Package: ProjecTILs (via r-universe)

May 29, 2026

Type Package

Title Reference-based analysis of scRNA-seq data

Version 3.7.0

Description This package implements methods to project single-cell RNA-seq data onto a reference atlas, enabling interpretation of unknown cell transcriptomic states in the the context of known, reference states.

Depends R(>= 4.3.0)

Imports Seurat(>= 5.0.0), SeuratObject(>= 5.0.0), uwot, umap, Matrix, BiocParallel, BiocNeighbors, patchwork, reshape2, ggplot2, grDevices, scales, pracma, STACAS, UCell, scGate, pheatmap, RColorBrewer, dplyr, tidyr, jsonlite, digest

Suggests fastICA, EnhancedVolcano, plotly,

BugReports https://github.com/carmonalab/ProjecTILs/issues

URL https://github.com/carmonalab/ProjecTILs

License GPL-3 + file LICENSE

Encoding UTF-8

LazyData true

LazyDataCompression xz

RoxygenNote 7.3.3

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

Repository https://carmonalab.r-universe.dev

Date/Publication 2025-12-19 08:30:18 UTC

RemoteUrl https://github.com/carmonalab/ProjecTILs

RemoteRef HEAD

RemoteSha 1159e1778820180fd0233bbfc0d7f296e35bd25f

Contents

`cell.cycle.obj` *Cell cycling signatures*

Description

A list of cell cycling signatures (G1.S and G2.M phases), for mouse and human.

Usage

```
cell.cycle.obj
```

Format

A list of cycling signatures.

Source

doi:10.1126/science.aad0501

`cellstate.predict` *Predict cell states of a projected dataset*

Description

This function uses a nearest-neighbor algorithm to predict a feature (e.g. the cell state) of the query cells. Distances between cells in the reference map and cells in the query are calculated in a reduced space (PCA or UMAP) and the feature is assigned to query cells based on a consensus of its nearest neighbors in the reference object.

Usage

```
cellstate.predict(
  ref,
  query,
  reduction = "pca",
  ndim = NULL,
  k = 5,
  min.confidence = 0.2,
  nn.decay = 0.1,
  labels.col = "functional.cluster"
)
```

Arguments

`ref`	Reference Atlas
`query`	Seurat object with query data
`reduction`	The dimensionality reduction used to calculate pairwise distances. One of "pca" or "umap"
`ndim`	How many dimensions in the reduced space to be used for distance calculations
`k`	Number of neighbors to assign the cell type
`min.confidence`	Minimum confidence score to return cell type labels (otherwise NA)
`nn.decay`	Weight decay for internal nearest neighbors (between 0 and 1)
`labels.col`	The metadata field of the reference to annotate the clusters (default: functional.cluster)

Value

The query object submitted as parameter, with two additional metadata slots for predicted state and its confidence score

Examples

```
data(query_example_seurat)
ref <- load.reference.map()
q <- make.projection(query_example_seurat, ref=ref)
```

```
q <- cellstate.predict(ref, query=q)
table(q$functional.cluster)
```

`celltype.heatmap` *Plot a averaged expression heatmap from a Seurat object*

Description

This function allows to calculate and plot pseudo-bulk gene expression by cell type and custom grouping variables. Data can be split in principle by any metadata present in the starting Seurat object (e.g. patient, tissue, study, etc.). This can be useful to evaluate consistency of expression profiles for different cell types across samples, studies or other grouping variables.

Usage

```
celltype.heatmap(
  data,
  assay = "RNA",
  slot = "data",
  genes,
  ref = NULL,
  scale = "row",
  method = c("ward.D2", "ward.D", "average"),
  brewer.palette = "RdBu",
  palette_reverse = F,
  palette = NULL,
  cluster.col = "functional.cluster",
  group.by = NULL,
  flip = FALSE,
  cluster_genes = FALSE,
  cluster_samples = FALSE,
  min.cells = 10,
  show_samplenames = FALSE,
  remove.NA.meta = TRUE,
  breaks = seq(-2, 2, by = 0.1),
  return.matrix = FALSE,
  ...
)
```

Arguments

`data`	A Seurat object to be used for the heatmap
`assay`	A string indicating the assay type, default is "RNA"
`slot`	Data slot (layer) in Seurat object
`genes`	A vector of genes to be used in the heatmap

ref	A ProjecTILs reference Seurat object to define the order of functional.cluster
scale	A string indicating the scale of the heatmap, default is "row"
method	A string or vector of strings indicating the clustering method to be used, default is "ward.D2"
brewer.palette	A string indicating the color palette to be used, default is "RdBu"
palette_reverse	A boolean indicating if color palette should be reversed, default is FALSE
palette	A named list containing colors vectors compatible with pheatmap. The list is named by the metadata names, default is taking these palettes to plot metadata: "Paired","Set2","Accent","Dark2","Set1","Set3".
cluster.col	The metadata column name containing the cell type labels
group.by	The metadata column names used as grouping variables
flip	A boolean indicating if the heatmap should be flipped, default is FALSE
cluster_genes	A boolean indicating if genes should be clustered, default is FALSE
cluster_samples	A boolean indicating if samples should be clustered, default is FALSE
min.cells	A value defining the minimum number of cells a sample should have to be kept, default is 10
show_samplenames	A boolean indicating whether the heatmap should display the sample names or not, default is FALSE
remove.NA.meta	A boolean indicating if missing samples with missing metadata should be plotted, default is TRUE
breaks	Range of values for plotting (see 'breaks' parameter in pheatmap)
return.matrix	If true, return the pseudo-bulk data matrix instead of graphical output
...	Additional parameters for 'pheatmap'

Value

A pheatmap plot, displaying averaged expression values across genes for each selected genes and samples.

Examples

```
library(Seurat)
ref <- load.reference.map(ref = "https://figshare.com/ndownloader/files/38921366")
celltype.heatmap(ref, assay = "RNA", genes = c("LEF1","SELL","GZMK","FGFBP2"),
   ref = ref, cluster.col = "functional.cluster", group.by = c("orig.ident", "Tissue"))
```

compute_silhouette *Calculate Silhouette coefficient*

Description

Given a projected object and its reference, calculate silhouette coefficient for query cells with respect to reference cells with the same cell labels.

Usage

```
compute_silhouette(
  ref,
  query = NULL,
  reduction = "pca",
  ndim = NULL,
  label_col = "functional.cluster",
  normalize.scores = FALSE,
  min.cells = 20
)
```

Arguments

ref	Reference object
query	Query object. If not specified, the silhouette coefficient of only the reference will be calculated
reduction	Which dimensionality reduction to use for euclidian distance calculation
ndim	Number of dimensions in the dimred to use for distance calculation. If NULL, use all dimensions.
label_col	Metadata column with cell type annotations. Must be present both in reference and query
normalize.scores	Whether to normalize silhouette scores by the average cell type silhouettes of the reference
min.cells	Only report silhouette scores for cell type with at least this number of cells

Value

A dataframe with average silhouette coefficient for each cell type

Examples

```
data(query_example_seurat)
ref <- load.reference.map()
q <- Run.ProjecTILs(query_example_seurat, ref=ref, fast.umap.predict=TRUE)
combined <- compute_silhouette(ref, query=q)
```

`find.discriminant.dimensions` *Find discriminant dimensions*

Description

Searches PCA or ICA dimensions where the query set deviates the most from a control set or from the reference map. It can be useful to suggest novel cell states that escape from the main axes of diversity of the UMAP

Usage

```
find.discriminant.dimensions(
  ref,
  query,
  query.control = NULL,
  query.assay = "RNA",
  state = "largest",
  labels.col = "functional.cluster",
  reduction = "ICA",
  test = c("ks", "t.test"),
  ndim = 50,
  print.n = 3,
  verbose = T
)
```

Arguments

`ref`	Seurat object with reference atlas
`query`	Seurat object with query data
`query.control`	Optionally, you can compare your query with a control sample, instead of the reference
`query.assay`	The data slot to be used for enrichment analysis
`state`	Perform discriminant analysis on this cell state. Can be either: • "largest" - Performs analysis on the cell state most represented in the query set(s) • "all" - Performs analysis on the complete dataset, using all cells • A specific cell state, one of the states in metadata field labels.col
`labels.col`	The metadata field used to annotate the clusters (default: functional.cluster)
`reduction`	Which dimensionality reduction to use (either ICA or PCA)
`test`	Which test to perform between the dataset distributions in each ICA/PCA dimension. One of ‘ks‘ (Kolmogorov-Smirnov) or ‘t.test‘ (T-test)
`ndim`	How many dimensions to consider in the reduced ICA/PCA space
`print.n`	The number of top dimensions to return to STDOUT
`verbose`	Print results to STDOUT

Value

A dataframe, where rows are ICA/PCA dimensions. ICA/PCAs are ranked by statistical significance when comparing their distribution between query and control (or query vs. reference map)

Examples

```
find.discriminant.dimensions(ref, query=query.set)
find.discriminant.dimensions(ref, query=query.set, query.control=control.set)
```

`find.discriminant.genes`

Find discriminant genes

Description

Based on 'FindMarkers'. It performs differential expression analysis between a projected query and a control (either the reference map or a control sample), for a given cell type. Useful to detect whether specific cell states over/under-express genes between conditions or with respect to the reference.

Usage

```
find.discriminant.genes(
  ref,
  query,
  query.control = NULL,
  ref.assay = "RNA",
  query.assay = "RNA",
  state = "largest",
  labels.col = "functional.cluster",
  test = "wilcox",
  min.cells = 10,
  genes.use = c("variable", "all"),
  ...
)
```

Arguments

`ref`	Seurat object with reference atlas
`query`	Seurat object with query data
`query.control`	Optionally, you can compare your query with a control sample, instead of the reference
`ref.assay`	The referece assay to be used for DE analysis
`query.assay`	The query assay to be used for DEG analyis, if comparing to the reference

state	Perform discriminant analysis on this cell state. Can be either: • "largest" - Performs analysis on the cell state most represented in the query set(s) • "all" - Performs analysis on the complete dataset, using all cells • A specific cell state, one of the states in metadata field labels.col
labels.col	The metadata field used to annotate the clusters (default: functional.cluster)
test	Type of test for DE analysis. See help for 'FindMarkers' for implemented tests.
min.cells	Minimum number of cells in the cell type to proceed with analysis.
genes.use	What subset of genes to consider for DE analysis: • "variable" - Only consider variable genes of the reference • "all" - Use intersection of all genes in query and control • A custom list of genes
...	Adding parameters for 'FindMarkers'

Value

A dataframe with a ranked list of genes as rows, and statistics as columns (e.g. log fold-change, p-values). See help for 'FindMarkers' for more details.

Examples

```
# Discriminant genes between query and reference in cell type "Tex"
markers <- find.discriminant.genes(ref, query=query.set, state="Tex")

# Discriminant genes between query and control sample in most represented cell type
markers <- find.discriminant.genes(ref, query=query.set, query.control=control.set)

# Pass results to EnhancedVolcano for visual results
library(EnhancedVolcano)
EnhancedVolcano(markers, lab = rownames(markers), x = 'avg_logFC', y = 'p_val')
```

FindAllMarkers.bygroup

Gene expression markers shared by multiple groups of cells

Description

This function expands FindAllMarkers to find markers that are differentially expressed across multiple datasets or samples. Given a Seurat object with identity classes (for example annotated clusters) and a grouping variable (for example a Sample ID), it calculate differentially expressed genes (DEGs) individually for each sample. Then it determines the fraction of samples for which the gene was found to be differentially expressed.

Usage

```
FindAllMarkers.bygroup(
  object,
  split.by = NULL,
  only.pos = TRUE,
  features = NULL,
  min.cells.group = 10,
  min.freq = 0.5,
  ...
)
```

Arguments

object	A Seurat object
split.by	A metadata column name - the data will be split by this column to calculate FindAllMarkers separately for each data split
only.pos	Only return positive markers (TRUE by default)
features	Genes to test. Default is to use all genes
min.cells.group	Minimum number of cells in the group - if lower the group is skipped
min.freq	Only return markers which are differentially expressed in at least this fraction of datasets.
...	Additional paramters to FindAllMarkers

Details

This function can be useful to find marker genes that are specific for individual cell types, and that are found to be so consistently across multiple samples.

Value

A list of marker genes for each identity class (typically clusters), with two associated numerical values: i) the fraction of datasets for which the marker was found to be differentially expressed; ii) the average log-fold change for the genes across datasets

Examples

```
library(Seurat)
ref <- load.reference.map(ref = "https://figshare.com/ndownloader/files/38921366")
Idents(ref) <- "functional.cluster"
FindAllMarkers.bygroup(ref, split.by = "Sample", min.cells.group=30, min.freq=0.8)
```

`get.reference.maps` *Retrieve and load reference atlas*

Description

Download and load reference atlases.

Usage

```
get.reference.maps(
  collection = NULL,
  reference = NULL,
  update = FALSE,
  directory = "./ProjecTILs_references",
  as.list = TRUE,
  verbose = TRUE
)
```

Arguments

`collection`	Collection to download and load. See available collection using list.reference.maps. If NULL, all are downloaded and loaded (default)
`reference`	References to download and load. See available collection using list.reference.maps. If NULL, all are downloaded and loaded (default)
`update`	Boolean whether to delete current reference maps and download them again
`directory`	Directory where to download and load from reference maps. By default a directory named "ProjecTILs_references" is created in working directory.
`as.list`	Boolean whether to simplify list (`FALSE`) or, by default, keep a list of lists for each collection (`TRUE`).
`verbose`	Inform of the status of processes

Examples

```
# explore available reference maps
list.reference.maps()

# consider increasing downloading timeout
options(timeout = 1000)

# get all available reference maps
ref.maps <- get.reference.maps()

# get certain collections or reference maps
# all human references maps
ref.maps.human <- get.reference.maps(collection = "human")

# only some references
```

```
ref.maps <- get.reference.maps(reference = "DC")
ref.maps.CD4 <- get.reference.maps(reference = c("CD4", "Virus_CD4T"))

# update previously downloaded maps
ref.maps <- get.reference.maps(update = TRUE)
```

Hs2Mm.convert.table *Human-mouse ortholog conversion table*

Description

A conversion table of stable orthologs between Hs and Mm.

Usage

```
Hs2Mm.convert.table
```

Format

A dataframe containing gene ortholog mapping.

Source

https://www.ensembl.org/Mus_musculus/Info/Index

---

list.reference.maps *Available reference atlas for ProjecTILs*

Description

Obtain the list of available reference atlas for ProjecTILs to then download and load them using get.reference.maps.

Usage

```
list.reference.maps()
```

Examples

```
# explore available reference maps
list.reference.maps()
```

load.reference.map *Load Reference Atlas*

Description

Load or download the reference map for dataset projection. By the default it downloads a reference atlas of tumour-infiltrating lymphocytes (TILs) from mouse.

Usage

```
load.reference.map(ref = "referenceTIL")
```

Arguments

ref	Reference atlas as a Seurat object (by default downloads a mouse reference TIL atlas). To use a custom reference atlas, provide a .rds object or a URL to a .rds object, storing a Seurat object prepared using make.reference

Examples

```
# consider increasing downloading timeout, if downloading Default reference atlas or large reference
options(timeout = 1000)

# Download and load default reference map
ref <- load.reference.map()

# download reference map from url
ref.web <- load.reference.map(ref = url)

# Load any reference map
ref <- load.reference.map(ref = "path/to/ref")
```

make.projection *Project a query scRNA-seq dataset onto a reference atlas*

Description

This function allows projecting ("query") single-cell RNA-seq datasets onto a reference map (i.e. a curated and annotated scRNA-seq dataset). To project multiple datasets, submit a list of Seurat objects with the query parameter. The projection consists of 3 steps:

- pre-processing: optional steps which might include pre-filtering of cells by markers using 'scGate', data normalization, and ortholog conversion.
- batch-effect correction: uses built-in STACAS algorithm to detect and correct for batch effects (this step assumes that at least a fraction of the cells in the query are in the same state than cells in the reference)
- embedding of corrected query data in the reduced-dimensionality spaces (PCA and UMAP) of the reference map.

Usage

```
make.projection(
  query,
  ref = NULL,
  filter.cells = TRUE,
  query.assay = NULL,
  direct.projection = FALSE,
  STACAS.anchor.coverage = 0.7,
  STACAS.correction.scale = 100,
  STACAS.k.anchor = 5,
  STACAS.k.weight = "max",
  skip.normalize = FALSE,
  fast.umap.predict = FALSE,
  ortholog_table = NULL,
  scGate_model = NULL,
  ncores = 1,
  progressbar = TRUE
)
```

Arguments

query	Query data, either as single Seurat object or as a list of Seurat object
ref	Reference Atlas - if NULL, downloads the default TIL reference atlas
filter.cells	Pre-filter cells using 'scGate'. Only set to FALSE if the dataset has been previously subset to cell types represented in the reference.
query.assay	Which assay slot to use for the query (defaults to DefaultAssay(query))
direct.projection	If true, apply PCA transformation directly without alignment
STACAS.anchor.coverage	Focus on few robust anchors (low STACAS.anchor.coverage) or on a large amount of anchors (high STACAS.anchor.coverage). Must be number between 0 and 1.
STACAS.correction.scale	Slope of sigmoid function used to determine strength of batch effect correction.
STACAS.k.anchor	Integer. For alignment, how many neighbors (k) to use when picking anchors.
STACAS.k.weight	Number of neighbors to consider when weighting anchors. Default is "max", which disables local anchor weighting.
skip.normalize	By default, log-normalize the count data. If you have already normalized your data, you can skip normalization.
fast.umap.predict	Fast approximation for UMAP projection. Uses coordinates of nearest neighbors in PCA space to assign UMAP coordinates (credits to Changsheng Li for the implementation)
ortholog_table	Dataframe for conversion between ortholog genes (by default package object Hs2Mm.convert.table)

scGate_model	scGate model used to filter target cell type from query data (if NULL use the model stored in ref@misc$scGate)
ncores	Number of cores for parallel execution (requires BiocParallel)
progressbar	Whether to show a progress bar for projection process or not (requires BiocParallel)

Details

See load.reference.map to load or download a reference atlas. See also ProjecTILs.classifier to use ProjecTILs as a cell type classifier.

Value

An augmented Seurat object with projected UMAP coordinates on the reference map

Examples

```
data(query_example_seurat)
ref <- load.reference.map()
make.projection(query_example_seurat, ref=ref)
```

---

make.reference *Make a ProjecTILs reference*

Description

Converts a Seurat object to a ProjecTILs reference atlas. You can preserve your low-dimensionality embeddings (e.g. UMAP) in the reference atlas by setting 'recalculate.umap=FALSE', or recalculate the UMAP using one of the two methods umap::umap or uwot::umap. Recalculation allows exploting the 'predict' functionalities of these methods for embedding of new points; skipping recalculation will make the projection use an approximation for UMAP embedding of the query.

Usage

```
make.reference(
  ref,
  assay = NULL,
  assay.raw = "RNA",
  atlas.name = "custom_reference",
  annotation.column = "functional.cluster",
  recalculate.umap = FALSE,
  umap.method = c("umap", "uwot"),
  metric = "cosine",
  min_dist = 0.3,
  n_neighbors = 30,
  ndim = 20,
```

```
  dimred = "umap",
  nfeatures = 1000,
  color.palette = NULL,
  scGate.model.human = NULL,
  scGate.model.mouse = NULL,
  store.markers = FALSE,
  n.markers = 10,
  seed = 123,
  layer1_link = NULL
)
```

Arguments

ref	Seurat object with reference atlas
assay	The assay storing the reference expression data (e.g. "integrated")
assay.raw	The assay storing raw expression data (e.g. "RNA")
atlas.name	An optional name for your reference
annotation.column	The metadata column with the cluster annotations for this atlas
recalculate.umap	If TRUE, run the 'umap' or 'uwot' algorithm to generate embeddings. Otherwise use the embeddings stored in the 'dimred' slot.
umap.method	Which method to use for calculating the umap reduction
metric	Distance metric to use to find nearest neighbors for UMAP
min_dist	Effective minimum distance between UMAP embedded points
n_neighbors	Size of local neighborhood for UMAP
ndim	Number of PCA dimensions
dimred	Use the pre-calculated embeddings stored at 'Embeddings(ref, dimred)'
nfeatures	Number of variable features (only calculated if not already present)
color.palette	A (named) vector of colors for the reference plotting functions. One color for each cell type in 'functional.cluster'
scGate.model.human	A human scGate model to purify the cell types represented in the map. For example, if the map contains CD4 T cell subtype, specify an scGate model for CD4 T cells.
scGate.model.mouse	A mouse scGate model to purify the cell types represented in the map.
store.markers	Whether to store the top differentially expressed genes in 'ref@misc$gene.panel'
n.markers	Store the top 'n.markers' for each subtype given by differential expression analysis
seed	Random seed
layer1_link	Broad cell type contained in this reference atlas (i.e. CD4T, CL:0000624...) to link with broad cell type annotation (layer1).

Value

A reference atlas compatible with ProjecTILs

Examples

```
custom_reference <- ProjecTILs::make.reference(my_dataset, recalculate.umap=T)
```

`merge.Seurat.embeddings` *Merge Seurat objects, including reductions (e.g. PCA, UMAP, ICA)*

Description

Given two Seurat objects, merge counts and data as well as dim reductions (PCA, UMAP, ICA, etc.)

Usage

```
## S3 method for class 'Seurat.embeddings'
merge(x = NULL, y = NULL, merge.dr = TRUE, ...)
```

Arguments

`x`	First object to merge
`y`	Second object to merge
`merge.dr`	How to handle merging dimensional reductions (see merge.Seurat)
`...`	More parameters to merge function

Value

A merged Seurat object

Examples

```
o1 <- query_example_seurat
o2 <- query_example_seurat
seurat.merged <- merge.Seurat.embeddings(o1, o2)
#To merge multiple object stored in a list
seurat.merged <- Reduce(f=merge.Seurat.embeddings, x=obj.list)
```

`plot.discriminant.3d` *3D plot of reference map with extra discriminant dimension*

Description

Add an extra dimension to the reference map (it can be suggested by 'find.discriminant.dimensions'), to explore additional axes of variability in a query dataset compared to the reference map.

Usage

```
## S3 method for class 'discriminant.3d'
plot(
  ref,
  query,
  query.control = NULL,
  query.assay = "RNA",
  labels.col = "functional.cluster",
  extra.dim = "ICA_1",
  query.state = NULL
)
```

Arguments

`ref`	Seurat object with reference object
`query`	Seurat object with query data
`query.control`	Optionally, you can compare your query with a control sample, instead of the reference
`query.assay`	The data slot to be used for enrichment analysis
`labels.col`	The metadata field used to annotate the clusters
`extra.dim`	The additional dimension to be added on the z-axis of the plot. Can be either: • An ICA or PCA dimension (e.g. ICA_10). See 'find.discriminant.dimensions' • Any numeric metadata field associated to the cells (e.g. 'cycling.score')
`query.state`	Only plot the query cells from this specific state

Value

A three dimensional plot with UMAP_1 and UMAP_2 on the x and y axis respectively, and the specified 'extra.dim' on the z-axis.

Examples

```
plot.discriminant.3d(ref, query=query, extra.dim="ICA_19")
plot.discriminant.3d(ref, query=treated.set, query.control=control.set, extra.dim="ICA_2")
```

plot.projection *Show UMAP projection of query on reference map*

Description

Plots the UMAP representation of the reference map, together with the projected coordinates of a query dataset.

Usage

```
## S3 method for class 'projection'
plot(
  ref,
  query = NULL,
  labels.col = "functional.cluster",
  cols = NULL,
  linesize = 1,
  pointsize = 1,
  density_adjust = 1,
  ref.alpha = 0.3,
  ref.size = NULL,
  ...
)
```

Arguments

ref	Reference object
query	Seurat object with query data
labels.col	The metadata field to annotate the clusters (default: functional.cluster)
cols	Custom color palette for clusters
linesize	Contour line thickness for projected query
pointsize	Point size for cells in projected query
density_adjust	Adjust factor for contour line density
ref.alpha	Transparency parameter for reference cells
ref.size	Adjust point size for reference cells
...	Additional parameters for DimPlot, e.g. raster=T to limit image size

Value

UMAP plot of reference map with projected query set in the same space

Examples

```
data(query_example_seurat)
ref <- load.reference.map()
q <- Run.ProjecTILs(query_example_seurat, ref=ref, fast.umap.predict=TRUE)
plot.projection(ref=ref, query=q)
```

`plot.statepred.composition` *Summarize the predicted cell states of an object*

Description

Makes a barplot of the frequency of cell states in a query object.

Usage

```
## S3 method for class 'statepred.composition'
plot(
  ref,
  query,
  labels.col = "functional.cluster",
  cols = NULL,
  metric = c("Count", "Percent")
)
```

Arguments

`ref`	Reference object
`query`	Seurat object with query data
`labels.col`	The metadata field used to annotate the clusters (default: functional.cluster)
`cols`	Custom color palette for clusters
`metric`	One of 'Count' or 'Percent'. 'Count' plots the absolute number of cells, 'Percent' the fraction on the total number of cells.

Value

Barplot of predicted state composition

Examples

```
data(query_example_seurat)
ref <- load.reference.map()
q <- make.projection(query_example_seurat, ref=ref)
q <- cellstate.predict(ref, query=q)
plot.statepred.composition(query_example.seurat)
```

plot.states.radar *Show expression level of key genes*

Description

Makes a radar plot of the expression level of a set of genes. It can be useful to compare the gene expression profile of different cell states in the reference atlas vs. a projected set.

Usage

```
## S3 method for class 'states.radar'
plot(
  ref,
  query = NULL,
  labels.col = "functional.cluster",
  ref.assay = "RNA",
  query.assay = "RNA",
  genes4radar = c("Foxp3", "Cd4", "Cd8a", "Tcf7", "Ccr7", "Gzmb", "Gzmk", "Pdcd1",
    "Havcr2", "Tox", "Mki67"),
  meta4radar = NULL,
  norm.factor = 1,
  min.cells = 20,
  cols = NULL,
  return = FALSE,
  return.as.list = FALSE
)
```

Arguments

ref	Reference object
query	Query data, either as a Seurat object or as a list of Seurat objects
labels.col	The metadata field used to annotate the clusters
ref.assay	The assay to pull the reference expression data
query.assay	The assay to pull the query expression data
genes4radar	Which genes to use for plotting
meta4radar	Which metadata columns (numeric) to use for plotting. If not NULL, genes4radar are ignored
norm.factor	Normalization factor for rescaling expression or metadata values
min.cells	Only display cell states with a minimum number of cells
cols	Custom color palette for samples in radar plot
return	Return the combined plots instead of printing them to the default device (deprecated)
return.as.list	Return plots in a list, instead of combining them in a single plot

Value

Radar plot of gene expression of key genes by cell subtype

Examples

```
ref <- load.reference.map()
plot.states.radar(ref)
```

`ProjecTILs.classifier` *Annotate query dataset using a reference object*

Description

Apply label transfer to annotate a query dataset with the cell types of a reference object. Compared to Run.ProjecTILs, only cell labels are returned. The low-dim embeddings of the query object (PCA, UMAP) are not modified.

Usage

```
ProjecTILs.classifier(
  query,
  ref = NULL,
  filter.cells = TRUE,
  split.by = NULL,
  reduction = "pca",
  ndim = NULL,
  k = 5,
  nn.decay = 0.1,
  min.confidence = 0.2,
  labels.col = "functional.cluster",
  overwrite = TRUE,
  ncores = 1,
  ...
)
```

Arguments

`query`	Query data, either as single Seurat object or as a list of Seurat object
`ref`	Reference Atlas - if NULL, downloads the default TIL reference atlas
`filter.cells`	Pre-filter cells using 'scGate'. Only set to FALSE if the dataset has been previously subset to cell types represented in the reference.
`split.by`	Grouping variable to split the query object (e.g. if the object contains multiple samples)
`reduction`	The dimensionality reduction used to assign cell type labels
`ndim`	The number of dimensions used for cell type classification

k	Number of neighbors for cell type classification
nn.decay	Weight decay for internal nearest neighbors (between 0 and 1)
min.confidence	Minimum confidence score to return cell type labels (otherwise NA)
labels.col	The metadata field with label annotations of the reference, which will be transferred to the query dataset
overwrite	Replace any existing labels in labels.col with new labels. This may be useful for predicting cell types using multiple reference maps; run this function with overwrite=FALSE to combine existing labels with new labels from a second reference map.
ncores	Number of cores for parallel processing
...	Additional parameters to make.projection

Details

See load.reference.map to load or download a reference atlas. See Run.ProjecTILs to embed the query in the same space of the reference

Value

The query object with a additional metadata columns containing predicted cell labels and confidence scores for the predicted cell labels If cells were filtered prior to projection, they will be labeled as 'NA'

Examples

```
## Not run:
data(query_example_seurat)
ref <- load.reference.map()
q <- ProjecTILs.classifier(query_example_seurat, ref=ref)
table(q$functional.cluster, useNA="ifany")

## End(Not run)
```

---

query_example_seurat *Test dataset for ProjecTILs*

Description

A small dataset of CD8 T cells, to test the ProjecTILs installation.

Usage

```
query_example_seurat
```

Format

A Seurat object

Source

https://pmc.ncbi.nlm.nih.gov/articles/PMC6673650/

`read.sc.query` *Read to memory a query expression matrix*

Description

Load a query expression matrix to be projected onto the reference atlas. Several formats (10x, hdf5, raw and log counts) are supported - see `type` parameter for details

Usage

```
read.sc.query(
  filename,
  type = c("10x", "hdf5", "raw", "raw.log2"),
  project.name = "Query",
  min.cells = 3,
  min.features = 50,
  gene.column.10x = 2,
  raw.rownames = 1,
  raw.sep = c("auto", " ", "\t", ","),
  raw.header = TRUE,
  use.readmtx = TRUE
)
```

Arguments

`filename`	Path to expression matrix file or folder
`type`	Expression matrix format (10x, hdf5, raw, raw.log2)
`project.name`	Title for the project
`min.cells`	Only keep genes represented in at least min.cells number of cells
`min.features`	Only keep cells expressing at least min.features genes
`gene.column.10x`	For 10x format - which column of genes.tsv or features.tsv to use for gene names
`raw.rownames`	For raw matrix format - A vector of row names, or a single number giving the column of the table which contains the row names
`raw.sep`	For raw matrix format - Separator for raw expression matrix
`raw.header`	For raw matrix format - Use headers in expression matrix
`use.readmtx`	Use ReadMtx function to read in 10x files with custom names

Value

A Seurat object populated with raw counts and normalized counts for single-cell expression

Examples

```
fname <- "./sample_data"
querydata <- read.sc.query(fname, type="10x")
```

`recalculate.embeddings` *Recalculate low dimensional embeddings after projection*

Description

Given a reference object and a (list of) projected objects, recalculate low-dim embeddings accounting for the projected cells

Usage

```
recalculate.embeddings(
  ref,
  projected,
  ref.assay = "integrated",
  proj.assay = "integrated",
  ndim = NULL,
  n.neighbors = 20,
  min.dist = 0.3,
  recalc.pca = FALSE,
  resol = 0.4,
  k.param = 15,
  metric = "cosine",
  umap.method = c("umap", "uwot"),
  seed = 123
)
```

Arguments

`ref`	Reference map
`projected`	A projected object (or list of projected objects) generated using make.projection
`ref.assay`	Assay for reference object
`proj.assay`	Assay for projected object(s)
`ndim`	Number of dimensions for recalculating dimensionality reductions
`n.neighbors`	Number of neighbors for UMAP algorithm
`min.dist`	Tightness parameter for UMAP embedding
`recalc.pca`	Whether to recalculate the PCA embeddings with the combined reference and projected data
`resol`	Resolution for unsupervised clustering

k.param	Number of nearest neighbors for clustering
metric	Distance metric to use to find nearest neighbors for UMAP
umap.method	Which method should be used to calculate UMAP embeddings
seed	Random seed for reproducibility

Value

A combined reference object of reference and projected object(s), with new low dimensional embeddings

Examples

```
combined <- recalculate.embeddings(ref, projected, ndim=10)
```

Run.ProjecTILs *Project a query scRNA-seq dataset onto a reference atlas*

Description

This function allows projecting ("query") single-cell RNA-seq datasets onto a reference map (i.e. a curated and annotated scRNA-seq dataset). To project multiple datasets, submit a list of Seurat objects with the query parameter. The projection consists of 3 steps:

- pre-processing: optional steps which might include pre-filtering of cells by markers using 'scGate', data normalization, and ortholog conversion.
- batch-effect correction: uses built-in STACAS algorithm to detect and correct for batch effects (this step assumes that at least a fraction of the cells in the query are in the same state than cells in the reference)
- embedding of corrected query data in the reduced-dimensionality spaces (PCA and UMAP) of the reference map.

This function acts as a wrapper for make.projection and cellstate.predict

Usage

```
Run.ProjecTILs(
  query,
  ref = NULL,
  filter.cells = TRUE,
  split.by = NULL,
  reduction = "pca",
  ndim = NULL,
  k = 5,
  nn.decay = 0.1,
  min.confidence = 0.2,
  labels.col = "functional.cluster",
  ...
)
```

Arguments

query	Query data, either as single Seurat object or as a list of Seurat object
ref	Reference Atlas - if NULL, downloads the default TIL reference atlas
filter.cells	Pre-filter cells using 'scGate'. Only set to FALSE if the dataset has been previously subset to cell types represented in the reference.
split.by	Grouping variable to split the query object (e.g. if the object contains multiple samples)
reduction	The dimensionality reduction used to assign cell type labels, based on majority voting of nearest neighbors between reference and query.
ndim	The number of dimensions used for cell type classification
k	Number of neighbors for cell type classification
nn.decay	Weight decay for internal nearest neighbors (between 0 and 1)
min.confidence	Minimum confidence score to return cell type labels (otherwise NA)
labels.col	The metadata field of the reference to annotate the clusters
...	Additional parameters to make.projection

Details

See load.reference.map to load or download a reference atlas. See also ProjecTILs.classifier to use ProjecTILs as a cell type classifier.

Value

An augmented Seurat object with projected UMAP coordinates on the reference map and cell classifications

Examples

```
data(query_example_seurat)
ref <- load.reference.map()
q <- Run.ProjecTILs(query_example_seurat, ref=ref, fast.umap.predict=TRUE)
plot.projection(ref=ref, query=q)
```

Index