from the CellTypist online resource.

Usage

```
CellTypist
```

Format

A named list with 12 elements, each corresponding to an organ: Blood, Bone_marrow, Heart, Hippocampus, Intestine, Kidney, Liver, Lung, Lymph_node, Pancreas, Skeletal_muscle, Spleen. Each organ element is a list of data frames (one per cell type), containing marker genes and associated statistics. In total, 399 cell types are represented.

Details

This list comprises markers for 399 cell types across 12 human organs, obtained from the CellTypist organ atlas (https://www.celltypist.org/organs). Marker genes were generated using a standard Scanpy workflow: log1p-normalised data, `sc.tl.rank_genes_groups` (Wilcoxon test), adjusted p-value < 0.01 and log2 fold-change > 0, ranked by log fold-change and cut to the top 100 genes per cluster. The resulting Excel file was imported with `Read_excel_markers`. Please cite: Xu et al. (2023) doi:10.1016/j.cell.2023.11.026 and Domínguez Conde et al. (2022) doi:10.1126/science.abl5197.

Source

doi:10.1016/j.cell.2023.11.026

doi:10.1126/science.abl5197

See Also

Other Section_0_Database: `Cellmarker2`, `Cellmarker2_raw`, `Cellmarker2_table`, `Markers_list_PCTAM`, `Markers_list_PCTIT`, `Markers_list_TCellSI`, `Markers_list_scIBD`, `PanglaoDB`, `PanglaoDB_raw`, `PanglaoDB_table`, `ScType`, `ScType_raw`, `ScType_table`, `scBaseCount`

---

compute_adaptive_parameters

Compute Adaptive Parameters Based on Dataset Features (Use in package)

Description

Calculates optimal min_expression, specificity_weight, and threshold parameters using continuous adaptive algorithms based on dataset characteristics.

Usage

```
compute_adaptive_parameters(dataset_features, n_celltypes = 50)
```

Arguments

dataset_features	List of dataset characteristics from extract_dataset_features()
n_celltypes	Expected number of cell types in marker database

Value

List containing min_expression, specificity_weight, threshold, and rationale

See Also

Other Section_1_Functions_Use_in_Package: calculate_cluster_variability(), calculate_expression(), calculate_expression_skewness(), calculate_probability(), estimate_batch_effect(), extract_dataset_features()

Compute_Gene_AUC_ROC *Compute AUC and Optionally Plot ROC Curve for a Single Gene*

Description

Evaluates the discriminatory power of a single gene in separating a user-defined positive cell group from the rest, using the Area Under the Receiver Operating Characteristic curve (AUC). Several scoring strategies and optional cell subsetting are provided to handle dropout noise and sparse expression typical of single-cell data. If `plot = TRUE` (the default), a publication-ready ROC curve is automatically displayed via **ggplot2**.

Usage

```
Compute_Gene_AUC_ROC(
  seurat_obj,
  gene,
  group_col,
  group_label,
  assay = "RNA",
  method = c("raw", "rank"),
  min_expression = NULL,
  keep_expression_above = NULL,
  plot = TRUE,
  plot_title = "Gene ROC Curve | SlimR",
  line_color = "navy",
  line_size = 1
)
```

Arguments

seurat_obj	A Seurat object containing single-cell expression data.
gene	A single character string specifying the gene to evaluate. Must be present in the rownames of the chosen assay.
group_col	Character string giving the name of a column in seurat_obj@meta.data that defines cell groups (e.g., "seurat_clusters").
group_label	The value in group_col that defines the positive class (e.g., 1 for cluster 1). All other cells are treated as negatives.
assay	Character string specifying which assay to use. Default is "RNA".
method	Scoring method: "raw" (raw expression, optionally truncated by min_expression) or "rank" (expression ranks, which is robust to dropout and does not require min_expression).
min_expression	Numeric threshold for expression truncation (only used when method = "raw"). Values below this threshold are set to zero. Default is NULL (no truncation).
keep_expression_above	Optional numeric threshold. If set, only cells with expression greater than this value are retained for AUC computation. **Note:** this changes the population over which AUC is evaluated, shifting the interpretation from "overall discrimination among all cells" to "discrimination among cells that express the gene above this level". Use with caution and always compare with the result without subsetting. Default is NULL (all cells retained).
plot	Logical indicating whether to create and automatically display a ggplot2 ROC curve. Default is TRUE.
plot_title	Character string used as the title of the ROC plot. Default is "ROC Curve | SlimR".
line_color	Colour of the ROC curve. Default is "navy".
line_size	Numeric value for the thickness of the ROC curve line. Default is 1.

Value

Invisibly, a list with the following elements:

AUC Numeric value (between 0 and 1) of the area under the ROC curve.

roc_data A data.frame with columns fpr (False Positive Rate) and tpr (True Positive Rate) that can be used for custom plotting.

predictions Numeric vector of the (possibly truncated) expression values used as prediction scores.

labels Logical vector indicating whether each cell belongs to the positive class (TRUE) or not (FALSE).

roc_plot If plot = TRUE, a **ggplot** object displaying the ROC curve; otherwise NULL.

See Also

Other Section_5_Other_Functions_Provided: Celltype_Compare(), Celltype_annotation_Cellmarker2(), Celltype_annotation_Excel(), Celltype_annotation_PanglaoDB(), Celltype_annotation_Seurat(), Compute_Gene_PRC(), Plot_Group_Gene_Heatmap(), Plot_Hierarchy_Proportion(), Plot_Voronoi_diagram(), paletteDiscrete()

Examples

```
## Not run:
# Default: raw expression, plot displayed automatically
res <- Compute_Gene_AUC_ROC(sce, "CD3D", "seurat_clusters", 12)

# Rank-based (robust to dropout)
res_rank <- Compute_Gene_AUC_ROC(sce, "CD3D", "seurat_clusters", 12,
                                 method = "rank")

# Keep only cells expressing above 0.5 (exploratory)
res_sub <- Compute_Gene_AUC_ROC(sce, "CD3D", "seurat_clusters", 12,
                                keep_expression_above = 0.5)

## End(Not run)
```

Compute_Gene_PRC *Compute PR-AUC and Optionally Plot Precision-Recall Curve for a Single Gene*

Description

Evaluates the discriminatory power of a single gene in separating a user-defined positive cell group from the rest, using the Area Under the Precision-Recall Curve (PR-AUC), reported here as Average Precision (AP). Unlike ROC-AUC, which is insensitive to class balance, PR-AUC summarises the trade-off between precision and recall and is therefore better suited to highly imbalanced settings—such as identifying a rare cell type among thousands of cells—where the positive class is the minority. Several scoring strategies and optional cell subsetting are provided to handle dropout noise and sparse expression typical of single-cell data. If `plot = TRUE` (the default), a publication-ready precision-recall curve is automatically displayed via **ggplot2**.

Usage

```
Compute_Gene_PRC(
  seurat_obj,
  gene,
  group_col,
  group_label,
  assay = "RNA",
  method = c("raw", "rank"),
  min_expression = NULL,
  keep_expression_above = NULL,
  plot = TRUE,
  plot_title = "Gene Precision-Recall Curve | SlimR",
  line_color = "darkmagenta",
  line_size = 1
)
```

Arguments

seurat_obj	A Seurat object containing single-cell expression data.
gene	A single character string specifying the gene to evaluate. Must be present in the rownames of the chosen assay.
group_col	Character string giving the name of a column in seurat_obj@meta.data that defines cell groups (e.g., "seurat_clusters").
group_label	The value in group_col that defines the positive class (e.g., 1 for cluster 1). All other cells are treated as negatives.
assay	Character string specifying which assay to use. Default is "RNA".
method	Scoring method: "raw" (raw expression, optionally truncated by min_expression) or "rank" (expression ranks, which is robust to dropout and does not require min_expression).
min_expression	Numeric threshold for expression truncation (only used when method = "raw"). Values below this threshold are set to zero. Default is NULL (no truncation).
keep_expression_above	Optional numeric threshold. If set, only cells with expression greater than this value are retained for PR-AUC computation. **Note:** this changes the population over which PR-AUC is evaluated, shifting the interpretation from "overall discrimination among all cells" to "discrimination among cells that express the gene above this level". Because the no-skill baseline is the prevalence of the positive class, subsetting also changes the baseline. Use with caution and always compare with the result without subsetting. Default is NULL (all cells retained).
plot	Logical indicating whether to create and automatically display a ggplot2 precision-recall curve. Default is TRUE.
plot_title	Character string used as the title of the PR plot. Default is "Gene Precision-Recall Curve | SlimR".
line_color	Colour of the PR curve. Default is "darkmagenta".
line_size	Numeric value for the thickness of the PR curve line. Default is 1.

Details

The difference between ROC-AUC and PR-AUC is important when interpreting the results:

- **ROC-AUC** plots the True Positive Rate against the False Positive Rate. It is insensitive to class imbalance and has a fixed no-skill baseline of 0.5.
- **PR-AUC** plots Precision (Positive Predictive Value) against Recall (Sensitivity). It focuses on the performance on the positive class and has a no-skill baseline equal to the prevalence of the positive class. For rare cell populations this baseline is close to zero, so PR-AUC provides a more informative and more stringent measure of discriminatory power than ROC-AUC.

PR-AUC is computed here as Average Precision (AP), i.e. the mean of the precision values obtained at each positive cell, weighted by the corresponding increase in recall:

$$AP = \sum_{k \in \text{positives}} P(k)\,\Delta R(k)$$

where the sum runs over all cells in the positive class. This definition is equivalent to `sklearn.metrics.average_precisior` and avoids the systematic underestimation that a naive trapezoidal integration would introduce on the non-monotonic PR curve. The no-skill `Baseline` is returned alongside `PR_AUC` because the two must be interpreted together: a PR-AUC of, say, 0.55 is informative when the baseline is 0.02 but nearly uninformative when the baseline is 0.50.

Value

Invisibly, a list with the following elements:

PR_AUC Numeric value (between 0 and 1) of the area under the precision-recall curve, computed as Average Precision.

Baseline Numeric value of the no-skill baseline, equal to the prevalence of the positive class. A PR-AUC above this value indicates better-than-random discrimination; a value close to the baseline indicates no discriminatory power.

pr_data A data.frame with columns `recall` and `precision` that can be used for custom plotting.

predictions Numeric vector of the (possibly truncated) expression values used as prediction scores.

labels Logical vector indicating whether each cell belongs to the positive class (`TRUE`) or not (`FALSE`).

pr_plot If `plot = TRUE`, a **ggplot2** object displaying the precision-recall curve; otherwise `NULL`.

See Also

Other Section_5_Other_Functions_Provided: `Celltype_Compare()`, `Celltype_annotation_Cellmarker2()`, `Celltype_annotation_Excel()`, `Celltype_annotation_PanglaoDB()`, `Celltype_annotation_Seurat()`, `Compute_Gene_AUC_ROC()`, `Plot_Group_Gene_Heatmap()`, `Plot_Hierarchy_Proportion()`, `Plot_Voronoi_diagram()`, `paletteDiscrete()`

Examples

```
## Not run:
# Default: raw expression, plot displayed automatically
res <- Compute_Gene_PRC(sce, "DCLK1", "seurat_clusters", 12)

# Rank-based (robust to dropout)
res_rank <- Compute_Gene_PRC(sce, "DCLK1", "seurat_clusters", 12,
                             method = "rank")

# Keep only cells expressing above 0.5 (exploratory)
res_sub <- Compute_Gene_PRC(sce, "DCLK1", "seurat_clusters", 12,
                            keep_expression_above = 0.5)

## End(Not run)
```

estimate_batch_effect *Estimate Batch Effect Strength (Use in package)*

Description

Roughly estimates the potential impact of batch effects using available metadata.

Usage

```
estimate_batch_effect(seurat_obj, assay)
```

Arguments

seurat_obj	Seurat object
assay	Assay name

Value

Batch effect score (0 indicates no detectable batch effect)

See Also

Other Section_1_Functions_Use_in_Package: calculate_cluster_variability(), calculate_expression(), calculate_expression_skewness(), calculate_probability(), compute_adaptive_parameters(), extract_dataset_features()

extract_dataset_features *Extract Dataset Characteristics for Adaptive Parameter Calculation (Use in package)*

Description

Computes various statistical features from single-cell data that are used as input for the parameter prediction model.

Usage

```
extract_dataset_features(
  seurat_obj,
  features,
  assay = NULL,
  cluster_col = NULL
)
```

Arguments

seurat_obj	Seurat object
features	Features to analyze
assay	Assay name
cluster_col	Cluster column name

Value

List of dataset characteristics including expression statistics, variability measures, and cluster properties

See Also

Other Section_1_Functions_Use_in_Package: calculate_cluster_variability(), calculate_expression(), calculate_expression_skewness(), calculate_probability(), compute_adaptive_parameters(), estimate_batch_effect()

Markers_filter_Cellmarker2

Create Marker_list from the Cellmarkers2 database

Description

Create Marker_list from the Cellmarkers2 database

Usage

```
Markers_filter_Cellmarker2(
  df,
  species = NULL,
  tissue_class = NULL,
  tissue_type = NULL,
  cancer_type = NULL,
  cell_type = NULL
)
```

Arguments

df	Standardized Cellmarkers2 database. It is read as data(Cellmarkers2) in the SlimR library.
species	Species information in Cellmarkers2 database. The default input is "Human" or "Mouse".The input can be retrieved by "Cellmarkers2_table". For more information,please refer to http://117.50.127.228/CellMarker/ on Cellmarkers2's official website.

tissue_class	Tissue_class information in Cellmarkers2 database. The input can be retrieved by "Cellmarkers2_table". For more information, please refer to http://117.50.127.228/CellMarker/ on Cellmarkers2's official website.
tissue_type	Tissue_type information in Cellmarkers2 database. The input can be retrieved by "Cellmarkers2_table". For more information, please refer to http://117.50.127.228/CellMarker/ on Cellmarkers2's official website.
cancer_type	Cancer_type information in Cellmarkers2 database. The input can be retrieved by "Cellmarkers2_table". For more information, please refer to http://117.50.127.228/CellMarker/ on Cellmarkers2's official website.
cell_type	Cell_type information in Cellmarkers2 database. The input can be retrieved by "Cellmarkers2_table". For more information, please refer to http://117.50.127.228/CellMarker/ on Cellmarkers2's official website.

Value

The standardized "Marker_list" in the SlimR package

See Also

Other Section_2_Standardized_Markers_List: `Markers_filter_PanglaoDB()`, `Markers_filter_ScType()`, `Read_excel_markers()`, `Read_seurat_markers()`

Examples

```
Cellmarker2 <- SlimR::Cellmarker2
Markers_list_Cellmarker2 <- Markers_filter_Cellmarker2(
    Cellmarker2,
    species = "Human",
    tissue_class = "Intestine",
    tissue_type = NULL,
    cancer_type = NULL,
    cell_type = NULL
    )
```

Markers_filter_PanglaoDB

Create Marker_list from the PanglaoDB database

Description

Create Marker_list from the PanglaoDB database

Usage

```
Markers_filter_PanglaoDB(df, species_input, organ_input)
```

Arguments

df	Standardized PanglaoDB database. It is read as data(PanglaoDB) in the SlimR library.
species_input	Species information in PanglaoDB database. The default input is "Human" or "Mouse".The input can be retrieved by "PanglaoDB_table". For more information,please refer to https://panglaodb.se/ on PanglaoDB's official website.
organ_input	Organ type information in the PanglaoDB database. The input can be retrieved by "PanglaoDB_table".For more information, please refer to https://panglaodb.se/ on PanglaoDB's official website.

Value

The standardized "Marker_list" in the SlimR package

See Also

Other Section_2_Standardized_Markers_List: Markers_filter_Cellmarker2(), Markers_filter_ScType(), Read_excel_markers(), Read_seurat_markers()

Examples

```
PanglaoDB <- SlimR::PanglaoDB
Markers_list_panglaoDB <- Markers_filter_PanglaoDB(
    PanglaoDB,
    species_input = 'Human',
    organ_input = 'GI tract'
    )
```

Markers_filter_ScType *Create Marker_list from the ScType database*

Description

Create Marker_list from the ScType database

Usage

```
Markers_filter_ScType(df, tissue_type = NULL, cell_name = NULL)
```

Arguments

df	Standardized ScType database. It is read as data(ScType) in the SlimR library.
tissue_type	Tissue type information in the ScType database. The input can be retrieved by "ScType_table". For more information, please refer to https://github.com/IanevskiAleksandr/sc-type.
cell_name	Cell type name information in the ScType database. The input can be retrieved by "ScType_table". For more information, please refer to https://github.com/IanevskiAleksandr/sc-type.

Value

The standardized "Marker_list" in the SlimR package

See Also

Other Section_2_Standardized_Markers_List: `Markers_filter_Cellmarker2()`, `Markers_filter_PanglaoDB()`, `Read_excel_markers()`, `Read_seurat_markers()`

Examples

```
ScType <- SlimR::ScType
Markers_list_ScType <- Markers_filter_ScType(
    ScType,
    tissue_type = "Immune system",
    cell_name = NULL
    )
```

Markers_list_PCTAM *List of Macrophage subtype markers in the article "Macrophage diversity in cancer revisited in the era of single-cell omics"*

Description

A dataset containing marker genes for different Macrophage subtypes from the article "Macrophage diversity in cancer revisited in the era of single-cell omics"

Usage

```
Markers_list_PCTAM
```

Format

A list with 7 tables.

Details

This list is a table of 7 types of Tumor-associated macrophages (TAMs) markers obtained from the article "Macrophage diversity in cancer revisited in the era of single-cell omics". The data source is "https://doi.org/10.1016/j.it.2022.04.008", and the reference literature is: Ruo-Yu Ma et al. (2022) doi:10.1016/j.it.2022.04.008.

Source

doi:10.1016/j.it.2022.04.008

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

---

`Markers_list_PCTIT` *List of T cell subtype markers in the article "Pan-cancer single cell landscape of tumor-infiltrating T cells"*

Description

A dataset containing marker genes for different T cell types from the article "Pan-cancer single cell landscape of tumor-infiltrating T cells"

Usage

```
Markers_list_PCTIT
```

Format

A list with 40 tables.

Details

This list is a table of 40 types of pan-cancer tumor-infiltrating T cell (PCTIT) markers obtained from the article "Pan-cancer single cell landscapeof tumor-infiltrating T cells". The data source is "https://doi.org/10.1126/science.abe6474", and the reference literature is: L. Zheng et al. (2021) doi:10.1126/science.abe6474.

Source

doi:10.1126/science.abe6474

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

Markers_list_scIBD *List of cell type markers from scIBD*

Description

A dataset containing marker genes for human intestinal cell types derived from scIBD.

Usage

```
Markers_list_scIBD
```

Format

A named list of 101 data frames, each representing one cell type and containing marker genes along with associated statistics.

Details

This list comprises markers for 101 human intestine cell types, obtained from the scIBD study (Nie et al. 2023). Marker genes were generated using a standard Scanpy workflow: log1p-normalised data, `sc.tl.rank_genes_groups` (Wilcoxon test), adjusted p-value < 0.01 and log2 fold-change > 0, ranked by log fold-change and cut to the top 100 genes per cluster (equivalent to `sort_by = "logFC"` and `gene_filter = 100`). The resulting Excel file was imported with `Read_excel_markers`.

Source

doi:10.1038/s43588023004649

See Also

Other Section_0_Database: `CellTypist`, `Cellmarker2`, `Cellmarker2_raw`, `Cellmarker2_table`, `Markers_list_PCTAM`, `Markers_list_PCTIT`, `Markers_list_TCellSI`, `PanglaoDB`, `PanglaoDB_raw`, `PanglaoDB_table`, `ScType`, `ScType_raw`, `ScType_table`, `scBaseCount`

Markers_list_TCellSI *List of T cell subtype markers in the article TCellSI*

Description

A dataset containing marker genes for different T cell subtypes from TCellSI

Usage

```
Markers_list_TCellSI
```

Format

A list with ten tables.

Details

This list is a table of 10 types of T cell markers obtained from TCellSI. The data source is "https://github.com/GuoBioinfoLab/ and the reference literature is: Yang et al. (2024) doi:10.1002/imt2.231.

Source

https://github.com/GuoBioinfoLab/TCellSI/

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

---

PanglaoDB *PanglaoDB dataset*

Description

A dataset containing marker genes for different cell types from PanglaoDB

Usage

```
PanglaoDB
```

Format

A data frame with 9 columns:

Details

This dataset is used to filter and create a standardized marker list.'

Source

https://panglaodb.se/

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

PanglaoDB_raw *PanglaoDB raw dataset*

Description

A dataset containing marker genes for different cell types from PanglaoDB

Usage

```
PanglaoDB_raw
```

Format

A data frame with 14 columns contined in the PanglaoDB database:

Details

This dataset is used to filter and create a standardized marker list.'

Source

https://panglaodb.se/

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_table, ScType, ScType_raw, ScType_table, scBaseCount

PanglaoDB_table *PanglaoDB table*

Description

A dataset containing marker genes for different cell types from PanglaoDB

Usage

```
PanglaoDB_table
```

Format

A list contain different types like species, organ, cell type.

Details

This list is used to choose filters for creation of standardized marker list.

Source

https://panglaodb.se/

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, ScType, ScType_raw, ScType_table, scBaseCount

---

Parameter_Calculate *Adaptive Parameter Tuning for Single-Cell Data Annotation in SlimR*

Description

This function automatically determines optimal min_expression, specificity_weight, and threshold parameters for single-cell data analysis based on dataset characteristics using adaptive algorithms derived from empirical analysis of single-cell datasets.

Usage

```
Parameter_Calculate(
  seurat_obj,
  features = NULL,
  assay = NULL,
  cluster_col = NULL,
  n_celltypes = 50,
  verbose = TRUE
)
```

Arguments

seurat_obj	A Seurat object containing single-cell data
features	Character vector of feature names (genes) to analyze. If NULL, will use highly variable features from the Seurat object.
assay	Name of assay to use (default: default assay)
cluster_col	Column name in metadata containing cluster information
n_celltypes	Expected number of cell types in marker database (default: 50). Used for threshold recommendation calculation.
verbose	Whether to print progress messages (default: TRUE)

Value

A list containing:

- min_expression: Recommended expression threshold
- specificity_weight: Recommended specificity weight
- threshold: Recommended probability threshold for candidate selection
- dataset_features: Extracted dataset characteristics
- parameter_rationale: Explanation of parameter choices

See Also

Other Section_3_Automated_Annotation: Celltype_Annotation(), Celltype_Annotation_PerCell(), Celltype_Calculate(), Celltype_Calculate_PerCell(), Celltype_Verification(), Celltype_Verification_PerC percell_workflow

Examples

```
## Not run:
SlimR_params <- Parameter_Calculate(
  seurat_obj = sce,
  features = c("CD3E", "CD4", "CD8A"),
  assay = "RNA",
  cluster_col = "seurat_clusters",
  n_celltypes = 98,
  verbose = TRUE
  )

## End(Not run)
```

---

percell_workflow *Per-Cell Annotation Workflow Example*

Description

Example workflow for using SlimR's per-cell annotation functions

Overview

The per-cell annotation workflow in SlimR provides an alternative to cluster-based annotation by scoring and labeling individual cells based on marker expression. This is useful when:

- Clusters contain mixed cell types
- You want finer-grained annotations
- Cell states exist on a continuum
- UMAP spatial context can improve annotation quality

Basic Workflow

```
# 1. Prepare your Seurat object (must have normalized data)
library(SlimR)
library(Seurat)

# 2. Create or load marker list
Markers_list <- Markers_filter_Cellmarker2(
    Cellmarker2,
    species = "Human",
    tissue_class = "Intestine"
)

# 3. Run per-cell annotation
result <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    method = "weighted",          # "weighted", "mean", or "AUCell"
    min_expression = 0.1,
    min_score = 0.1,
    verbose = TRUE
)

# 4. Annotate Seurat object
sce <- Celltype_Annotation_PerCell(
    seurat_obj = sce,
    SlimR_percell_result = result,
    plot_UMAP = TRUE,
    plot_confidence = TRUE,
    annotation_col = "Cell_type_PerCell"
)

# 5. Verify annotations
dotplot <- Celltype_Verification_PerCell(
    seurat_obj = sce,
    SlimR_percell_result = result,
    gene_number = 5,
    annotation_col = "Cell_type_PerCell"
)
print(dotplot)
```

Advanced

UMAP Spatial Smoothing:

```
# Use UMAP coordinates to smooth predictions via k-NN
# This reduces noise and improves consistency in spatial regions
```

```
result_smooth <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    use_umap_smoothing = TRUE,
    k_neighbors = 20,              # Number of neighbors to consider
    smoothing_weight = 0.3,        # 30
    verbose = TRUE
)

# Compare smoothed vs unsmoothed
sce$Cell_type_Smooth <- result_smooth$Cell_annotations$Predicted_cell_type
sce$Cell_type_Raw <- result$Cell_annotations$Predicted_cell_type

DimPlot(sce, group.by = "Cell_type_Raw") |
  DimPlot(sce, group.by = "Cell_type_Smooth")
```

Scoring Methods Comparison

```
# Method 1: Weighted (recommended for most cases)
# Combines expression with marker specificity and detection rate
result_weighted <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    method = "weighted"
)

# Method 2: Mean (simple, fast)
# Just averages normalized marker expression
result_mean <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    method = "mean"
)

# Method 3: AUCell (rank-based, robust to batch effects)
# Scores based on proportion of markers in top 5
result_aucell <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    method = "AUCell"
)
```

Comparing Cluster vs Per-Cell Annotation

```
# Cluster-based annotation (original SlimR approach)
```

```
cluster_result <- Celltype_Calculate(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    cluster_col = "seurat_clusters"
)

sce <- Celltype_Annotation(
    seurat_obj = sce,
    cluster_col = "seurat_clusters",
    SlimR_anno_result = cluster_result,
    annotation_col = "Cell_type_Cluster"
)

# Per-cell annotation
percell_result <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human"
)

sce <- Celltype_Annotation_PerCell(
    seurat_obj = sce,
    SlimR_percell_result = percell_result,
    annotation_col = "Cell_type_PerCell"
)

# Compare
library(ggplot2)
library(patchwork)

p1 <- DimPlot(sce, group.by = "Cell_type_Cluster") +
      ggtitle("Cluster-based")
p2 <- DimPlot(sce, group.by = "Cell_type_PerCell") +
      ggtitle("Per-cell")

p1 | p2

# Check agreement
table(sce$Cell_type_Cluster, sce$Cell_type_PerCell)
```

Performance Optimization

```
# For large datasets, adjust chunk_size to manage memory
result <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    chunk_size = 10000,  # Process 10k cells at a time
```

```
    verbose = TRUE
)

# For UMAP smoothing, install RANN for 10-100x speedup
# install.packages("RANN")

result_smooth <- Celltype_Calculate_PerCell(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human",
    use_umap_smoothing = TRUE,
    k_neighbors = 15
    # RANN will be used automatically if installed
)
```

Accessing Results

```
# Cell-level annotations
head(result$Cell_annotations)
#   Cell_barcode Predicted_cell_type Max_score Confidence
# 1  AAACCTGAG... Enterocyte          0.85      0.62
# 2  AAACCTGCA... Goblet cell         0.72      0.45

# Summary statistics
result$Summary
#   Cell_type       Count Percentage
# 1 Enterocyte      5432  45.2
# 2 Goblet cell     2156  17.9

# Full probability matrix (if return_scores = TRUE)
result$Probability_matrix[1:5, 1:3]
#              Enterocyte Goblet_cell Stem_cell
# AAACCTGAG... 0.85       0.10        0.05

# Extract high-confidence cells
high_conf <- result$Cell_annotations$Cell_barcode[
    result$Cell_annotations$Confidence > 0.5
]

# Extract uncertain cells for manual review
uncertain <- result$Cell_annotations$Cell_barcode[
    result$Cell_annotations$Confidence < 0.2
]
```

See Also

Other Section_3_Automated_Annotation: Celltype_Annotation(), Celltype_Annotation_PerCell(), Celltype_Calculate(), Celltype_Calculate_PerCell(), Celltype_Verification(), Celltype_Verification_PerC Parameter_Calculate()

`plot.pheatmap` *Plot Method for pheatmap Objects*

Description

This S3 method allows pheatmap objects (returned by `Celltype_Calculate()`) to be plotted using the generic `plot()` function. Without this method, attempting to use `plot()` on a pheatmap object results in an error.

Usage

```
## S3 method for class 'pheatmap'
plot(x, ...)
```

Arguments

`x`	A pheatmap object, typically from `cluster_results$Heatmap_plot`
`...`	Additional arguments (currently ignored)

Details

Pheatmap objects contain a gtable component that needs to be drawn using grid graphics. This method handles that automatically when `plot()` is called.

Alternative ways to display pheatmaps:

- `print(pheatmap_object)` - Works natively
- `plot(pheatmap_object)` - Works after loading SlimR
- `grid::grid.draw(pheatmap_object$gtable)` - Direct access

Value

Invisibly returns the input pheatmap object after displaying it

Examples

```
## Not run:
# After running Celltype_Calculate()
cluster_results <- Celltype_Calculate(
    seurat_obj = sce,
    gene_list = Markers_list,
    species = "Human"
)

# Now both of these work:
print(cluster_results$Heatmap_plot)
plot(cluster_results$Heatmap_plot)

## End(Not run)
```

`Plot_Group_Gene_Heatmap` *Plot Grouped Gene Z-Score Heatmap with Optional Integrated Score Row*

Description

Generates a publication-quality mean Z-score expression heatmap across cell groups from a Seurat object using **pheatmap**. Optionally computes and appends a top row representing the integrated pathway or marker signature score (mean gene expression across target features per cell) before group mean calculation and Z-score scaling. Row annotations distinguish individual gene expression rows from the optional integrated weighted row. Column annotations and row annotations are automatically styled using SlimR's internal discrete palettes.

Usage

```
Plot_Group_Gene_Heatmap(
  seurat_obj,
  features,
  group_by,
  assay = NULL,
  slot_or_layer = "data",
  add_aucell_row = TRUE,
  aucell_row_name = "Integrated_Score",
  cluster_rows = FALSE,
  cluster_cols = FALSE,
  palette_set = "stallion",
  scale_range = c(-1.25, 1.25),
  annotation_legend = FALSE,
  cellwidth = 18,
  cellheight = 18,
  main = "Group Gene Heatmap Plot | SlimR"
)
```

Arguments

`seurat_obj`	A Seurat object (supports both Seurat v4 and v5).
`features`	A character vector of gene names to display in the heatmap.
`group_by`	Character string specifying the column name in `seurat_obj@meta.data` used to group cells (e.g., `"orig.ident"` or `"celltype"`).
`assay`	Character string specifying the assay to pull data from. If `NULL`, uses `Seurat::DefaultAssay(seurat_ob`
`slot_or_layer`	Character string specifying the layer/slot to extract. Default is `"data"`.
`add_aucell_row`	Logical. If `TRUE` (default), computes an integrated mean expression signature across `features` per cell and appends its group-level mean Z-score as the top row of the heatmap.

aucell_row_name
: Character string naming the top integrated row when add_aucell_row = TRUE. Default is "Integrated_Score".

cluster_rows
: Logical. Whether to perform hierarchical clustering on rows. Default is FALSE to maintain input gene order.

cluster_cols
: Logical. Whether to perform hierarchical clustering on columns. Default is FALSE to preserve original group ordering.

palette_set
: Character string naming the discrete palette set to use for group/type annotation bars (e.g., "stallion", "bear", "calm"). Default is "stallion".

scale_range
: Numeric vector of length 2 defining the minimum and maximum limits for the heatmap break range. Default is c(-1.25, 1.25).

annotation_legend
: Logical. Whether to show legends for the column and row annotations (e.g., Group and Feature Type color bars). Default is FALSE.

cellwidth
: Numeric. Width of individual heatmap cells in points. Set to make cells square. Default is 18.

cellheight
: Numeric. Height of individual heatmap cells in points. Set to make cells square. Default is 18.

main
: Character string for the main plot title. Default is "Group Gene Heatmap Plot | SlimR".

Value

A **pheatmap** object invisibly.

Colour palette

Discrete annotation colors are generated by internal paletteDiscrete() based on ArchR's palettes (*stallion* by default). Heatmap body values use a classic blue-white-red (RdBu reversed) palette matching custom breaks.

See Also

Other Section_5_Other_Functions_Provided: Celltype_Compare(), Celltype_annotation_Cellmarker2(), Celltype_annotation_Excel(), Celltype_annotation_PanglaoDB(), Celltype_annotation_Seurat(), Compute_Gene_AUC_ROC(), Compute_Gene_PRC(), Plot_Hierarchy_Proportion(), Plot_Voronoi_diagram(), paletteDiscrete()

Examples

```
## Not run:
# Basic expression heatmap without integrated row
Plot_Group_Gene_Heatmap(
  seurat_obj = sce,
  features = c("CD3D", "CD3E", "CD4", "CD8A"),
  group_by = "celltype",
  add_aucell_row = FALSE
)
```

```
# With top integrated gene signature score row
Plot_Group_Gene_Heatmap(
  seurat_obj = sce,
  features = c("TNF", "IL6", "NFKB1", "RELA"),
  group_by = "orig.ident",
  add_aucell_row = TRUE,
  aucell_row_name = "NFkB_Pathway_AUC"
)

## End(Not run)
```

`Plot_Hierarchy_Proportion` *Plot a Hierarchical Cell Type Tree with Optional Group Proportion Heatmap*

Description

Creates a publication-quality composite figure with a hierarchical tree panel and an optional proportion heatmap. Leaf labels are connected to the terminal nodes by short vertical sticks and placed at a fixed distance below the tree, guaranteeing a clean separation and perfect top alignment of all cell-type names. Short vertical ticks on top of the heatmap mark each column, aligning with the labels above. The tree height is independent of label length; the heatmap sits tightly beneath the labels.

Usage

```
Plot_Hierarchy_Proportion(
  seurat_obj,
  Main_cell_types,
  col_Main_cell_types = NULL,
  Cell_types = NULL,
  col_Cell_types = NULL,
  Sub_cell_types = NULL,
  col_Sub_cell_types = NULL,
  proportion = TRUE,
  Groups = NULL,
  adjust_by_group = FALSE,
  show_value_proportion = FALSE,
  show_labels = TRUE,
  low_col = "white",
  high_col = "navy"
)
```

Arguments

seurat_obj	A Seurat object.
Main_cell_types	Character string naming a column in seurat_obj@meta.data that holds the top-level cell-type labels.
col_Main_cell_types	Optional named or unnamed character vector of colours for the Main_cell_types categories. If NULL, colours are generated automatically.
Cell_types	Character string naming a column with second-level (sub-type) labels. Use NULL if no second level exists.
col_Cell_types	Optional colours for the Cell_types categories.
Sub_cell_types	Character string naming a column with third-level (sub-sub-type) labels. Use NULL if absent. Cell_types must be supplied when Sub_cell_types is used.
col_Sub_cell_types	Optional colours for the Sub_cell_types categories.
proportion	Logical scalar. If TRUE (the default), a proportion heatmap is drawn below the tree; if FALSE, only the tree with labels is returned.
Groups	Character string naming a meta.data column that defines sample groups for the proportion heatmap. Required when proportion = TRUE.
adjust_by_group	Logical scalar. If FALSE (default), each row of the heatmap shows the proportion (%) of each terminal cell type within the corresponding group. Set to TRUE when displaying sub-types derived from a larger, unevenly distributed population (e.g., immune subsets isolated from different conditions where total immune cell numbers differ).
show_value_proportion	Logical scalar. If TRUE, displays the calculated percentage proportion value as formatted text inside each cell of the heatmap. Default is FALSE.
show_labels	Logical scalar. If TRUE (the default), text labels are placed next to the non-leaf Main and Cell level nodes.
low_col	Character string for the lowest proportion colour in the heatmap. Default is "white".
high_col	Character string for the highest proportion colour in the heatmap. Default is "navy".

Value

Invisibly, a list with the following components:

tree_plot A **ggplot2** object with the hierarchical tree, leaf sticks, and labels.

prop_plot If proportion = TRUE, a **ggplot2** heatmap with top ticks; otherwise NULL.

combined_plot If proportion = TRUE and **patchwork** is installed, a combined plot with title *Hierarchical Proportion Plot | SlimR*; otherwise NULL.

The function automatically prints the combined_plot (or the tree_plot when no proportion is requested) to the active graphics device.

Colour palette

The default colour palette is derived from the **ArchR** package (ArchR project site, GitHub). If no custom palette is supplied, the function calls the internal `paletteDiscrete()` function, which replicates the *stallion* palette from ArchR. When the number of categories exceeds the palette size, colours are interpolated. ArchR is described in: Granja JM, Corces MR et al. ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nature Genetics*, 2021. doi:10.1038/s41588021007906

See Also

Other Section_5_Other_Functions_Provided: `Celltype_Compare()`, `Celltype_annotation_Cellmarker2()`, `Celltype_annotation_Excel()`, `Celltype_annotation_PanglaoDB()`, `Celltype_annotation_Seurat()`, `Compute_Gene_AUC_ROC()`, `Compute_Gene_PRC()`, `Plot_Group_Gene_Heatmap()`, `Plot_Voronoi_diagram()`, `paletteDiscrete()`

Examples

```
## Not run:
# Full three-level hierarchy with percentage proportion heatmap
res <- Plot_Hierarchy_Proportion(
  seurat_obj            = sce,
  Main_cell_types       = "Main_cell_type",
  Cell_types            = "Cell_type",
  Sub_cell_types        = "Sub_cell_type",
  proportion            = TRUE,
  Groups                = "orig.ident",
  show_value_proportion = TRUE
)

## End(Not run)
```

`Plot_Voronoi_diagram` *Generate a Weighted Voronoi Treemap for Hierarchical Cell Types*

Description

Creates a weighted Voronoi treemap displaying the hierarchical composition of single-cell data. The area of each polygon is proportional to the number of cells in the corresponding sub-type, and the polygons are grouped by the main cell type. Each sub-type is filled with a distinct colour from the **ArchR**-inspired *stallion* palette (or a user-supplied custom palette).

Note on colour handling: The function uses `WeightedTreemaps::voronoiTreemap()` to compute polygon coordinates, but **replaces the original drawing routine** with a custom `ggplot2`-based renderer. This is necessary because the upstream `add_color()` function treats discrete palettes as continuous gradients and does not respect the order of polygon cells, making it impossible to guarantee exact colour correspondence with functions like `Plot_Hierarchy_Proportion` or `Seurat::DimPlot`. The custom drawing code extracts polygon vertices and uses `scale_fill_manual()` with a named colour vector, ensuring that each cell type receives exactly its intended colour regardless of internal cell order.

Usage

```
Plot_Voronoi_diagram(
  seurat_obj,
  Main_cell_types,
  Cell_types,
  label_type = c("both", "count", "percentage", "none"),
  shape = c("rounded_rect", "circle"),
  seed = 1,
  col_Cell_types = NULL,
  label_size = 3.5,
  label_color = "black",
  label_fontface = "bold",
  border_color = "grey90",
  subtype_border_lwd = 0.5,
  main_border_lwd = 1.5,
  outer_border_lwd = 2,
  legend = TRUE,
  legend_position = "right"
)
```

Arguments

seurat_obj	A Seurat object.
Main_cell_types	Character string naming a column in `seurat_obj@meta.data` that holds the top-level (main) cell-type labels.
Cell_types	Character string naming a column in `seurat_obj@meta.data` that holds the sub-type labels.
label_type	Character string specifying the type of label to display inside each polygon. One of `"both"`, `"count"`, `"percentage"`, or `"none"`. Default is `"both"`.
shape	Character string specifying the shape of the polygons. `"rounded_rect"` (default) or `"circle"`.
seed	Integer seed for reproducibility of the Voronoi layout. Default is `1`.
col_Cell_types	Optional named or unnamed character vector of colours for the `Cell_types` categories. If provided, it overrides the default ArchR-derived palette.
label_size	Numeric value controlling the size of the polygon labels. Default is `3`.
label_color	Colour of the polygon labels. Default is `"black"`.
label_fontface	Font face for polygon labels (`"plain"`, `"italic"`, `"bold"`, etc.). Default is `"bold"`.
border_color	Colour of the border lines separating polygons. Default is `"grey90"` (a very light grey).
subtype_border_lwd	Line width for borders between sub-type polygons. Default is `0.15`.
main_border_lwd	Line width for borders between main cell types. Default is `0.35`.

`outer_border_lwd`
: Line width for the outer frame of the entire plot. The frame is drawn as a tight convex hull around the outermost polygons, perfectly following the natural outline of the treemap. Default is `0.4`.

`legend`
: Logical indicating whether to display a colour legend. Default is `TRUE`.

`legend_position`
: Position of the legend when `legend = TRUE`. Default is `"right"`.

Value

Invisibly, a list with `voronoi_treemap` (the raw treemap object) and `plot` (the **ggplot2** object).

Colour palette

The default colour palette is derived from the **ArchR** package (ArchR project site, GitHub). If no custom palette is supplied, the function calls the internal `paletteDiscrete()` function, which replicates the *stallion* palette from ArchR. When the number of categories exceeds the palette size, colours are interpolated. ArchR is described in: Granja JM, Corces MR et al. ArchR is a scalable software package for integrative single-cell chromatin accessibility analysis. *Nature Genetics*, 2021. doi:10.1038/s41588021007906

See Also

Other Section_5_Other_Functions_Provided: `Celltype_Compare()`, `Celltype_annotation_Cellmarker2()`, `Celltype_annotation_Excel()`, `Celltype_annotation_PanglaoDB()`, `Celltype_annotation_Seurat()`, `Compute_Gene_AUC_ROC()`, `Compute_Gene_PRC()`, `Plot_Group_Gene_Heatmap()`, `Plot_Hierarchy_Proportion()`, `paletteDiscrete()`

Examples

```
## Not run:
# Basic rounded-rectangle treemap with both count and percentage labels
Plot_Voronoi_diagram(
  seurat_obj      = sce,
  Main_cell_types = "compartment",
  Cell_types      = "cell_type"
)

# Circular polygons, show only percentages, custom colours
Plot_Voronoi_diagram(
  sce,
  "compartment",
  "cell_type",
  label_type     = "percentage",
  shape          = "circle",
  seed           = 42,
  col_Cell_types = c(
    "Enterocyte"   = "#1f78b4",
    "Goblet"       = "#33a02c",
    "Stem"         = "#e31a1c",
    "Paneth"       = "#ff7f00"
```

```
  ),
  legend_position = "right"
)

# Only sub-type names, no legend
Plot_Voronoi_diagram(
  sce,
  "compartment",
  "cell_type",
  label_type = "none",
  legend     = FALSE
)

## End(Not run)
```

`Read_excel_markers` *Create "Marker_list" from Excel files ".xlsx"*

Description

Create "Marker_list" from Excel files ".xlsx"

Usage

```
Read_excel_markers(path, has_colnames = TRUE)
```

Arguments

`path`	The path information of Marker files stored in ".xlsx" format. The Sheet name in the file is filled with cell type. The first line of each Sheet is the table head, the first column is filled with markers information, and the following column is filled with mertic information.
`has_colnames`	Logical value indicating whether the first row contains column names. If FALSE, the first column will be named "Markers" and subsequent columns will be named "Col1", "Col2", etc.

Value

The standardized "Marker_list" in the SlimR package.

See Also

Other Section_2_Standardized_Markers_List: `Markers_filter_Cellmarker2()`, `Markers_filter_PanglaoDB()`, `Markers_filter_ScType()`, `Read_seurat_markers()`

Examples

```
## Not run:
Markers_list_Excel <- Read_excel_markers(
    "D:/Laboratory/Marker_load.xlsx"
    )

## End(Not run)
```

`Read_seurat_markers` *Create "Marker_list" from Seurat object*

Description

Create "Marker_list" from Seurat object

Usage

```
Read_seurat_markers(
  df,
  sources = c("Seurat", "presto"),
  sort_by = "FSS",
  gene_filter = 20
)
```

Arguments

`df`	Dataframe generated by "FindAllMarkers" function, recommend to use parameter "group.by = "Cell_type"" and "only.pos = TRUE".
`sources`	Type of markers sources to use. Be one of: `"Seurat"` or `"presto"`.
`sort_by`	Marker sorting parameter, for Seurat sources, select "avg_log2FC" or "p_val_adj" or "FSS" (Feature Significance Score, FSS, product value of `log2FC` and `Expression ratio`). Default parameters use "sort_by = 'FSS'".for presto sources, select "logFC" or "padj" or "FSS". Default parameters use "sort_by = 'FSS'".
`gene_filter`	The number of markers left for each cell type based on the "sort_by" parameter's level of difference. Default parameters use "gene_fliter = 20"

Value

The standardized "Marker_list" in the SlimR package.

See Also

Other Section_2_Standardized_Markers_List: `Markers_filter_Cellmarker2()`, `Markers_filter_PanglaoDB()`, `Markers_filter_ScType()`, `Read_excel_markers()`

Examples

```
## Not run:
# Example for Seurat sources markers
seurat_markers <- Seurat::FindAllMarkers(
    object = sce,
    group.by = "Cell_type",
    only.pos = TRUE)

Markers_list_Seurat <- Read_seurat_markers(seurat_markers,
    sources = "Seurat",
    sort_by = "avg_log2FC",
    gene_filter = 20
    )

# Example for presto sources markers
seurat_markers <- dplyr::filter(
    presto::wilcoxauc(
      X = sce,
      group_by = "Cell_type",
      seurat_assay = "RNA"
      ),
    padj < 0.05, logFC > 0.5
    )

Markers_list_Seurat <- Read_seurat_markers(seurat_markers,
    sources = "presto",
    sort_by = "logFC",
    gene_filter = 20
    )

## End(Not run)
```

`scBaseCount` *scBaseCount dataset*

Description

A processed cross-species marker gene resource derived from scBaseCount. The dataset contains four species-specific gene set lists: human, mouse, Drosophila, and zebrafish. Each species-specific element is a named list of marker gene sets, where names correspond to cell types and values are character vectors of marker genes.

Usage

```
scBaseCount
```

Format

A named list with 4 elements:

human_gene_sets A named list of human marker gene sets. Names are cell types; each element is a character vector of marker genes.

mouse_gene_sets A named list of mouse marker gene sets. Names are cell types; each element is a character vector of marker genes.

fly_gene_sets A named list of Drosophila marker gene sets. Names are cell types; each element is a character vector of marker genes.

zebrafish_gene_sets A named list of zebrafish marker gene sets. Names are cell types; each element is a character vector of marker genes.

Details

The scBaseCount resource was built from more than 500 million cells across four species. The species-specific gene set lists can be extracted and used directly as standardized marker lists, for example: `Markers_list_scBaseCount_human <- scBaseCount$human_gene_sets`, `Markers_list_scBaseCount_mouse <- scBaseCount$mouse_gene_sets`, `Markers_list_scBaseCount_fly <- scBaseCount$fly_gene_sets`, and `Markers_list_scBaseCount_zebrafish <- scBaseCount$zebrafish_gene_sets`. These lists are compatible with SlimR annotation functions.

When using this database, please cite Youngblut et al. (2026) doi:10.1016/j.cell.2026.08.025.

Source

doi:10.1016/j.cell.2026.08.025

See Also

Other Section_0_Database: `CellTypist`, `Cellmarker2`, `Cellmarker2_raw`, `Cellmarker2_table`, `Markers_list_PCTAM`, `Markers_list_PCTIT`, `Markers_list_TCellSI`, `Markers_list_scIBD`, `PanglaoDB`, `PanglaoDB_raw`, `PanglaoDB_table`, `ScType`, `ScType_raw`, `ScType_table`

`ScType` *ScType dataset*

Description

A processed long-format dataset containing marker genes for different cell types from the ScType database. Each row represents one marker gene for a given tissue type and cell type.

Usage

```
ScType
```

Format

A tibble with 3 columns:

tissue_type Tissue type (e.g., "Immune system", "Brain", "Liver")

cell_name Cell type name, formatted as "cellName(shortName)" when a short name is available, or "cellName" otherwise

marker Gene symbol of the marker

Details

This dataset is used to filter and create a standardized marker list. The dataset can be filtered based on tissue type and cell name to generate a list of marker genes for specific cell types using `Markers_filter_ScType`.

Source

https://github.com/IanevskiAleksandr/sc-type

See Also

Other Section_0_Database: `CellTypist`, `Cellmarker2`, `Cellmarker2_raw`, `Cellmarker2_table`, `Markers_list_PCTAM`, `Markers_list_PCTIT`, `Markers_list_TCellSI`, `Markers_list_scIBD`, `PanglaoDB`, `PanglaoDB_raw`, `PanglaoDB_table`, `ScType_raw`, `ScType_table`, `scBaseCount`

ScType_raw *ScType raw dataset*

Description

The original ScType marker database before processing.

Usage

```
ScType_raw
```

Format

A tibble with 5 columns:

tissueType Tissue type

cellName Full cell type name

geneSymbolmore1 Comma-separated positive marker genes

geneSymbolmore2 Comma-separated negative marker genes (not used in processing)

shortName Abbreviated cell type name

Source

https://github.com/IanevskiAleksandr/sc-type

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_table, scBaseCount

---

ScType_table *ScType metadata table*

Description

A list of frequency tables summarizing the ScType database, useful for exploring available tissue types and cell types before filtering.

Usage

```
ScType_table
```

Format

A list with 2 elements:

tissue_type Frequency table of tissue types

cell_name Frequency table of cell type names

Source

https://github.com/IanevskiAleksandr/sc-type

See Also

Other Section_0_Database: CellTypist, Cellmarker2, Cellmarker2_raw, Cellmarker2_table, Markers_list_PCTAM, Markers_list_PCTIT, Markers_list_TCellSI, Markers_list_scIBD, PanglaoDB, PanglaoDB_raw, PanglaoDB_table, ScType, ScType_raw, scBaseCount

Index

∗ **Section_0_Database**
Cellmarker2, 7
Cellmarker2_raw, 7
Cellmarker2_table, 8
CellTypist, 33
Markers_list_PCTAM, 44
Markers_list_PCTIT, 45
Markers_list_scIBD, 46
Markers_list_TCellSI, 46
PanglaoDB, 47
PanglaoDB_raw, 48
PanglaoDB_table, 48
scBaseCount, 65
ScType, 66
ScType_raw, 67
ScType_table, 68
∗ **Section_1_Functions_Use_in_Package**
calculate_cluster_variability, 4
calculate_expression, 4
calculate_expression_skewness, 5
calculate_probability, 6
compute_adaptive_parameters, 34
estimate_batch_effect, 40
extract_dataset_features, 40
∗ **Section_2_Standardized_Markers_List**
Markers_filter_Cellmarker2, 41
Markers_filter_PanglaoDB, 42
Markers_filter_ScType, 43
Read_excel_markers, 63
Read_seurat_markers, 64
∗ **Section_3_Automated_Annotation**
Celltype_Annotation, 9
Celltype_Annotation_PerCell, 20
Celltype_Calculate, 23
Celltype_Calculate_PerCell, 26
Celltype_Verification, 31
Celltype_Verification_PerCell, 32
Parameter_Calculate, 49
percell_workflow, 50
∗ **Section_4_Semi_Automated_Annotation**
Celltype_Annotation_Combined, 12
Celltype_Annotation_Features, 15
Celltype_Annotation_Heatmap, 17
∗ **Section_5_Other_Functions_Provided**
Celltype_annotation_Cellmarker2, 10
Celltype_annotation_Excel, 13
Celltype_annotation_PanglaoDB, 19
Celltype_annotation_Seurat, 22
Celltype_Compare, 29
Compute_Gene_AUC_ROC, 35
Compute_Gene_PRC, 37
Plot_Group_Gene_Heatmap, 56
Plot_Hierarchy_Proportion, 58
Plot_Voronoi_diagram, 60
∗ **datasets**
Cellmarker2, 7
Cellmarker2_raw, 7
Cellmarker2_table, 8
CellTypist, 33
Markers_list_PCTAM, 44
Markers_list_PCTIT, 45
Markers_list_scIBD, 46
Markers_list_TCellSI, 46
PanglaoDB, 47
PanglaoDB_raw, 48
PanglaoDB_table, 48
scBaseCount, 65
ScType, 66
ScType_raw, 67
ScType_table, 68
∗ **documentation**
percell_workflow, 50

ArchRPalettes, 3

calculate_cluster_variability, 4
calculate_cluster_variability(), *5, 6*, *35*, *40, 41*

calculate_expression, 4
calculate_expression(), *4–6*, *35*, *40, 41*
calculate_expression_skewness, 5
calculate_expression_skewness(), *4–6*, *35*, *40, 41*
calculate_probability, 6
calculate_probability(), *4, 5*, *35*, *40, 41*
Cellmarker2, 7, *8*, *34*, *45–49*, *66–68*
Cellmarker2_raw, 7, *7, 8*, *34*, *45–49*, *66–68*
Cellmarker2_table, *7*, 8, *8*, *34*, *45–49*, *66–68*
Celltype_Annotation, 9
Celltype_Annotation(), *21*, *25*, *28*, *32, 33*, *50*, *54*
Celltype_annotation_Cellmarker2, 10
Celltype_annotation_Cellmarker2(), *14*, *20*, *23*, *30*, *36*, *39*, *57*, *60*, *62*
Celltype_Annotation_Combined, 12
Celltype_Annotation_Combined(), *16*, *18*
Celltype_annotation_Excel, 13
Celltype_annotation_Excel(), *11*, *20*, *23*, *30*, *36*, *39*, *57*, *60*, *62*
Celltype_Annotation_Features, 15
Celltype_Annotation_Features(), *13*, *18*
Celltype_Annotation_Heatmap, 17
Celltype_Annotation_Heatmap(), *13*, *16*
Celltype_annotation_PanglaoDB, 19
Celltype_annotation_PanglaoDB(), *11*, *14*, *23*, *30*, *36*, *39*, *57*, *60*, *62*
Celltype_Annotation_PerCell, 20
Celltype_Annotation_PerCell(), *9*, *25*, *28*, *32, 33*, *50*, *54*
Celltype_annotation_Seurat, 22
Celltype_annotation_Seurat(), *11*, *14*, *20*, *30*, *36*, *39*, *57*, *60*, *62*
Celltype_Calculate, 23
Celltype_Calculate(), *9*, *21*, *28*, *32, 33*, *50*, *54*
Celltype_Calculate_PerCell, 26
Celltype_Calculate_PerCell(), *9*, *21*, *25*, *32, 33*, *50*, *54*
Celltype_Compare, 29
Celltype_Compare(), *11*, *14*, *20*, *23*, *36*, *39*, *57*, *60*, *62*
Celltype_Verification, 31
Celltype_Verification(), *9*, *21*, *25*, *28*, *33*, *50*, *54*
Celltype_Verification_PerCell, 32
Celltype_Verification_PerCell(), *9*, *21*, *25*, *28*, *32*, *50*, *54*
CellTypist, *7, 8*, 33, *45–49*, *66–68*
compute_adaptive_parameters, 34
compute_adaptive_parameters(), *4–6*, *40, 41*
Compute_Gene_AUC_ROC, 35
Compute_Gene_AUC_ROC(), *11*, *14*, *20*, *23*, *30*, *39*, *57*, *60*, *62*
Compute_Gene_PRC, 37
Compute_Gene_PRC(), *11*, *14*, *20*, *23*, *30*, *36*, *57*, *60*, *62*

estimate_batch_effect, 40
estimate_batch_effect(), *4–6*, *35*, *41*
extract_dataset_features, 40
extract_dataset_features(), *4–6*, *35*, *40*

Markers_filter_Cellmarker2, 41
Markers_filter_Cellmarker2(), *43, 44*, *63, 64*
Markers_filter_PanglaoDB, 42
Markers_filter_PanglaoDB(), *42*, *44*, *63, 64*
Markers_filter_ScType, 43, *67*
Markers_filter_ScType(), *42, 43*, *63, 64*
Markers_list_PCTAM, *7, 8*, *34*, 44, *45–49*, *66–68*
Markers_list_PCTIT, *7, 8*, *34*, 45, *45–49*, *66–68*
Markers_list_scIBD, *7, 8*, *34*, *45*, 46, *47–49*, *66–68*
Markers_list_TCellSI, *7, 8*, *34*, *45*, 46, *46–49*, *66–68*

paletteDiscrete(), *11*, *14*, *20*, *23*, *30*, *36*, *39*, *57*, *60*, *62*
PanglaoDB, *7, 8*, *34*, *45, 46*, 47, *47–49*, *66–68*
PanglaoDB_raw, *7, 8*, *34*, *45–47*, 48, *49*, *66–68*
PanglaoDB_table, *7, 8*, *34*, *45–47*, 48, *48*, *66–68*
Parameter_Calculate, 49
Parameter_Calculate(), *9*, *21*, *25*, *28*, *32, 33*, *54*
percell_workflow, *9*, *21*, *25*, *28*, *32, 33*, 50, *50*
plot.pheatmap, 55
Plot_Group_Gene_Heatmap, 56
Plot_Group_Gene_Heatmap(), *11*, *14*, *20*, *23*, *30*, *36*, *39*, *60*, *62*

Plot_Hierarchy_Proportion, 58
Plot_Hierarchy_Proportion(), *11*, *14*, *20*, *23*, *30*, *36*, *39*, *57*, *62*
Plot_Voronoi_diagram, 60
Plot_Voronoi_diagram(), *11*, *14*, *20*, *23*, *30*, *36*, *39*, *57*, *60*

Read_excel_markers, 63
Read_excel_markers(), *42–44*, *64*
Read_seurat_markers, 64
Read_seurat_markers(), *42–44*, *63*

scBaseCount, *7, 8*, *34*, *45–49*, 65, *67, 68*
ScType, *7, 8*, *34*, *45–49*, 66, *66*, *68*
ScType_raw, *7, 8*, *34*, *45–49*, *66*, 67, *67, 68*
ScType_table, *7, 8*, *34*, *45–49*, *66, 67*, 68, *68*