Package 'scCustomize'

August 26, 2025

Type Package

Title Custom Visualizations & Functions for Streamlined Analyses of Single Cell Sequencing

Description Collection of functions created and/or curated to aid in the visualization and analysis of single-cell data using 'R'. 'scCustomize' aims to provide 1) Customized visualizations for aid in ease of use and to create more aesthetic and functional visuals. 2) Improve speed/reproducibility of common tasks/pieces of code in scRNA-seq analysis with a single or group of functions. For citation please use: Marsh SE (2021) ``Custom Visualizations & Functions for Streamlined Analyses of Single Cell Sequencing" <doi:10.5281/zenodo.5706430> RRID:SCR_024675.

Version 3.1.3

Date 2025-08-25

URL https://github.com/samuel-marsh/scCustomize, https://samuel-marsh.github.io/scCustomize/, https://doi.org/10.5281/zenodo.5706431

BugReports https://github.com/samuel-marsh/scCustomize/issues

Depends R (>= 4.0.0), Seurat (>= 4.3.0.1)

Imports circlize, cli (>= 3.2.0), cowplot, data.table, dplyr, forcats, ggbeeswarm, ggplot2, ggprism, ggrastr, ggrepel, glue, grDevices, grid, janitor, lifecycle, magrittr, Matrix (>= 1.5.0), methods, paletteer, patchwork, pbapply, purrr, rlang (>= 1.1.3), scales, scattermore (>= 1.2), SeuratObject (>= 5.0.0), stats, stringi, stringr, tibble, tidyr

Suggests BiocFileCache, ComplexHeatmap, dittoSeq, DropletUtils, ggpubr, hdf5r, knitr, Nebulosa, remotes, reticulate, rliger, rmarkdown, scuttle, tidyselect, qs, viridis

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.3.2

NeedsCompilation no

Author Samuel Marsh [aut, cre] (ORCID:
<https://orcid.org/0000-0002-3012-6945>),
Ming Tang [ctb],
Velina Kozareva [ctb],
Lucas Graybuck [ctb],
Zoe Clarke [ctb] (ORCID: <https://orcid.org/0000-0003-0336-8049>)

Maintainer Samuel Marsh <sccustomize@gmail.com>

Repository CRAN

Date/Publication 2025-08-26 08:30:02 UTC

Contents

`Add_Alt_Feature_ID` *Add Alternative Feature IDs*

Description

Add alternative feature ids data.frame to the misc slot of Seurat object.

Usage

```
Add_Alt_Feature_ID(
  seurat_object,
  features_tsv_file = NULL,
  hdf5_file = NULL,
  assay = NULL,
  data_name = "feature_id_mapping_table",
  overwrite = FALSE
)
```

Arguments

`seurat_object`	object name.
`features_tsv_file`	output file from Cell Ranger used for creation of Seurat object. (Either provide this of `hdf5_file`)
`hdf5_file`	output file from Cell Ranger used for creation of Seurat object. (Either provide this of `features_tsv_file`)
`assay`	name of assay(s) to add the alternative features to. Can specify "all" to add to all assays.
`data_name`	name to use for data.frame when stored in `@misc` slot.
`overwrite`	logical, whether to overwrite item with the same `data_name` in the `@misc` slot of object (default is FALSE).

Value

Seurat Object with new entries in the `obj@misc` slot.

Examples

```
## Not run:
# Using features.tsv.gz file
   # Either file from filtered or raw outputs can be used as they are identical.
obj <- Add_Alt_Feature_ID(seurat_object = obj,
features_tsv = "sample01/outs/filtered_feature_bc_matrix/features.tsv.gz", assay = "RNA")

#' # Using hdf5 file
  # Either filtered_feature_bc or raw_feature_bc can be used as the features slot is identical
   # Though it is faster to load filtered_feature_bc file due to droplet filtering
```

```
obj <- Add_Alt_Feature_ID(seurat_object = obj,
hdf5_file = "sample01/outs/outs/filtered_feature_bc_matrix.h5", assay = "RNA")

## End(Not run)
```

`Add_CellBender_Diff` *Calculate and add differences post-cell bender analysis*

Description

Calculate the difference in features and UMIs per cell when both cell bender and raw assays are present.

Usage

```
Add_CellBender_Diff(seurat_object, raw_assay_name, cell_bender_assay_name)
```

Arguments

`seurat_object`	object name.
`raw_assay_name`	name of the assay containing the raw data.
`cell_bender_assay_name`	name of the assay containing the Cell Bender'ed data.

Value

Seurat object with 2 new columns in the meta.data slot.

Examples

```
## Not run:
object <- Add_CellBender_Diff(seurat_object = obj, raw_assay_name = "RAW",
cell_bender_assay_name = "RNA")

## End(Not run)
```

`Add_Cell_Complexity` *Add Cell Complexity*

Description

Add measure of cell complexity/novelty (log10GenesPerUMI) for data QC.

Usage

```
Add_Cell_Complexity(object, ...)

## S3 method for class 'liger'
Add_Cell_Complexity(
  object,
  meta_col_name = "log10GenesPerUMI",
  overwrite = FALSE,
  ...
)

## S3 method for class 'Seurat'
Add_Cell_Complexity(
  object,
  meta_col_name = "log10GenesPerUMI",
  assay = "RNA",
  overwrite = FALSE,
  ...
)
```

Arguments

`object`	Seurat or LIGER object
`...`	Arguments passed to other methods
`meta_col_name`	name to use for new meta data column. Default is "log10GenesPerUMI".
`overwrite`	Logical. Whether to overwrite existing an meta.data column. Default is FALSE meaning that function will abort if column with name provided to `meta_col_name` is present in meta.data slot.
`assay`	assay to use in calculation. Default is "RNA". *Note* This should only be changed if storing corrected and uncorrected assays in same object (e.g. outputs of both Cell Ranger and Cell Bender).

Value

An object of the same class as `object` with columns added to object meta data.

Examples

```
## Not run:
# Liger
liger_object <- Add_Cell_Complexity(object = liger_object)

## End(Not run)

# Seurat
library(Seurat)
pbmc_small <- Add_Cell_Complexity(object = pbmc_small)
```

`Add_Cell_QC_Metrics` *Add Multiple Cell Quality Control Values with Single Function*

Description

Add Mito/Ribo %, Cell Complexity (log10GenesPerUMI), Top Gene Percent with single function call to Seurat or liger objects.

Usage

```
Add_Cell_QC_Metrics(object, ...)

## S3 method for class 'liger'
Add_Cell_QC_Metrics(
  object,
  add_mito_ribo = TRUE,
  add_complexity = TRUE,
  add_top_pct = TRUE,
  add_MSigDB = TRUE,
  add_IEG = TRUE,
  add_hemo = TRUE,
  add_lncRNA = TRUE,
  add_cell_cycle = TRUE,
  species,
  mito_name = "percent_mito",
  ribo_name = "percent_ribo",
  mito_ribo_name = "percent_mito_ribo",
  complexity_name = "log10GenesPerUMI",
  top_pct_name = NULL,
  oxphos_name = "percent_oxphos",
  apop_name = "percent_apop",
  dna_repair_name = "percent_dna_repair",
  ieg_name = "percent_ieg",
  hemo_name = "percent_hemo",
  lncRNA_name = "percent_lncRNA",
```

```
  mito_pattern = NULL,
  ribo_pattern = NULL,
  hemo_pattern = NULL,
  mito_features = NULL,
  ribo_features = NULL,
  hemo_features = NULL,
  ensembl_ids = FALSE,
  num_top_genes = 50,
  assay = NULL,
  list_species_names = FALSE,
  overwrite = FALSE,
  ...
)

## S3 method for class 'Seurat'
Add_Cell_QC_Metrics(
  object,
  species,
  add_mito_ribo = TRUE,
  add_complexity = TRUE,
  add_top_pct = TRUE,
  add_MSigDB = TRUE,
  add_IEG = TRUE,
  add_IEG_module_score = TRUE,
  add_hemo = TRUE,
  add_lncRNA = TRUE,
  add_cell_cycle = TRUE,
  mito_name = "percent_mito",
  ribo_name = "percent_ribo",
  mito_ribo_name = "percent_mito_ribo",
  complexity_name = "log10GenesPerUMI",
  top_pct_name = NULL,
  oxphos_name = "percent_oxphos",
  apop_name = "percent_apop",
  dna_repair_name = "percent_dna_repair",
  ieg_name = "percent_ieg",
  ieg_module_name = "ieg_score",
  hemo_name = "percent_hemo",
  lncRNA_name = "percent_lncRNA",
  mito_pattern = NULL,
  ribo_pattern = NULL,
  hemo_pattern = NULL,
  mito_features = NULL,
  ribo_features = NULL,
  hemo_features = NULL,
  ensembl_ids = FALSE,
  num_top_genes = 50,
  assay = NULL,
```

```
  list_species_names = FALSE,
  overwrite = FALSE,
  ...
)
```

Arguments

object	Seurat or LIGER object
...	Arguments passed to other methods
add_mito_ribo	logical, whether to add percentage of counts belonging to mitochondrial/ribosomal genes to object (Default is TRUE).
add_complexity	logical, whether to add Cell Complexity to object (Default is TRUE).
add_top_pct	logical, whether to add Top Gene Percentages to object (Default is TRUE).
add_MSigDB	logical, whether to add percentages of counts belonging to genes from of mSigDB hallmark gene lists: "HALLMARK_OXIDATIVE_PHOSPHORYLATION", "HALLMARK_APOPTOSIS", and "HALLMARK_DNA_REPAIR" to object (Default is TRUE).
add_IEG	logical, whether to add percentage of counts belonging to IEG genes to object (Default is TRUE).
add_hemo	logical, whether to add percentage of counts belonging to homoglobin genes to object (Default is TRUE).
add_lncRNA	logical, whether to add percentage of counts belonging to lncRNA genes to object (Default is TRUE).
add_cell_cycle	logical, whether to addcell cycle scores and phase based on CellCycleScoring. Only applicable if species = "human". (Default is TRUE).
species	Species of origin for given Seurat Object. If mouse, human, marmoset, zebrafish, rat, drosophila, rhesus macaque, or chicken (name or abbreviation) are provided the function will automatically generate patterns and features.
mito_name	name to use for the new meta.data column containing percent mitochondrial counts. Default is "percent_mito".
ribo_name	name to use for the new meta.data column containing percent ribosomal counts. Default is "percent_ribo".
mito_ribo_name	name to use for the new meta.data column containing percent mitochondrial+ribosomal counts. Default is "percent_mito_ribo".
complexity_name	name to use for new meta data column for Add_Cell_Complexity. Default is "log10GenesPerUMI".
top_pct_name	name to use for new meta data column for Add_Top_Gene_Pct. Default is "percent_topXX", where XX is equal to the value provided to num_top_genes.
oxphos_name	name to use for new meta data column for percentage of MSigDB oxidative phosphorylation counts. Default is "percent_oxphos".
apop_name	name to use for new meta data column for percentage of MSigDB apoptosis counts. Default is "percent_apop".

`dna_repair_name`
: name to use for new meta data column for percentage of MSigDB DNA repair counts. Default is "percent_dna_repair"..

`ieg_name`
: name to use for new meta data column for percentage of IEG counts. Default is "percent_ieg".

`hemo_name`
: name to use for the new meta.data column containing percent hemoglobin counts. Default is "percent_mito".

`lncRNA_name`
: name to use for the new meta.data column containing percent lncRNA counts. Default is "percent_lncRNA".

`mito_pattern`
: A regex pattern to match features against for mitochondrial genes (will set automatically if species is mouse or human; marmoset features list saved separately).

`ribo_pattern`
: A regex pattern to match features against for ribosomal genes (will set automatically if species is in default list).

`hemo_pattern`
: A regex pattern to match features against for hemoglobin genes (will set automatically if species is in default list).

`mito_features`
: A list of mitochondrial gene names to be used instead of using regex pattern. Will override regex pattern if both are present (including default saved regex patterns).

`ribo_features`
: A list of ribosomal gene names to be used instead of using regex pattern. Will override regex pattern if both are present (including default saved regex patterns).

`hemo_features`
: A list of hemoglobin gene names to be used instead of using regex pattern. Will override regex pattern if both are present (including default saved regex patterns).

`ensembl_ids`
: logical, whether feature names in the object are gene names or ensembl IDs (default is FALSE; set TRUE if feature names are ensembl IDs).

`num_top_genes`
: An integer vector specifying the size(s) of the top set of high-abundance genes. Used to compute the percentage of library size occupied by the most highly expressed genes in each cell.

`assay`
: assay to use in calculation. Default is "RNA". *Note* This should only be changed if storing corrected and uncorrected assays in same object (e.g. outputs of both Cell Ranger and Cell Bender).

`list_species_names`
: returns list of all accepted values to use for default species names which contain internal regex/feature lists (human, mouse, marmoset, zebrafish, rat, drosophila, rhesus macaque, and chicken). Default is FALSE.

`overwrite`
: Logical. Whether to overwrite existing an meta.data column. Default is FALSE meaning that function will abort if column with name provided to `meta_col_name` is present in meta.data slot.

`add_IEG_module_score`
: logical, whether to add module score belonging to IEG genes to object (Default is TRUE).

`ieg_module_name`
: name to use for new meta data column for module score of IEGs. Default is "ieg_score".

Value

A liger Object

A Seurat Object

Examples

```
## Not run:
obj <- Add_Cell_QC_Metrics(object = obj, species = "Human")

## End(Not run)

## Not run:
obj <- Add_Cell_QC_Metrics(object = obj, species = "Human")

## End(Not run)
```

Add_Hemo *Add Hemoglobin percentages*

Description

Add hemoglobin percentages to meta.data slot of Seurat Object or cell.data/cellMeta slot of Liger object

Usage

```
Add_Hemo(object, ...)

## S3 method for class 'liger'
Add_Hemo(
  object,
  species,
  hemo_name = "percent_hemo",
  hemo_pattern = NULL,
  hemo_features = NULL,
  ensembl_ids = FALSE,
  overwrite = FALSE,
  list_species_names = FALSE,
  ...
)

## S3 method for class 'Seurat'
Add_Hemo(
  object,
  species,
  hemo_name = "percent_hemo",
```

```
  hemo_pattern = NULL,
  hemo_features = NULL,
  ensembl_ids = FALSE,
  assay = NULL,
  overwrite = FALSE,
  list_species_names = FALSE,
  ...
)
```

Arguments

object	Seurat or LIGER object
...	Arguments passed to other methods
species	Species of origin for given Seurat Object. If mouse, human, marmoset, zebrafish, rat, drosophila, rhesus macaque, or chicken (name or abbreviation) are provided the function will automatically generate hemo_pattern values.
hemo_name	name to use for the new meta.data column containing percent hemoglobin counts. Default is "percent_hemo".
hemo_pattern	A regex pattern to match features against for hemoglobin genes (will set automatically if species is mouse or human; marmoset features list saved separately).
hemo_features	A list of hemoglobin gene names to be used instead of using regex pattern.
ensembl_ids	logical, whether feature names in the object are gene names or ensembl IDs (default is FALSE; set TRUE if feature names are ensembl IDs).
overwrite	Logical. Whether to overwrite existing meta.data columns. Default is FALSE meaning that function will abort if columns with any one of the names provided to hemo_name is present in meta.data slot.
list_species_names	returns list of all accepted values to use for default species names which contain internal regex/feature lists (human, mouse, marmoset, zebrafish, rat, drosophila, and rhesus macaque). Default is FALSE.
assay	Assay to use (default is the current object default assay).

Value

An object of the same class as object with columns added to object meta data.

Examples

```
## Not run:
# Liger
liger_object <- Add_Hemo(object = liger_object, species = "human")

## End(Not run)

## Not run:
# Seurat
seurat_object <- Add_Hemo(object = seurat_object, species = "human")
```

```
## End(Not run)
```

`Add_MALAT1_Threshold` *Add MALAT1 QC Threshold*

Description

Adds TRUE/FALSE values to each cell based on calculation of MALAT1 threshold. This function incorporates a threshold calculation and procedure as described in Clarke & Bader (2024). bioRxiv doi:10.1101/2024.07.14.603469. Please cite this preprint whenever using this function.

Usage

```
Add_MALAT1_Threshold(object, ...)

## S3 method for class 'Seurat'
Add_MALAT1_Threshold(
  object,
  species,
  sample_col = NULL,
  malat1_threshold_name = NULL,
  ensembl_ids = FALSE,
  assay = NULL,
  overwrite = FALSE,
  print_plots = NULL,
  save_plots = FALSE,
  save_plot_path = NULL,
  save_plot_name = NULL,
  plot_width = 11,
  plot_height = 8,
  whole_object = FALSE,
  homolog_name = NULL,
  bw = 0.1,
  lwd = 2,
  breaks = 100,
  chosen_min = 1,
  smooth = 1,
  abs_min = 0.3,
  rough_max = 2,
  ...
)
```

Arguments

`object`	Seurat or LIGER object

...	Arguments passed to other methods
species	Species of origin for given Seurat Object. Only accepted species are: mouse, human (name or abbreviation).
sample_col	column name in meta.data that contains sample ID information.
malat1_threshold_name	name to use for the new meta.data column containing percent IEG gene counts. Default is set dependent on species gene symbol.
ensembl_ids	logical, whether feature names in the object are gene names or ensembl IDs (default is FALSE; set TRUE if feature names are ensembl IDs).
assay	Assay to use (default is the current object default assay).
overwrite	Logical. Whether to overwrite existing meta.data columns. Default is FALSE meaning that function will abort if columns with the name provided to malat1_threshold_name is present in meta.data slot.
print_plots	logical, should plots be printed to output when running function (default is NULL). Will automatically set to FALSE if performing across samples or TRUE if performing across whole object.
save_plots	logical, whether or not to save plots to pdf (default is FALSE).
save_plot_path	path to save location for plots (default is NULL; current working directory).
save_plot_name	name for pdf file containing plots.
plot_width	the width (in inches) for output page size. Default is 11.
plot_height	the height (in inches) for output page size. Default is 8.
whole_object	logical, whether to perform calculation on whole object (default is FALSE). Should be only be run if object contains single sample.
homolog_name	feature name for MALAT1 homolog in non-default species (if annotated).
bw	The "bandwidth" value when plotting the density function to the MALAT1 distribution; default is bw = 0.1, but this parameter should be lowered (e.g. to 0.01) if you run the function and the line that's produced doesn't look like it's tracing the shape of the histogram accurately (this will make the line less "stiff" and more fitted to the data)
lwd	The "line width" fed to the abline function which adds the vertical red line to the output plots; default is 2, and it can be increased or decreased depending on the user's plotting preferences
breaks	The number of bins used for plotting the histogram of normalized MALAT1 values; default is 100
chosen_min	The minimum MALAT1 value cutoff above which a MALAT1 peak in the density function should be found. This value is necessary to determine which peak in the density function fitted to the MALAT1 distribution is likely representative of what we would expect to find in real cells. This is because some samples may have large numbers of cells or empty droplets with lower than expected normalized MALAT1 values, and therefore have a peak close to or at zero. Ideally, "chosen_min" would be manually chosen after looking at a histogram of MALAT1 values, and be the normalized MALAT1 value that cuts out all of the

cells that look like they stray from the expected distribution (a unimodal distribution above zero). The default value is 1 as this works well in many test cases, but different types of normalization may make the user want to change this parameter (e.g. Seurat's original normalization function generates different results to their SCT function) which may change the MALAT1 distribution). Increase or decrease chosen_min depending on where your MALAT1 peak is located.

`smooth` The "smoothing parameter" fed into the "smooth.spline" function that adjusts the trade-off between the smoothness of the line fitting the histogram, and how closely it fits the histogram; the default is 1, and can be lowered if it looks like the line is underfitting the data, and raised in the case of overfitting. The ideal scenario is for the line to trace the histogram in a way where the only inflection point(s) are between major peaks, e.g. separating the group of poor-quality cells or empty droplets with lower normalized MALAT1 expression from higher-quality cells with higher normalized MALAT1 expression.

`abs_min` The absolute lowest value allowed as the MALAT1 threshold. This parameter increases the robustness of the function if working with an outlier data distribution (e.g. an entire sample is poor quality so there is a unimodal MALAT1 distribution that is very low but above zero, but also many values close to zero) and prevents a resulting MALAT1 threshold of zero. In the case where a calculated MALAT1 value is zero, the function will return 0.3 by default.

`rough_max` A rough value for the location of a MALAT1 peak if a peak is not found. This is possible if there are so few cells with higher MALAT1 values, that a distribution fitted to the data finds no local maxima. For example, if a sample only has poor-quality cells such that all have near-zero MALAT1 expression, the fitted function may look similar to a positive quadratic function which has no local maxima. In this case, the function searches for the closest MALAT1 value to the default value, 2, to use in place of a real local maximum.

Value

Seurat object with added meta.data column

Author(s)

Zoe Clark (original function and manuscript) & Samuel Marsh (wrappers and updates for inclusion in package)

References

This function incorporates a threshold calculation and procedure as described in Clarke & Bader (2024). bioRxiv doi:10.1101/2024.07.14.603469. Please cite this preprint whenever using this function.

Examples

```
## Not run:
object <- Add_MALAT1_Threshold(object = object, species = "Human")

## End(Not run)
```

Add_Mito_Ribo *Add Mito and Ribo percentages*

Description

Add Mito, Ribo, & Mito+Ribo percentages to meta.data slot of Seurat Object or cell.data slot of Liger object

Usage

```
Add_Mito_Ribo(object, ...)

## S3 method for class 'liger'
Add_Mito_Ribo(
  object,
  species,
  mito_name = "percent_mito",
  ribo_name = "percent_ribo",
  mito_ribo_name = "percent_mito_ribo",
  mito_pattern = NULL,
  ribo_pattern = NULL,
  mito_features = NULL,
  ribo_features = NULL,
  ensembl_ids = FALSE,
  overwrite = FALSE,
  list_species_names = FALSE,
  ...
)

## S3 method for class 'Seurat'
Add_Mito_Ribo(
  object,
  species,
  mito_name = "percent_mito",
  ribo_name = "percent_ribo",
  mito_ribo_name = "percent_mito_ribo",
  mito_pattern = NULL,
  ribo_pattern = NULL,
  mito_features = NULL,
  ribo_features = NULL,
  ensembl_ids = FALSE,
  assay = NULL,
  overwrite = FALSE,
  list_species_names = FALSE,
  species_prefix = NULL,
```

```
  ...
)
```

Arguments

object	Seurat or LIGER object
...	Arguments passed to other methods
species	Species of origin for given Seurat Object. If mouse, human, marmoset, zebrafish, rat, drosophila, rhesus macaque, or chicken (name or abbreviation) are provided the function will automatically generate mito_pattern and ribo_pattern values.
mito_name	name to use for the new meta.data column containing percent mitochondrial counts. Default is "percent_mito".
ribo_name	name to use for the new meta.data column containing percent ribosomal counts. Default is "percent_ribo".
mito_ribo_name	name to use for the new meta.data column containing percent mitochondrial+ribosomal counts. Default is "percent_mito_ribo".
mito_pattern	A regex pattern to match features against for mitochondrial genes (will set automatically if species is mouse, human, zebrafish, rat, drosophila, rhesus macaque, or chicken; marmoset features list saved separately).
ribo_pattern	A regex pattern to match features against for ribosomal genes (will set automatically if species is mouse, human, marmoset, zebrafish, rat, drosophila, rhesus macaque, or chicken).
mito_features	A list of mitochondrial gene names to be used instead of using regex pattern. Will override regex pattern if both are present (including default saved regex patterns).
ribo_features	A list of ribosomal gene names to be used instead of using regex pattern. Will override regex pattern if both are present (including default saved regex patterns).
ensembl_ids	logical, whether feature names in the object are gene names or ensembl IDs (default is FALSE; set TRUE if feature names are ensembl IDs).
overwrite	Logical. Whether to overwrite existing meta.data columns. Default is FALSE meaning that function will abort if columns with any one of the names provided to mito_name ribo_name or mito_ribo_name is present in meta.data slot.
list_species_names	returns list of all accepted values to use for default species names which contain internal regex/feature lists (human, mouse, marmoset, zebrafish, rat, drosophila, rhesus macaque, and chicken). Default is FALSE.
assay	Assay to use (default is the current object default assay).
species_prefix	the species prefix in front of gene symbols in object if providing two species for multi-species aligned dataset.

Value

An object of the same class as object with columns added to object meta data.

Examples

```
## Not run:
# Liger
liger_object <- Add_Mito_Ribo(object = liger_object, species = "human")

## End(Not run)

## Not run:
# Seurat
seurat_object <- Add_Mito_Ribo(object = seurat_object, species = "human")

## End(Not run)
```

`Add_Pct_Diff` *Add percentage difference to DE results*

Description

Adds new column labeled "pct_diff" to the data.frame output of `FindMarkers`, `FindAllMarkers`, or other DE test data.frames.

Usage

```
Add_Pct_Diff(
  marker_dataframe,
  pct.1_name = "pct.1",
  pct.2_name = "pct.2",
  overwrite = FALSE
)
```

Arguments

`marker_dataframe`	data.frame containing the results of `FindMarkers`, `FindAllMarkers`, or other DE test data.frame.
`pct.1_name`	the name of data.frame column corresponding to percent expressed in group 1. Default is Seurat default "pct.1".
`pct.2_name`	the name of data.frame column corresponding to percent expressed in group 2. Default is Seurat default "pct.2".
`overwrite`	logical. If the `marker_dataframe` already contains column named "pct_diff" whether to overwrite or return error message. Default is FALSE.

Value

Returns input `marker_dataframe` with additional "pct_diff" column.

Examples

```
## Not run:
marker_df <- FindAllMarkers(object = obj_name)
marker_df <- Add_Pct_Diff(marker_dataframe = marker_df)
# or piped with function
marker_df <- FindAllMarkers(object = obj_name) %>%
  Add_Pct_Diff()

## End(Not run)
```

Add_Sample_Meta *Add Sample Level Meta Data*

Description

Add meta data from ample level data.frame/tibble to cell level seurat `@meta.data` slot

Usage

```
Add_Sample_Meta(
  seurat_object,
  meta_data,
  join_by_seurat,
  join_by_meta,
  na_ok = FALSE,
  overwrite = FALSE
)
```

Arguments

`seurat_object`	object name.
`meta_data`	data.frame/tibble containing meta data or path to file to read. Must be formatted as either data.frame or tibble.
`join_by_seurat`	name of the column in `seurat_object@meta.data` that contains matching variables to `join_by_meta` in `meta_data`.
`join_by_meta`	name of the column in `meta_data` that contains matching variables to `join_by_seurat` in `seurat_object@meta.data`.
`na_ok`	logical, is it ok to add NA values to `seurat_object@meta.data`. Default is FALSE. Be very careful if setting TRUE because if there is error in join operation it may result in all `@meta.data` values being replaced with NA.
`overwrite`	logical, if there are shared columns between `seurat_object@meta.data` and `meta_data` should the current `seurat_object@meta.data` columns be overwritten. Default is FALSE. This parameter excludes values provided to `join_by_seurat` and `join_by_meta`.

Value

Seurat object with new `@meta.data` columns

Examples

```
## Not run:
# meta_data present in environment
sample_level_meta <- data.frame(...)
obj <- Add_Sample_Meta(seurat_object = obj, meta_data = sample_level_meta,
join_by_seurat = "orig.ident", join_by_meta = "sample_ID")

# from meta data file
obj <- Add_Sample_Meta(seurat_object = obj, meta_data = "meta_data/sample_level_meta.csv",
join_by_seurat = "orig.ident", join_by_meta = "sample_ID")

## End(Not run)
```

Add_Top_Gene_Pct *Add Percent of High Abundance Genes*

Description

Add the percentage of counts occupied by the top XX most highly expressed genes in each cell.

Usage

```
Add_Top_Gene_Pct(object, ...)

## S3 method for class 'liger'
Add_Top_Gene_Pct(
  object,
  num_top_genes = 50,
  meta_col_name = NULL,
  overwrite = FALSE,
  verbose = TRUE,
  ...
)

## S3 method for class 'Seurat'
Add_Top_Gene_Pct(
  object,
  num_top_genes = 50,
  meta_col_name = NULL,
  assay = "RNA",
  overwrite = FALSE,
  verbose = TRUE,
  ...
)
```

Arguments

object	Seurat or LIGER object.
...	Arguments passed to other methods
num_top_genes	An integer vector specifying the size(s) of the top set of high-abundance genes. Used to compute the percentage of library size occupied by the most highly expressed genes in each cell.
meta_col_name	name to use for new meta data column. Default is "percent_topXX", where XX is equal to the value provided to num_top_genes.
overwrite	Logical. Whether to overwrite existing an meta.data column. Default is FALSE meaning that function will abort if column with name provided to meta_col_name is present in meta.data slot.
verbose	logical, whether to print messages with status updates, default is TRUE.
assay	assay to use in calculation. Default is "RNA". *Note* This should only be changed if storing corrected and uncorrected assays in same object (e.g. outputs of both Cell Ranger and Cell Bender).

Value

A liger Object

A Seurat Object

References

This function uses scuttle package (license: GPL-3) to calculate the percent of expression coming from top XX genes in each cell. Parameter description for num_top_genes also from scuttle. If using this function in analysis, in addition to citing scCustomize, please cite scuttle: McCarthy DJ, Campbell KR, Lun ATL, Willis QF (2017). "Scater: pre-processing, quality control, normalisation and visualisation of single-cell RNA-seq data in R." Bioinformatics, 33, 1179-1186. doi:10.1093/bioinformatics/btw777.

See Also

https://bioconductor.org/packages/release/bioc/html/scuttle.html

Examples

```
## Not run:
liger_object <- Add_Top_Gene_Pct(object = liger_object, num_top_genes = 50)

## End(Not run)

## Not run:
library(Seurat)
pbmc_small <- Add_Top_Gene_Pct(seurat_object = pbmc_small, num_top_genes = 50)

## End(Not run)
```

`as.anndata` *Convert objects to anndata objects*

Description

Convert objects (Seurat & LIGER) to anndata objects

Usage

```
as.anndata(x, ...)

## S3 method for class 'Seurat'
as.anndata(
  x,
  file_path,
  file_name,
  assay = NULL,
  main_layer = "data",
  other_layers = "counts",
  transer_dimreduc = TRUE,
  verbose = TRUE,
  ...
)

## S3 method for class 'liger'
as.anndata(
  x,
  file_path,
  file_name,
  transfer_norm.data = FALSE,
  reduction_label = NULL,
  add_barcode_names = FALSE,
  barcode_prefix = TRUE,
  barcode_cell_id_delimiter = "_",
  verbose = TRUE,
  ...
)
```

Arguments

`x`	Seurat or LIGER object
`...`	Arguments passed to other methods
`file_path`	directory file path and/or file name prefix. Defaults to current wd.
`file_name`	file name.
`assay`	Assay containing data to use, (default is object default assay).

`main_layer` the layer of data to become default layer in anndata object (default is "data").

`other_layers` other data layers to transfer to anndata object (default is "counts").

`transer_dimreduc`
logical, whether to transfer dimensionality reduction coordinates from Seurat to anndata object (default is TRUE).

`verbose` logical, whether to print status messages during object conversion (default is TRUE).

`transfer_norm.data`
logical, whether to transfer the norm.data in addition to raw.data, default is FALSE.

`reduction_label`
What to label the visualization dimensionality reduction. LIGER does not store name of technique and therefore needs to be set manually.

`add_barcode_names`
logical, whether to add dataset names to the cell barcodes when merging object data, default is FALSE.

`barcode_prefix` logical, if `add_barcode_names = TRUE` should the names be added as prefix to current cell barcodes/names or a suffix (default is TRUE; prefix).

`barcode_cell_id_delimiter`
The delimiter to use when adding dataset id to barcode prefix/suffix. Default is "_".

Value

an anndata object generated from `x`, saved at path provided.

References

Seurat version modified and enhanced version of `sceasy::seurat2anndata` (sceasy package: https://github.com/cellgeni/sceasy; License: GPL-3. Function has additional checks and supports Seurat V3 and V5 object structure.

LIGER version inspired by `sceasy::seurat2anndata` modified and updated to apply to LIGER objects (sceasy package: https://github.com/cellgeni/sceasy; License: GPL-3.

Examples

```
## Not run:
as.anndata(x = seurat_object, file_path = "/folder_name", file_name = "anndata_converted.h5ad")

## End(Not run)

## Not run:
as.anndata(x = liger_object, file_path = "/folder_name", file_name = "anndata_converted.h5ad")

## End(Not run)
```

as.LIGER *Convert objects to LIGER objects*

Description

Convert objects (Seurat & lists of Seurat Objects) to anndata objects

Usage

```
as.LIGER(x, ...)

## S3 method for class 'Seurat'
as.LIGER(
  x,
  group.by = "orig.ident",
  layers_name = NULL,
  assay = "RNA",
  remove_missing = FALSE,
  renormalize = TRUE,
  use_seurat_var_genes = FALSE,
  use_seurat_dimreduc = FALSE,
  reduction = NULL,
  keep_meta = TRUE,
  verbose = TRUE,
  ...
)

## S3 method for class 'list'
as.LIGER(
  x,
  group.by = "orig.ident",
  dataset_names = NULL,
  assay = "RNA",
  remove_missing = FALSE,
  renormalize = TRUE,
  use_seurat_var_genes = FALSE,
  var_genes_method = "intersect",
  keep_meta = TRUE,
  verbose = TRUE,
  ...
)
```

Arguments

x	An object to convert to class liger
...	Arguments passed to other methods

group.by	Variable in meta data which contains variable to split data by, (default is "orig.ident").
layers_name	name of meta.data column used to split layers if setting group.by = "layers".
assay	Assay containing raw data to use, (default is "RNA").
remove_missing	logical, whether to remove missing genes with no counts when converting to LIGER object (default is FALSE).
renormalize	logical, whether to perform normalization after LIGER object creation (default is TRUE).
use_seurat_var_genes	logical, whether to transfer variable features from Seurat object to new LIGER object (default is FALSE).
use_seurat_dimreduc	logical, whether to transfer dimensionality reduction coordinates from Seurat to new LIGER object (default is FALSE).
reduction	Name of Seurat reduction to transfer if use_seurat_dimreduc = TRUE.
keep_meta	logical, whether to transfer columns in Seurat meta.data slot to LIGER cell.data slot (default is TRUE).
verbose	logical, whether to print status messages during object conversion (default is TRUE).
dataset_names	optional, vector of names to use for naming datasets.
var_genes_method	how variable genes should be selected from Seurat objects if use_seurat_var_genes = TRUE. Can be either "intersect" or "union", (default is "intersect").

Value

a liger object generated from x

References

modified and enhanced version of rliger::seuratToLiger.

Examples

```
## Not run:
liger_object <- as.LIGER(x = seurat_object)

## End(Not run)

## Not run:
liger_object <- as.LIGER(x = seurat_object_list)

## End(Not run)
```

`as.Seurat.liger` *Convert objects to* `Seurat` *objects*

Description

Merges raw.data and scale.data of object, and creates Seurat object with these values along with slots containing dimensionality reduction coordinates, iNMF factorization, and cluster assignments. Supports Seurat V3/4 and V4.

Usage

```
## S3 method for class 'liger'
as.Seurat(
  x,
  nms = names(x@H),
  renormalize = TRUE,
  use.liger.genes = TRUE,
  by.dataset = FALSE,
  keep_meta = TRUE,
  reduction_label = "UMAP",
  seurat_assay = "RNA",
  assay_type = NULL,
  add_barcode_names = FALSE,
  barcode_prefix = TRUE,
  barcode_cell_id_delimiter = "_",
  ...
)
```

Arguments

`x`	`liger` object.
`nms`	By default, labels cell names with dataset of origin (this is to account for cells in different datasets which may have same name). Other names can be passed here as vector, must have same length as the number of datasets. (default names(H)).
`renormalize`	Whether to log-normalize raw data using Seurat defaults (default TRUE).
`use.liger.genes`	Whether to carry over variable genes (default TRUE).
`by.dataset`	Include dataset of origin in cluster identity in Seurat object (default FALSE).
`keep_meta`	logical. Whether to transfer additional metadata (nGene/nUMI/dataset already transferred) to new Seurat Object. Default is TRUE.
`reduction_label`	Name of dimensionality reduction technique used. Enables accurate transfer or name to Seurat object instead of defaulting to "tSNE".
`seurat_assay`	Name to set for assay in Seurat Object. Default is "RNA".

assay_type — what type of Seurat assay to create in new object (Assay vs Assay5). Default is NULL which will default to the current user settings. See Convert_Assay parameter convert_to for acceptable values.

add_barcode_names — logical, whether to add dataset names to the cell barcodes when creating Seurat object, default is FALSE.

barcode_prefix — logical, if add_barcode_names = TRUE should the names be added as prefix to current cell barcodes/names or a suffix (default is TRUE; prefix).

barcode_cell_id_delimiter — The delimiter to use when adding dataset id to barcode prefix/suffix. Default is "_".

... — unused.

Details

Stores original dataset identity by default in new object metadata if dataset names are passed in nms. iNMF factorization is stored in dim.reduction object with key "iNMF".

Value

Seurat object with raw.data, scale.data, reduction_label, iNMF, and ident slots set.

Seurat object.

References

Original function is part of LIGER package https://github.com/welch-lab/liger (Licence: GPL-3). Function was modified for use in scCustomize with additional parameters/functionality.

Examples

```
## Not run:
seurat_object <- as.Seurat(x = liger_object)

## End(Not run)
```

Barcode_Plot *Create Barcode Rank Plot*

Description

Plot UMI vs. Barcode Rank with inflection and knee. Requires input from DropletUtils package.

Usage

```
Barcode_Plot(
  br_out,
  pt.size = 6,
  plot_title = "Barcode Ranks",
  raster_dpi = c(1024, 1024),
  plateau = NULL
)
```

Arguments

br_out	DFrame output from barcodeRanks.
pt.size	point size for plotting, default is 6.
plot_title	Title for plot, default is "Barcode Ranks".
raster_dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(1024, 1024).
plateau	numerical value at which to add vertical line designating estimated empty droplet plateau (default is NULL).

Value

A ggplot object

Examples

```
## Not run:
mat <- Read10X_h5(filename = "raw_feature_bc_matrix.h5")

br_results <- DropletUtils::barcodeRanks(mat)

Barcode_Plot(br_out = br_results)

## End(Not run)
```

Blank_Theme *Blank Theme*

Description

Shortcut for thematic modification to remove all axis labels and grid lines

Usage

```
Blank_Theme(...)
```

Arguments

...	extra arguments passed to ggplot2::theme().

Value

Returns a list-like object of class *theme*.

Examples

```
# Generate a plot and customize theme
library(ggplot2)
df <- data.frame(x = rnorm(n = 100, mean = 20, sd = 2), y = rbinom(n = 100, size = 100, prob = 0.2))
p <- ggplot(data = df, mapping = aes(x = x, y = y)) + geom_point(mapping = aes(color = 'red'))
p + Blank_Theme()
```

Case_Check *Check for alternate case features*

Description

Checks Seurat object for the presence of features with the same spelling but alternate case.

Usage

```
Case_Check(
  seurat_object,
  gene_list,
  case_check_msg = TRUE,
  return_features = TRUE,
  assay = NULL
)
```

Arguments

seurat_object	Seurat object name.
gene_list	vector of genes to check.
case_check_msg	logical. Whether to print message to console if alternate case features are found in addition to inclusion in returned list. Default is TRUE.
return_features	logical. Whether to return vector of alternate case features. Default is TRUE.
assay	Name of assay to pull feature names from. If NULL will use the result of DefaultAssay(seurat_object).

Value

If features found returns vector of found alternate case features and prints message depending on parameters specified.

Examples

```
## Not run:
alt_features <- Case_Check(seurat_object = obj_name, gene_list = DEG_list)

## End(Not run)
```

`CellBender_Diff_Plot` *Plot Number of Cells/Nuclei per Sample*

Description

Plot of total cell or nuclei number per sample grouped by another meta data variable.

Usage

```
CellBender_Diff_Plot(
  feature_diff_df,
  pct_diff_threshold = 25,
  num_features = NULL,
  label = TRUE,
  num_labels = 20,
  min_count_label = 1,
  repel = TRUE,
  custom_labels = NULL,
  plot_line = TRUE,
  plot_title = "Raw Counts vs. Cell Bender Counts",
  x_axis_label = "Raw Data Counts",
  y_axis_label = "Cell Bender Counts",
  xnudge = 0,
  ynudge = 0,
  max.overlaps = 100,
  label_color = "dodgerblue",
  fontface = "bold",
  label_size = 3.88,
  bg.color = "white",
  bg.r = 0.15,
  ...
)
```

Arguments

`feature_diff_df`
: name of data.frame created using `CellBender_Feature_Diff`.

`pct_diff_threshold`
: threshold to use for feature plotting. Resulting plot will only contain features which exhibit percent change >= value. Default is 25.

num_features	Number of features to plot. Will ignore pct_diff_threshold and return plot with specified number of features. Default is NULL.
label	logical, whether or not to label the features that have largest percent difference between raw and CellBender counts (Default is TRUE).
num_labels	Number of features to label if label = TRUE, (default is 20).
min_count_label	Minimum number of raw counts per feature necessary to be included in plot labels (default is 1)
repel	logical, whether to use geom_text_repel to create a nicely-repelled labels; this is slow when a lot of points are being plotted. If using repel, set xnudge and ynudge to 0, (Default is TRUE).
custom_labels	A custom set of features to label instead of the features most different between raw and CellBender counts.
plot_line	logical, whether to plot diagonal line with slope = 1 (Default is TRUE).
plot_title	Plot title.
x_axis_label	Label for x axis.
y_axis_label	Label for y axis.
xnudge	Amount to nudge X and Y coordinates of labels by.
ynudge	Amount to nudge X and Y coordinates of labels by.
max.overlaps	passed to geom_text_repel, exclude text labels that overlap too many things. Defaults to 100.
label_color	Color to use for text labels.
fontface	font face to use for text labels ("plain", "bold", "italic", "bold.italic") (Default is "bold").
label_size	text size for feature labels (passed to geom_text_repel).
bg.color	color to use for shadow/outline of text labels (passed to geom_text_repel) (Default is white).
bg.r	radius to use for shadow/outline of text labels (passed to geom_text_repel) (Default is 0.15).
...	Extra parameters passed to geom_text_repel through LabelPoints.

Value

A ggplot object

Examples

```
## Not run:
# get cell bender differences data.frame
cb_stats <- CellBender_Feature_Diff(seurat_object - obj, raw_assay = "RAW",
cell_bender_assay = "RNA")

# plot
CellBender_Diff_Plot(feature_diff_df = cb_stats, pct_diff_threshold = 25)
```

```
## End(Not run)
```

`CellBender_Feature_Diff` *CellBender Feature Differences*

Description

Get quick values for raw counts, CellBender counts, count differences, and percent count differences per feature.

Usage

```
CellBender_Feature_Diff(
  seurat_object = NULL,
  raw_assay = NULL,
  cell_bender_assay = NULL,
  raw_mat = NULL,
  cell_bender_mat = NULL
)
```

Arguments

`seurat_object`	Seurat object name.
`raw_assay`	Name of the assay containing the raw count data.
`cell_bender_assay`	Name of the assay containing the CellBender count data.
`raw_mat`	Name of raw count matrix in environment if not using Seurat object.
`cell_bender_mat`	Name of CellBender count matrix in environment if not using Seurat object.

Value

A data.frame containing summed raw counts, CellBender counts, count difference, and percent difference in counts.

Examples

```
## Not run:
cb_stats <- CellBender_Feature_Diff(seurat_object - obj, raw_assay = "RAW",
cell_bender_assay = "RNA")

## End(Not run)
```

Cells.liger *Extract Cells from LIGER Object*

Description

Extract all cell barcodes from LIGER object

Usage

```
## S3 method for class 'liger'
Cells(x, by_dataset = FALSE, ...)
```

Arguments

x	LIGER object name.
by_dataset	logical, whether to return list with vector of cell barcodes for each dataset in LIGER object or to return single vector of cell barcodes across all datasets in object (default is FALSE; return vector of cells).
...	Arguments passed to other methods

Value

vector or list depending on by_dataset parameter

Examples

```
## Not run:
# return single vector of all cells
all_features <- Cells(x = object, by_dataset = FALSE)

# return list of vectors containing cells from each individual dataset in object
dataset_features <- Cells(x = object, by_dataset = TRUE)

## End(Not run)
```

Cells_by_Identities_LIGER *Extract Cells by identity*

Description

Extract all cell barcodes by identity from LIGER object

Usage

```
Cells_by_Identities_LIGER(liger_object, group.by = NULL, by_dataset = FALSE)
```

Arguments

liger_object	LIGER object name.
group.by	name of meta data column to use, default is current default clustering.
by_dataset	logical, whether to return list with entries for cell barcodes for each identity in group.by or to return list of lists (1 entry per dataset and each ident within the dataset) (default is FALSE; return list)

Value

list or list of lists depending on by_dataset parameter

Examples

```
## Not run:
# return single vector of all cells
cells_by_idents <- Cells_by_Identities_LIGER(liger_object = object, by_dataset = FALSE)

# return list of vectors containing cells from each individual dataset in object
cells_by_idents_by_dataset <- Cells_by_Identities_LIGER(liger_object = object, by_dataset = TRUE)

## End(Not run)
```

Cells_per_Sample *Cells per Sample*

Description

Get data.frame containing the number of cells per sample.

Usage

```
Cells_per_Sample(seurat_object, sample_col = NULL)
```

Arguments

seurat_object	Seurat object
sample_col	column name in meta.data that contains sample ID information. Default is NULL and will use "orig.ident column

Value

A data.frame

Examples

```
library(Seurat)
num_cells <- Cells_per_Sample(seurat_object = pbmc_small, sample_col = "orig.ident")
```

`Cell_Highlight_Plot` *Meta Highlight Plot*

Description

Create Plot with meta data variable of interest highlighted

Usage

```
Cell_Highlight_Plot(
  seurat_object,
  cells_highlight,
  highlight_color = NULL,
  background_color = "lightgray",
  pt.size = NULL,
  aspect_ratio = NULL,
  figure_plot = FALSE,
  raster = NULL,
  raster.dpi = c(512, 512),
  label = FALSE,
  split.by = NULL,
  split_seurat = FALSE,
  reduction = NULL,
  ggplot_default_colors = FALSE,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`cells_highlight`	Cell names to highlight in named list.
`highlight_color`	Color to highlight cells.
`background_color`	non-highlighted cell colors (default is "lightgray")..
`pt.size`	point size for both highlighted cluster and background.
`aspect_ratio`	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
`figure_plot`	logical. Whether to remove the axes and plot with legend on left of plot denoting axes labels. (Default is FALSE). Requires `split_seurat = TRUE`.
`raster`	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
`raster.dpi`	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).

label	Whether to label the highlighted meta data variable(s). Default is FALSE.
split.by	Variable in @meta.data to split the plot by.
split_seurat	logical. Whether or not to display split plots like Seurat (shared y axis) or as individual plots in layout. Default is FALSE.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
ggplot_default_colors	logical. If highlight_color = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
...	Extra parameters passed toDimPlot.

Value

A ggplot object

Examples

```
library(Seurat)

# Creating example non-overlapping vectors of cells
MS4A1 <- WhichCells(object = pbmc_small, expression = MS4A1 > 4)
GZMB <- WhichCells(object = pbmc_small, expression = GZMB > 4)

# Format as named list
cells <- list("MS4A1" = MS4A1,
              "GZMB" = GZMB)

Cell_Highlight_Plot(seurat_object = pbmc_small, cells_highlight = cells)
```

Change_Delim_All *Change all delimiters in cell name*

Description

Change all instances of delimiter in cell names from list of data.frames/matrices or single data.frame/matrix

Usage

```
Change_Delim_All(data, current_delim, new_delim)
```

Arguments

data	Either matrix/data.frame or list of matrices/data.frames with the cell barcodes in the column names.
current_delim	a single value of current delimiter.
new_delim	a single value of new delimiter desired.

Value

matrix or data.frame with new column names.

Examples

```
## Not run:
dge_matrix <- Change_Delim_All(data = dge_matrix, current_delim = ".", new_delim = "-")

## End(Not run)
```

Change_Delim_Prefix *Change barcode prefix delimiter*

Description

Change barcode prefix delimiter from list of data.frames/matrices or single data.frame/matrix

Usage

```
Change_Delim_Prefix(data, current_delim, new_delim)
```

Arguments

data	Either matrix/data.frame or list of matrices/data.frames with the cell barcodes in the column names.
current_delim	a single value of current delimiter.
new_delim	a single value of new delimiter desired.

Value

matrix or data.frame with new column names.

Examples

```
## Not run:
dge_matrix <- Change_Delim_Prefix(data = dge_matrix, current_delim = ".", new_delim = "-")

## End(Not run)
```

Change_Delim_Suffix *Change barcode suffix delimiter*

Description

Change barcode suffix delimiter from list of data.frames/matrices or single data.frame/matrix

Usage

```
Change_Delim_Suffix(data, current_delim, new_delim)
```

Arguments

data	Either matrix/data.frame or list of matrices/data.frames with the cell barcodes in the column names.
current_delim	a single value of current delimiter.
new_delim	a single value of new delimiter desired.

Value

matrix or data.frame with new column names.

Examples

```
## Not run:
dge_matrix <- Change_Delim_Suffix(data = dge_matrix, current_delim = ".", new_delim = "-")

## End(Not run)
```

CheckMatrix_scCustom *Check Matrix Validity*

Description

Native implementation of SeuratObjects CheckMatrix but with modified warning messages.

Usage

```
CheckMatrix_scCustom(
  object,
  checks = c("infinite", "logical", "integer", "na")
)
```

Arguments

object	A matrix
checks	Type of checks to perform, choose one or more from: • "infinite": Emit a warning if any value is infinite • "logical": Emit a warning if any value is a logical • "integer": Emit a warning if any value is *not* an integer • "na": Emit a warning if any value is an NA or NaN

Value

Emits warnings for each test and invisibly returns NULL

References

Re-implementing CheckMatrix only for sparse matrices with modified warning messages. Original function from SeuratObject https://github.com/satijalab/seurat-object/blob/9c0eda946e162d8595696e5280a6e R/utils.R#L625-L650 (License: MIT).

Examples

```
## Not run:
mat <- Read10X(...)
CheckMatrix_scCustom(object = mat)

## End(Not run)
```

Clustered_DotPlot *Clustered DotPlot*

Description

Clustered DotPlots using ComplexHeatmap

Usage

```
Clustered_DotPlot(
  seurat_object,
  features,
  label_selected_features = NULL,
  split.by = NULL,
  colors_use_exp = viridis_plasma_dark_high,
  exp_color_min = -2,
  exp_color_middle = NULL,
  exp_color_max = 2,
  exp_value_type = "scaled",
  print_exp_quantiles = FALSE,
```

```
  colors_use_idents = NULL,
  show_ident_colors = TRUE,
  x_lab_rotate = TRUE,
  plot_padding = NULL,
  flip = FALSE,
  k = 1,
  feature_km_repeats = 1000,
  ident_km_repeats = 1000,
  row_label_size = 8,
  row_label_fontface = "plain",
  grid_color = NULL,
  cluster_feature = TRUE,
  cluster_ident = TRUE,
  column_label_size = 8,
  legend_label_size = 10,
  legend_title_size = 10,
  legend_position = "right",
  legend_orientation = NULL,
  show_ident_legend = TRUE,
  show_row_names = TRUE,
  show_column_names = TRUE,
  column_names_side = "bottom",
  row_names_side = "right",
  raster = FALSE,
  plot_km_elbow = TRUE,
  elbow_kmax = NULL,
  assay = NULL,
  group.by = NULL,
  idents = NULL,
  show_parent_dend_line = TRUE,
  nan_error = FALSE,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  seed = 123
)
```

Arguments

`seurat_object`	Seurat object name.
`features`	Features to plot.
`label_selected_features`	a subset of `features` to only label some of the plotted features.
`split.by`	Variable in `@meta.data` to split the identities plotted by.
`colors_use_exp`	Color palette to use for plotting expression scale. Default is `viridis::plasma(n = 20, direction = -1)`.
`exp_color_min`	Minimum scaled average expression threshold (everything smaller will be set to this). Default is -2.

exp_color_middle
: What scaled expression value to use for the middle of the provided colors_use_exp. By default will be set to value in middle of exp_color_min and exp_color_max.

exp_color_max
: Minimum scaled average expression threshold (everything smaller will be set to this). Default is 2.

exp_value_type
: Whether to plot average normalized expression or scaled average normalized expression. Only valid when split.by is provided.

print_exp_quantiles
: Whether to print the quantiles of expression data in addition to plots. Default is FALSE. NOTE: These values will be altered by choices of exp_color_min and exp_color_min if there are values below or above those cutoffs, respectively.

colors_use_idents
: specify color palette to used for identity labels. By default if number of levels plotted is less than or equal to 36 it will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from DiscretePalette_scCustomize.

show_ident_colors
: logical, whether to show colors for idents on the column/rows of the plot (default is TRUE).

x_lab_rotate
: How to rotate column labels. By default set to TRUE which rotates labels 45 degrees. If set FALSE rotation is set to 0 degrees. Users can also supply custom angle for text rotation.

plot_padding
: if plot needs extra white space padding so no plot or labels are cutoff. The parameter accepts TRUE or numeric vector of length 4. If TRUE padding will be set to c(2, 10, 0 0) (bottom, left, top, right). Can also be customized further with numeric vector of length 4 specifying the amount of padding in millimeters. Default is NULL, no padding.

flip
: logical, whether to flip the axes of final plot. Default is FALSE; rows = features and columns = idents.

k
: Value to use for k-means clustering on features Sets (km) parameter in ComplexHeatmap::Heatmap(). From ComplexHeatmap::Heatmap(): Apply k-means clustering on rows. If the value is larger than 1, the heatmap will be split by rows according to the k-means clustering. For each row slice, hierarchical clustering is still applied with parameters above.

feature_km_repeats
: Number of k-means runs to get a consensus k-means clustering for features. Note if feature_km_repeats is set to value greater than one, the final number of groups might be smaller than row_km, but this might mean the original row_km is not a good choice. Default is 1000.

ident_km_repeats
: Number of k-means runs to get a consensus k-means clustering. Similar to feature_km_repeats. Default is 1000.

row_label_size
: Size of the feature labels. Provided to row_names_gp in Heatmap call.

row_label_fontface
: Fontface to use for row labels. Provided to row_names_gp in Heatmap call.

grid_color
: color to use for heatmap grid. Default is NULL which "removes" grid by using NA color.

`cluster_feature`
: logical, whether to cluster and reorder feature axis. Default is TRUE.

`cluster_ident` logical, whether to cluster and reorder identity axis. Default is TRUE.

`column_label_size`
: Size of the feature labels. Provided to `column_names_gp` in Heatmap call.

`legend_label_size`
: Size of the legend text labels. Provided to `labels_gp` in Heatmap legend call.

`legend_title_size`
: Size of the legend title text labels. Provided to `title_gp` in Heatmap legend call.

`legend_position`
: Location of the plot legend (default is "right").

`legend_orientation`
: Orientation of the legend (default is NULL).

`show_ident_legend`
: logical, whether to show the color legend for idents in plot (default is TRUE).

`show_row_names` logical, whether to show row names on plot (default is TRUE).

`show_column_names`
: logical, whether to show column names on plot (default is TRUE).

`column_names_side`
: Should the row names be on the "bottom" or "top" of plot. Default is "bottom".

`row_names_side` Should the row names be on the "left" or "right" of plot. Default is "right".

`raster` Logical, whether to render in raster format (faster plotting, smaller files). Default is FALSE.

`plot_km_elbow` Logical, whether or not to return the Sum Squared Error Elbow Plot for k-means clustering. Estimating elbow of this plot is one way to determine "optimal" value for k. Based on: https://stackoverflow.com/a/15376462/15568251.

`elbow_kmax` The maximum value of k to use for `plot_km_elbow`. Suggest setting larger value so the true shape of plot can be observed. Value must be 1 less than number of features provided. If NULL parameter will be set dependent on length of feature list up to `elbow_kmax = 20`.

`assay` Name of assay to use, defaults to the active assay.

`group.by` Group (color) cells in different ways (for example, orig.ident).

`idents` Which classes to include in the plot (default is all).

`show_parent_dend_line`
: Logical, Sets parameter of same name in `ComplexHeatmap::Heatmap()`. From `ComplexHeatmap::Heatmap()`: When heatmap is split, whether to add a dashed line to mark parent dendrogram and children dendrograms. Default is TRUE.

`nan_error` logical, default is FALSE. *ONLY* set this value to true if you get error related to NaN values when attempting to use plotting function. Plotting may be slightly slower if TRUE depending on number of features being plotted.

`ggplot_default_colors`
: logical. If `colors_use = NULL`, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.

color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
seed	Sets seed for reproducible plotting (ComplexHeatmap plot).

Value

A ComplexHeatmap or if plot_km_elbow = TRUE a list containing ggplot2 object and ComplexHeatmap.

Author(s)

Ming Tang (Original Code), Sam Marsh (Wrap single function, added/modified functionality)

References

https://divingintogeneticsandgenomics.rbind.io/post/clustered-dotplot-for-single-cell-rnaseq/

See Also

https://twitter.com/tangming2005

Examples

```
library(Seurat)
Clustered_DotPlot(seurat_object = pbmc_small, features = c("CD3E", "CD8", "GZMB", "MS4A1"))
```

Cluster_Highlight_Plot

Cluster Highlight Plot

Description

Create Plot with cluster of interest highlighted

Usage

```
Cluster_Highlight_Plot(
  seurat_object,
  cluster_name,
  highlight_color = NULL,
  background_color = "lightgray",
  pt.size = NULL,
  aspect_ratio = NULL,
  figure_plot = FALSE,
  raster = NULL,
  raster.dpi = c(512, 512),
```

```
  label = FALSE,
  split.by = NULL,
  split_seurat = FALSE,
  split_title_size = 15,
  num_columns = NULL,
  reduction = NULL,
  ggplot_default_colors = FALSE,
  ...
)
```

Arguments

seurat_object	Seurat object name.
cluster_name	Name(s) (or number(s)) identity of cluster to be highlighted.
highlight_color	Color(s) to highlight cells. The default is NULL and plot will use scCustomize_Palette().
background_color	non-highlighted cell colors.
pt.size	point size for both highlighted cluster and background.
aspect_ratio	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
figure_plot	logical. Whether to remove the axes and plot with legend on left of plot denoting axes labels. (Default is FALSE). Requires split_seurat = TRUE.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
label	Whether to label the highlighted cluster(s). Default is FALSE.
split.by	Feature to split plots by (i.e. "orig.ident").
split_seurat	logical. Whether or not to display split plots like Seurat (shared y axis) or as individual plots in layout. Default is FALSE.
split_title_size	size for plot title labels when using split.by.
num_columns	Number of columns in plot layout. Only valid if split.by != NULL.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
...	Extra parameters passed to DimPlot.

Value

A ggplot object

Examples

```
Cluster_Highlight_Plot(seurat_object = pbmc_small, cluster_name = "1", highlight_color = "gold",
background_color = "lightgray",  pt.size = 2)
```

`Cluster_Stats_All_Samples` *Calculate Cluster Stats*

Description

Calculates both overall and per sample cell number and percentages per cluster based on orig.ident.

Usage

```
Cluster_Stats_All_Samples(
  seurat_object,
  group_by_var = deprecated(),
  group.by = "orig.ident",
  order_by_freq = TRUE
)
```

Arguments

`seurat_object`	Seurat object name.
`group_by_var`	**[Deprecated]** soft-deprecated. See `group.by`.
`group.by`	meta data column to classify samples (default = "orig.ident").
`order_by_freq`	logical, whether the data.frame should be ordered by frequency of identity (default; TRUE), or by cluster/fector order (FALSE).

Value

A data.frame with rows in order of frequency or cluster order

Examples

```
## Not run:
stats <- Cluster_Stats_All_Samples(seurat_object = object, group.by = "orig.ident")

## End(Not run)
```

ColorBlind_Pal *Color Universal Design Short Palette*

Description

Shortcut ta a modified 8 color palette based on Color Universal Design (CUD) colorblindness friendly palette.

Usage

```
ColorBlind_Pal()
```

Value

modified/reordered color palette (8 colors) based on ditto-seq

References

palette is slightly modified version of the Color Universal Design (CUD) colorblindness friendly palette https://jfly.uni-koeln.de/color/.

Examples

```
cols <- ColorBlind_Pal()
PalettePlot(pal = cols)
```

Convert_Assay *Convert between Seurat Assay types*

Description

Will convert assays within a Seurat object between "Assay" and "Assay5" types.

Usage

```
Convert_Assay(seurat_object, assay = NULL, convert_to)
```

Arguments

seurat_object	Seurat object name.
assay	name(s) of assays to convert. Default is NULL and will check with users which assays they want to convert.
convert_to	value of what assay type to convert current assays to. #' • Accepted values for V3/4 are: "Assay", "assay", "V3", or "v3". • Accepted values for V5 are: "Assay5", "assay5", "V5", or "v5".

Examples

```
## Not run:
# Convert to V3/4 assay
obj <- Convert_Assay(seurat_object = obj, convert_to = "V3")

# Convert to 5 assay
obj <- Convert_Assay(seurat_object = obj, convert_to = "V5")

## End(Not run)
```

`Copy_From_GCP` *Copy folder from GCP bucket from R Console*

Description

Run command from R console without moving to terminal to copy folder from GCP bucket to local storage

Usage

```
Copy_From_GCP(folder_file_path, gcp_bucket_path)
```

Arguments

`folder_file_path`	folder to be copied to GCP bucket.
`gcp_bucket_path`	GCP bucket path to copy to files.

Value

No return value. Performs system copy from GCP bucket.

Examples

```
## Not run:
Copy_From_GCP(folder_file_path = "plots/", gcp_bucket_path = "gs://bucket_name_and_folder_path")

## End(Not run)
```

Copy_To_GCP *Copy folder to GCP bucket from R Console*

Description

Run command from R console without moving to terminal to copy folder to GCP bucket

Usage

```
Copy_To_GCP(folder_file_path, gcp_bucket_path)
```

Arguments

folder_file_path	folder to be copied to GCP bucket.
gcp_bucket_path	GCP bucket path to copy to files.

Value

No return value. Performs system copy to GCP bucket.

Examples

```
## Not run:
Copy_To_GCP(folder_file_path = "plots/", gcp_bucket_path = "gs://bucket_name_and_folder_path")

## End(Not run)
```

Create_10X_H5 *Create H5 from 10X Outputs*

Description

Creates HDF5 formatted output analogous to the outputs created by Cell Ranger and can be read into Seurat, LIGER, or SCE class object. Requires DropletUtils package from Bioconductor.

Usage

```
Create_10X_H5(
  raw_data_file_path,
  source_type = "10X",
  save_file_path,
  save_name
)
```

Arguments

raw_data_file_path	file path to raw data file(s).
source_type	type of source data (Default is "10X"). Alternatively can provide "Matrix" or "data.frame".
save_file_path	file path to directory to save file.
save_name	name prefix for output H5 file.

Value

A HDF5 format file that will be recognized as 10X Cell Ranger formatted file by Seurat or LIGER.

Examples

```
## Not run:
Create_10X_H5(raw_data_file_path = "file_path", save_file_path = "file_path2", save_name = "NAME")

## End(Not run)
```

Create_CellBender_Merged_Seurat *Create Seurat Object with Cell Bender and Raw data*

Description

Enables easy creation of Seurat object which contains both cell bender data and raw count data as separate assays within the object.

Usage

```
Create_CellBender_Merged_Seurat(
  raw_cell_bender_matrix = NULL,
  raw_counts_matrix = NULL,
  raw_assay_name = "RAW",
  min_cells = deprecated(),
  min_features = deprecated(),
  min.cells = 5,
  min.features = 200,
  ...
)
```

Arguments

raw_cell_bender_matrix	matrix file containing the cell bender correct counts.
raw_counts_matrix	matrix file contain the uncorrected Cell Ranger (or other) counts.
raw_assay_name	a key value to use specifying the name of assay. Default is "RAW".
min_cells	**[Deprecated]** soft-deprecated. See min.cells.
min_features	**[Deprecated]** soft-deprecated. See min.features.
min.cells	value to supply to min.cells parameter of CreateSeuratObject. Default is 5.
min.features	value to supply to min.features parameter of CreateSeuratObject. Default is 200.
...	Extra parameters passed to CreateSeuratObject.

Value

A Seurat Object contain both the Cell Bender corrected counts ("RNA" assay) and uncorrected counts ("RAW" assay; or other name specified to raw_assay_name).

Examples

```
## Not run:
seurat_obj <- Create_CellBender_Merged_Seurat(raw_cell_bender_matrix = cb_matrix,
raw_counts_matrix = cr_matrix)

## End(Not run)
```

Create_Cluster_Annotation_File

Create cluster annotation csv file

Description

create annotation file

Usage

```
Create_Cluster_Annotation_File(
  file_path = NULL,
  file_name = "cluster_annotation"
)
```

Arguments

file_path	path to directory to save file. Default is current working directory.
file_name	name to use for annotation file. Function automatically adds file type ".csv" suffix. Default is "cluster_annotation".

Value

No value returned. Creates .csv file.

Examples

```
## Not run:
Create_Cluster_Annotation_File(file_path = "cluster_annotation_folder_name")

## End(Not run)
```

`Dark2_Pal` *Dark2 Palette*

Description

Shortcut to Dark2 color palette from RColorBrewer (8 Colors)

Usage

```
Dark2_Pal()
```

Value

"Dark2" color palette (8 colors)

References

Dark2 palette from RColorBrewer being called through paletteer. See RColorBrewer for more info on palettes https://CRAN.R-project.org/package=RColorBrewer

Examples

```
cols <- Dark2_Pal()
PalettePlot(pal= cols)
```

Dataset_Size_LIGER *Check size of LIGER datasets*

Description

Returns size (number of cells) in each dataset within liger object along with other desired meta data.

Usage

```
Dataset_Size_LIGER(
  liger_object,
  meta_data_column = NULL,
  filter_by = NULL,
  print_filter = FALSE
)
```

Arguments

liger_object	LIGER object name.
meta_data_column	other meta data to include in returned data.frame.
filter_by	meta data column to filter data by. Will filter data to return only values for the largest dataset for each unique value in provided meta data column.
print_filter	logical, whether to print filtered results to console, default is FALSE.

Value

data.frame with dataset names, number of cells per dataset and if provided other meta data

Examples

```
## Not run:
# Return values for all datasets

## End(Not run)
```

`DimPlot_All_Samples` *DimPlot by Meta Data Column*

Description

Creates DimPlot layout containing all samples within Seurat Object from orig.ident column

Usage

```
DimPlot_All_Samples(
  seurat_object,
  meta_data_column = "orig.ident",
  colors_use = "black",
  pt.size = NULL,
  aspect_ratio = NULL,
  title_size = 15,
  num_columns = NULL,
  reduction = NULL,
  dims = c(1, 2),
  raster = NULL,
  raster.dpi = c(512, 512),
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`meta_data_column`	Meta data column to split plots by.
`colors_use`	single color to use for all plots or a vector of colors equal to the number of plots.
`pt.size`	Adjust point size for plotting.
`aspect_ratio`	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
`title_size`	size for plot title labels.
`num_columns`	number of columns in final layout plot.
`reduction`	Dimensionality Reduction to use (if NULL then defaults to Object default).
`dims`	Which dimensions to plot. Defaults to c(1,2) if not specified.
`raster`	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
`raster.dpi`	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
`...`	Extra parameters passed to `DimPlot`.

Value

A ggplot object

Examples

```
library(Seurat)

pbmc_small$sample_id <- sample(c("sample1", "sample2"), size = ncol(pbmc_small), replace = TRUE)

DimPlot_All_Samples(seurat_object = pbmc_small, meta_data_column = "sample_id", color = "black",
num_columns = 2)
```

DimPlot_LIGER *DimPlot LIGER Version*

Description

Standard and modified version of LIGER's plotByDatasetAndCluster

Usage

```
DimPlot_LIGER(
  liger_object,
  group_by = deprecated(),
  group.by = NULL,
  split_by = deprecated(),
  split.by = NULL,
  colors_use_cluster = NULL,
  colors_use_meta = NULL,
  pt_size = NULL,
  shuffle = TRUE,
  shuffle_seed = 1,
  reduction_label = "UMAP",
  reduction = NULL,
  aspect_ratio = NULL,
  label = TRUE,
  label_size = NA,
  label_repel = FALSE,
  label_box = FALSE,
  label_color = "black",
  combination = FALSE,
  raster = NULL,
  raster.dpi = c(512, 512),
  num_columns = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123
)
```

Arguments

liger_object	liger liger_object. Need to perform clustering before calling this function
group_by	**[Deprecated]** soft-deprecated. See group.by.
group.by	Variable to be plotted. If NULL will plot clusters from liger@clusters slot. If combination = TRUE will plot both clusters and meta data variable. If combination = TRUE will plot both clusters and meta data variable.
split_by	**[Deprecated]** soft-deprecated. See split.by.
split.by	Variable to split plots by.
colors_use_cluster	colors to use for plotting by clusters. By default if number of levels plotted is less than or equal to 36 will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from DiscretePalette_scCustomize.
colors_use_meta	colors to use for plotting by meta data (cell.data) variable. By default if number of levels plotted is less than or equal to 36 it will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from DiscretePalette_scCustomize.
pt_size	Adjust point size for plotting.
shuffle	logical. Whether to randomly shuffle the order of points. This can be useful for crowded plots if points of interest are being buried. (Default is TRUE).
shuffle_seed	Sets the seed if randomly shuffling the order of points.
reduction_label	What to label the x and y axes of resulting plots. LIGER does not store name of technique and therefore needs to be set manually. Default is "UMAP". (only valid for rliger < 2.0.0).
reduction	specify reduction to use when plotting. Default is current object default reduction (only valid for rliger v2.0.0 or greater).
aspect_ratio	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
label	logical. Whether or not to label the clusters. ONLY applies to plotting by cluster. Default is TRUE.
label_size	size of cluster labels.
label_repel	logical. Whether to repel cluster labels from each other if plotting by cluster (if group.by = NULL or group.by = "cluster). Default is FALSE.
label_box	logical. Whether to put a box around the label text (uses geom_text vs geom_label). Default is FALSE.
label_color	Color to use for cluster labels. Default is "black".
combination	logical, whether to return patchwork displaying both plots side by side. (Default is FALSE).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).

num_columns Number of columns in plot layout. Only valid if split.by != NULL.

ggplot_default_colors
logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.

color_seed random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Value

A ggplot/patchwork object

Examples

```
## Not run:
DimPlot_LIGER(liger_object = obj_name, reduction_label = "UMAP")

## End(Not run)
```

DimPlot_scCustom *DimPlot with modified default settings*

Description

Creates DimPlot with some of the settings modified from their Seurat defaults (colors_use, shuffle, label).

Usage

```
DimPlot_scCustom(
  seurat_object,
  colors_use = NULL,
  pt.size = NULL,
  reduction = NULL,
  group.by = NULL,
  split.by = NULL,
  split_seurat = FALSE,
  figure_plot = FALSE,
  aspect_ratio = NULL,
  add_prop_plot = FALSE,
  prop_plot_percent = FALSE,
  prop_plot_x_log = FALSE,
  prop_plot_label = FALSE,
  shuffle = TRUE,
  seed = 1,
  label = NULL,
  label.size = 4,
```

```
  label.color = "black",
  label.box = FALSE,
  dims = c(1, 2),
  repel = FALSE,
  raster = NULL,
  raster.dpi = c(512, 512),
  num_columns = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
colors_use	color palette to use for plotting. By default if number of levels plotted is less than or equal to 36 it will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from DiscretePalette_scCustomize.
pt.size	Adjust point size for plotting.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
split.by	Feature to split plots by (i.e. "orig.ident").
split_seurat	logical. Whether or not to display split plots like Seurat (shared y axis) or as individual plots in layout. Default is FALSE.
figure_plot	logical. Whether to remove the axes and plot with legend on left of plot denoting axes labels. (Default is FALSE). Requires split_seurat = TRUE.
aspect_ratio	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
add_prop_plot	logical, whether to add plot to returned layout with the number of cells per identity (or percent of cells per identity). Default is FALSE.
prop_plot_percent	logical, if add_prop_plot = TRUE this parameter controls whether proportion plot shows raw cell number or percent of cells per identity. Default is FALSE; plots raw cell number.
prop_plot_x_log	logical, if add_prop_plot = TRUE this parameter controls whether to change x axis to log10 scale (Default is FALSE).
prop_plot_label	logical, if add_prop_plot = TRUE this parameter controls whether to label the bars with total number of cells or percentages; Default is FALSE.
shuffle	logical. Whether to randomly shuffle the order of points. This can be useful for crowded plots if points of interest are being buried. (Default is TRUE).
seed	Sets the seed if randomly shuffling the order of points.

label	Whether to label the clusters. By default if group.by = NULL label = TRUE, and otherwise it is FALSE.
label.size	Sets size of labels.
label.color	Sets the color of the label text.
label.box	Whether to put a box around the label text (geom_text vs geom_label).
dims	Which dimensions to plot. Defaults to c(1,2) if not specified.
repel	Repel labels.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
num_columns	Number of columns in plot layout. Only valid if split.by != NULL.
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to DimPlot.

Value

A ggplot object

References

Many of the param names and descriptions are from Seurat to facilitate ease of use as this is simply a wrapper to alter some of the default parameters https://github.com/satijalab/seurat/blob/master/R/visualization.R (License: GPL-3). figure_plot parameter/code modified from code by Tim Stuart via twitter: https://twitter.com/timoast/status/1526237116035891200?s=20&t=foJOF81aPSjr1t7pk1cUPg.

Examples

```
library(Seurat)
DimPlot_scCustom(seurat_object = pbmc_small)
```

`DiscretePalette_scCustomize` *Discrete color palettes*

Description

Helper function to return a number of discrete color palettes.

Usage

```
DiscretePalette_scCustomize(
  num_colors,
  palette = NULL,
  shuffle_pal = FALSE,
  seed = 123
)
```

Arguments

`num_colors`	Number of colors to be generated.
`palette`	Options are "alphabet", "alphabet2", "glasbey", "polychrome", "stepped", "ditto_seq", "varibow".
`shuffle_pal`	randomly shuffle the outputted palette. Most useful for `varibow` palette as that is normally an ordered palette.
`seed`	random seed for the palette shuffle. Default = 123.

Value

A vector of colors

References

This function uses the paletteer package https://github.com/EmilHvitfeldt/paletteer to provide simplified access to color palettes from many different R package sources while minimizing scCustomize current and future dependencies.

The following packages & palettes are called by this function (see individual packages for palette references/citations):

1. pals (via paletteer) https://CRAN.R-project.org/package=pals
   - alphabet, alphabet2, glasbey, polychrome, and stepped.
2. dittoSeq https://bioconductor.org/packages/release/bioc/html/dittoSeq.html
   - dittoColors.
3. colorway https://github.com/hypercompetent/colorway
   - varibow

Function name and implementation modified from Seurat (License: GPL-3). https://github.com/satijalab/seurat

Examples

```
pal <- DiscretePalette_scCustomize(num_colors = 36, palette = "varibow")
PalettePlot(pal= pal)
```

`DotPlot_scCustom` *Customized DotPlot*

Description

Code for creating customized DotPlot

Usage

```
DotPlot_scCustom(
  seurat_object,
  features,
  group.by = NULL,
  colors_use = viridis_plasma_dark_high,
  remove_axis_titles = TRUE,
  x_lab_rotate = FALSE,
  y_lab_rotate = FALSE,
  facet_label_rotate = FALSE,
  flip_axes = FALSE,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`features`	Features to plot.
`group.by`	Name of metadata variable (column) to group cells by (for example, orig.ident); default is the current active.ident of the object.
`colors_use`	specify color palette to used. Default is viridis_plasma_dark_high.
`remove_axis_titles`	logical. Whether to remove the x and y axis titles. Default = TRUE.
`x_lab_rotate`	Rotate x-axis labels 45 degrees (Default is FALSE).
`y_lab_rotate`	Rotate x-axis labels 45 degrees (Default is FALSE).
`facet_label_rotate`	Rotate facet labels on grouped `DotPlots` by 45 degrees (Default is FALSE).
`flip_axes`	whether or not to flip and X and Y axes (Default is FALSE).
`...`	Extra parameters passed to `DotPlot`.

Value

A ggplot object

Examples

```
library(Seurat)
DotPlot_scCustom(seurat_object = pbmc_small, features = c("CD3E", "CD8", "GZMB", "MS4A1"))
```

Embeddings.liger *Extract matrix of embeddings*

Description

Extract matrix containing iNMF or dimensionality reduction embeddings.

Usage

```
## S3 method for class 'liger'
Embeddings(object, reduction = NULL, iNMF = FALSE, check_only = FALSE, ...)
```

Arguments

object	LIGER object name.
reduction	name of dimensionality reduction to pull
iNMF	logical, whether to extract iNMF h.norm matrix instead of dimensionality reduction embeddings.
check_only	logical, return TRUE if valid reduction is present.
...	Arguments passed to other methods

Value

matrix

Examples

```
## Not run:
# Extract embedding matrix for current dimensionality reduction
UMAP_coord <- Embeddings(object = liger_object)

# Extract iNMF h.norm matrix
iNMF_mat <- Embeddings(object = liger_object, reduction = "iNMF")

## End(Not run)
```

ensembl_exAM_list *Immediate Early Gene (IEG) gene lists*

Description

Ensembl IDs for immediate early genes (Ensembl version 112; 4/29/2024)

Usage

```
ensembl_exAM_list
```

Format

A list of three vectors

Mus_musculus_exAM_union_ensembl Ensembl ID for exAM genes from source publication (see below)

Homo_sapiens_exAM_union_ensembl Human Ensembl ID for exAM genes for homologous genes from mouse gene list

Homo_sapiens_exAM_micro_ensembl Human Ensembl ID for exAM genes for human microglia list

Source

Gene list is from: SI Table 22 Marsh et al., 2022 (Nature Neuroscience) from doi:10.1038/s41593-022010228. See data-raw directory for scripts used to create gene list.

ensembl_hemo_id *Ensembl Hemo IDs*

Description

A list of ensembl ids for hemoglobin genes (Ensembl version 112; 4/29/2024)

Usage

```
ensembl_hemo_id
```

Format

A list of six vectors

Mus_musculus_hemo_ensembl Ensembl IDs for mouse hemoglobin genes

Homo_sapiens_hemo_ensembl Ensembl IDs for human hemoglobin genes

Danio_rerio_hemo_ensembl Ensembl IDs for zebrafish hemoglobin genes

Rattus_norvegicus_hemo_ensembl Ensembl IDs for rat hemoglobin genes

Drosophila_melanogaster_hemo_ensembl Ensembl IDs for fly hemoglobin genes

Macaca_mulatta_hemo_ensembl Ensembl IDs for macaque hemoglobin genes

Gallus_gallus_ribo_ensembl Ensembl IDs for chicken hemoglobin genes

Source

See data-raw directory for scripts used to create gene list.

---

ensembl_ieg_list *Immediate Early Gene (IEG) gene lists*

Description

Ensembl IDs for immediate early genes (Ensembl version 112; 4/29/2024)

Usage

```
ensembl_ieg_list
```

Format

A list of seven vectors

Mus_musculus_IEGs Ensembl IDs for IEGs from source publication (see below)

Homo_sapiens_IEGs Ensembl IDs for homologous genes from mouse gene list

Source

Mouse gene list is from: SI Table 4 from doi:10.1016/j.neuron.2017.09.026. Human gene list was compiled by first creating homologous gene list using biomaRt and then adding some manually curated homologs according to HGNC. See data-raw directory for scripts used to create gene list.

ensembl_lncRNA_id *Ensembl lncRNA IDs*

Description

A list of ensembl ids for lncRNA genes (Ensembl version 113; 04/08/2025)

Usage

```
ensembl_lncRNA_id
```

Format

A list of seven vectors

Mus_musculus_lncRNA_ensembl Ensembl IDs for mouse lncRNA genes
Homo_sapiens_lncRNA_ensembl Ensembl IDs for human lncRNA genes
Callithrix_jacchus_lncRNA_ensembl Ensembl IDs for marmoset lncRNA genes
Danio_rerio_lncRNA_ensembl Ensembl IDs for zebrafish lncRNA genes
Rattus_norvegicus_lncRNA_ensembl Ensembl IDs for rat lncRNA genes
Macaca_mulatta_lncRNA_ensembl Ensembl IDs for macaque lncRNA genes
Gallus_gallus_lncRNA_ensembl Ensembl IDs for chicken lncRNA genes

Source

See data-raw directory for scripts used to create gene list.

ensembl_malat1_list *MALAT1 gene lists*

Description

Ensembl IDs for MALAT1 (Ensembl version 112; 4/29/2024)

Usage

```
ensembl_malat1_list
```

Format

A list of seven vectors

Mus_musculus_MALAT1_ensembl Ensembl ID for mouse Malat1
Homo_sapiens_MALAT1_ensembl Ensembl ID for human MALAT1

Source

See data-raw directory for scripts used to create gene list.

ensembl_mito_id *Ensembl Mito IDs*

Description

A list of ensembl ids for mitochondrial genes (Ensembl version 112; 4/29/2024)

Usage

```
ensembl_mito_id
```

Format

A list of six vectors

Mus_musculus_mito_ensembl Ensembl IDs for mouse mitochondrial genes

Homo_sapiens_mito_ensembl Ensembl IDs for human mitochondrial genes

Danio_rerio_mito_ensembl Ensembl IDs for zebrafish mitochondrial genes

Rattus_norvegicus_mito_ensembl Ensembl IDs for rat mitochondrial genes

Drosophila_melanogaster_mito_ensembl Ensembl IDs for fly mitochondrial genes

Macaca_mulatta_mito_ensembl Ensembl IDs for macaque mitochondrial genes

Gallus_gallus_mito_ensembl Ensembl IDs for chicken mitochondrial genes

Source

See data-raw directory for scripts used to create gene list.

ensembl_ribo_id *Ensembl Ribo IDs*

Description

A list of ensembl ids for ribosomal genes (Ensembl version 112; 4/29/2024)

Usage

```
ensembl_ribo_id
```

Format

A list of eight vectors

Mus_musculus_ribo_ensembl Ensembl IDs for mouse ribosomal genes

Homo_sapiens_ribo_ensembl Ensembl IDs for human ribosomal genes

Callithrix_jacchus_ribo_ensembl Ensembl IDs for marmoset ribosomal genes

Danio_rerio_ribo_ensembl Ensembl IDs for zebrafish ribosomal genes

Rattus_norvegicus_ribo_ensembl Ensembl IDs for rat ribosomal genes

Drosophila_melanogaster_ribo_ensembl Ensembl IDs for fly ribosomal genes

Macaca_mulatta_ribo_ensembl Ensembl IDs for macaque ribosomal genes

Gallus_gallus_ribo_ensembl Ensembl IDs for chicken ribosomal genes

Source

See data-raw directory for scripts used to create gene list.

`exAM_gene_list` *exAM gene lists*

Description

Gene symbols for exAM genes

Usage

```
exAM_gene_list
```

Format

A list of three vectors

Mus_musculus_exAM_union Gene symbols for exAM genes from source publication (see below)

Homo_sapiens_exAM_union Human gene symbols for homologous genes from mouse gene list

Homo_sapiens_exAM_micro Human gene symbols for human microglia list

Source

Gene list is from: SI Table 22 Marsh et al., 2022 (Nature Neuroscience) from doi:10.1038/s41593-022010228. See data-raw directory for scripts used to create gene list.

exAM_Scoring *Add exAM Gene List Module Scores*

Description

Adds module scores from exAM genes from mouse and human.

Usage

```
exAM_Scoring(
  seurat_object,
  species,
  exam_module_name = NULL,
  method = "Seurat",
  ensembl_ids = FALSE,
  assay = NULL,
  overwrite = FALSE,
  exclude_unfound = FALSE,
  seed = 1
)
```

Arguments

seurat_object	object name.
species	Species of origin for given Seurat Object. Only accepted species are: mouse, human (name or abbreviation).
exam_module_name	name to use for the new meta.data column containing module scores.
method	method to use for module scoring, currently only "Seurat" is supported but more to be added. .
ensembl_ids	logical, whether feature names in the object are gene names or ensembl IDs (default is FALSE; set TRUE if feature names are ensembl IDs).
assay	Assay to use (default is the current object default assay).
overwrite	Logical. Whether to overwrite existing meta.data columns. Default is FALSE meaning that function will abort if columns with the name provided to exam_module_name is present in meta.data slot.
exclude_unfound	logical, whether to exclude features not present in current object (default is FALSE).
seed	seed for reproducibility (default is 1).

Value

Seurat object

References

Gene list is from: SI Table 22 Marsh et al., 2022 (Nature Neuroscience) from doi:10.1038/s41593-022010228. See data-raw directory for scripts used to create gene list.

Examples

```
## Not run:
# Seurat
seurat_object <- exAM_Scoring(seurat_object = seurat_object, species = "human")

## End(Not run)
```

`Extract_Modality` *Extract multi-modal data into list by modality*

Description

Reorganize multi-modal data after import with `Read10X()` or scCustomize read functions. Organizes sub-lists by data modality instead of by sample.

Usage

```
Extract_Modality(matrix_list)
```

Arguments

`matrix_list`	list of matrices to split by modality

Value

list of lists, with one sublist per data modality. Sub-list contain 1 matrix entry per sample

Examples

```
## Not run:
multi_mat <- Read10X(...)
new_multi_mat <- Extract_Modality(matrix_list = multi_mat)

## End(Not run)
```

`Extract_Sample_Meta` *Extract sample level meta.data*

Description

Returns a by identity meta.data data.frame with one row per sample. Useful for downstream quick view of sample breakdown, meta data table creation, and/or use in pseudobulk analysis

Usage

```
Extract_Sample_Meta(
  object,
  sample_name = "orig.ident",
  variables_include = NULL,
  variables_exclude = NULL,
  include_all = FALSE
)
```

Arguments

`object`	Seurat or LIGER object
`sample_name`	meta.data column to use as sample. Output data.frame will contain one row per level or unique value in this variable.
`variables_include`	`@meta.data` columns to keep in final data.frame. All other columns will be discarded. Default is NULL.
`variables_exclude`	columns to discard in final data.frame. Many cell level columns are irrelevant at the sample level (e.g., nFeature_RNA, percent_mito). • Default parameter value is `NULL` but internally will set to discard nFeature_ASSAY(s), nCount_ASSAY(s), percent_mito, percent_ribo, percent_mito_ribo, and log10GenesPerUMI. • If sample level median values are desired for these type of variables the output of this function can be joined with output of `Median_Stats`. • Set parameter to `include_all = TRUE` to prevent any columns from being excluded.
`include_all`	logical, whether or not to include all object meta data columns in output data.frame. Default is FALSE.

Value

Returns a data.frame with one row per `sample_name`.

Examples

```
library(Seurat)
pbmc_small[["batch"]] <- sample(c("batch1", "batch2"), size = ncol(pbmc_small), replace = TRUE)

sample_meta <- Extract_Sample_Meta(object = pbmc_small, sample_name = "orig.ident")

# Only return specific columns from meta data (orig.ident and batch)
sample_meta2 <- Extract_Sample_Meta(object = pbmc_small, sample_name = "orig.ident",
variables_include = "batch")

# Return all columns from meta data
sample_meta3 <- Extract_Sample_Meta(object = pbmc_small, sample_name = "orig.ident",
include_all = TRUE)
```

Extract_Top_Markers *Extract Top N Marker Genes*

Description

Extract vector gene list (or named gene vector) from data.frame results of `FindAllMarkers` or similar analysis.

Usage

```
Extract_Top_Markers(
  marker_dataframe,
  num_features = 10,
  num_genes = deprecated(),
  group_by = deprecated(),
  group.by = "cluster",
  rank_by = "avg_log2FC",
  gene_column = "gene",
  gene_rownames_to_column = FALSE,
  data_frame = FALSE,
  named_vector = TRUE,
  make_unique = FALSE
)
```

Arguments

`marker_dataframe`	data.frame output from `FindAllMarkers` or similar analysis.
`num_features`	number of features per group (e.g., cluster) to include in output list.
`num_genes`	**[Deprecated]** soft-deprecated. See `num_features`.
`group_by`	**[Deprecated]** soft-deprecated. See `group.by`.

group.by column name of marker_dataframe to group data by. Default is "cluster" based on FindAllMarkers.

rank_by column name of marker_dataframe to rank data by when selecting num_genes per group.by. Default is "avg_log2FC" based on FindAllMarkers.

gene_column column name of marker_dataframe that contains the gene IDs. Default is "gene" based on FindAllMarkers.

gene_rownames_to_column logical. Whether gene IDs are stored in rownames and should be moved to column. Default is FALSE.

data_frame Logical, whether or not to return filtered data.frame of the original markers_dataframe or to return a vector of gene IDs. Default is FALSE.

named_vector Logical, whether or not to name the vector of gene names that is returned by the function. If TRUE will name the vector using the column provided to group.by. Default is TRUE.

make_unique Logical, whether an unnamed vector should return only unique values. Default is FALSE. Not applicable when data_frame = TRUE or named_vector = TRUE.

Value

filtered data.frame, vector, or named vector containing gene IDs.

Examples

```
## Not run:
top10_genes <- Extract_Top_Markers(marker_dataframe = markers_results, num_genes = 10,
group.by = "cluster", rank_by = "avg_log2FC")

## End(Not run)
```

Factor_Cor_Plot *Factor Correlation Plot*

Description

Plot positive correlations between gene loadings across W factor matrix in liger or feature loadings in reduction slot of Seurat object. Any negative correlations are set to NA and NA values set to bottom color of color gradient.

Usage

```
Factor_Cor_Plot(
  object,
  reduction = NULL,
  colors_use = NULL,
  label = FALSE,
```

```
  label_threshold = 0.5,
  label_size = 5,
  plot_title = NULL,
  plot_type = "full",
  positive_only = FALSE,
  x_lab_rotate = TRUE,
  cluster = TRUE,
  cluster_rect = FALSE,
  cluster_rect_num = NULL,
  cluster_rect_col = NULL
)
```

Arguments

object	liger or Seurat object.
reduction	name of dimensionality reduction containing NMF loadings, Seurat only.
colors_use	Color palette to use for correlation values. Default is RColorBrewer::RdBu if positive_only = FALSE. If positive_only = TRUE the default is viridis. Users can also supply vector of 3 colors (low, mid, high).
label	logical, whether to add correlation values to plot result.
label_threshold	threshold for adding correlation values if label = TRUE. Default is 0.5.
label_size	size of correlation labels
plot_title	Plot title.
plot_type	Controls plotting full matrix, or just the upper or lower triangles. Accepted values are: "full" (default), "upper", or "lower".
positive_only	logical, whether to limit the plotted values to only positive correlations (negative values set to 0); default is FALSE.
x_lab_rotate	logical, whether to rotate the axes labels on the x-axis. Default is TRUE.
cluster	logical, whether to cluster the plot using hclust (default TRUE). If FALSE factors are listed in numerical order.
cluster_rect	logical, whether to add rectangles around the clustered areas on plot, default is FALSE.
cluster_rect_num	number of rectangles to add to the plot, default NULL.
cluster_rect_col	color to use for rectangles, default MULL (will set color automatically).

Value

A ggplot object

Examples

```
## Not run:
Factor_Cor_Plot(object = obj)

## End(Not run)
```

FeaturePlot_DualAssay *Customize FeaturePlot of two assays*

Description

Create Custom FeaturePlots and preserve scale (no binning) from same features in two assays simultaneously. Intended for plotting same modality present in two assays.

Usage

```
FeaturePlot_DualAssay(
  seurat_object,
  features,
  assay1 = "RAW",
  assay2 = "RNA",
  colors_use = viridis_plasma_dark_high,
  colors_use_assay2 = NULL,
  na_color = "lightgray",
  order = TRUE,
  pt.size = NULL,
  aspect_ratio = NULL,
  reduction = NULL,
  na_cutoff = 1e-09,
  raster = NULL,
  raster.dpi = c(512, 512),
  layer = "data",
  num_columns = NULL,
  alpha_exp = NULL,
  alpha_na_exp = NULL,
  ...
)
```

Arguments

seurat_object	Seurat object name.
features	Feature(s) to plot.
assay1	name of assay one. Default is "RAW" as featured in Create_CellBender_Merged_Seurat
assay2	name of assay two Default is "RNA" as featured in Create_CellBender_Merged_Seurat
colors_use	list of colors or color palette to use.

colors_use_assay2	optional, a second color palette to use for the second assay.
na_color	color to use for points below lower limit.
order	whether to move positive cells to the top (default = TRUE).
pt.size	Adjust point size for plotting.
aspect_ratio	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
na_cutoff	Value to use as minimum expression cutoff. To set no cutoff set to NA.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
layer	Which layer to pull expression data from? Default is "data".
num_columns	Number of columns in plot layout. If number of features > 1 then num_columns dictates the number of columns in overall layout (num_columns = 1 means stacked layout & num_columns = 2 means adjacent layout).
alpha_exp	new alpha level to apply to expressing cell color palette (colors_use). Must be value between 0-1.
alpha_na_exp	new alpha level to apply to non-expressing cell color palette (na_color). Must be value between 0-1.
...	Extra parameters passed to FeaturePlot.

Value

A ggplot object

Examples

```
## Not run:
FeaturePlot_DualAssay(seurat_object = object, features = "Cx3cr1", assay1 = "RAW", assay2 = "RNA",
colors_use = viridis_plasma_dark_high, na_color = "lightgray")

## End(Not run)
```

FeaturePlot_scCustom *Customize FeaturePlot*

Description

Create Custom FeaturePlots and preserve scale (no binning)

Usage

```
FeaturePlot_scCustom(
  seurat_object,
  features,
  colors_use = viridis_plasma_dark_high,
  na_color = "lightgray",
  order = TRUE,
  pt.size = NULL,
  reduction = NULL,
  na_cutoff = 1e-09,
  raster = NULL,
  raster.dpi = c(512, 512),
  split.by = NULL,
  split_collect = NULL,
  aspect_ratio = NULL,
  figure_plot = FALSE,
  num_columns = NULL,
  layer = "data",
  alpha_exp = NULL,
  alpha_na_exp = NULL,
  label = FALSE,
  label_feature_yaxis = FALSE,
  max.cutoff = NA,
  min.cutoff = NA,
  combine = TRUE,
  ...
)
```

Arguments

seurat_object	Seurat object name.
features	Feature(s) to plot.
colors_use	list of colors or color palette to use.
na_color	color to use for points below lower limit.
order	whether to move positive cells to the top (default = TRUE).
pt.size	Adjust point size for plotting.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
na_cutoff	Value to use as minimum expression cutoff. This will be lowest value plotted use palette provided to colors_use. Leave as default value to plot only positive non-zero values using color scale and zero/negative values as NA. To plot all values using color palette set to NA.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).

split.by	Variable in @meta.data to split the plot by.
split_collect	logical, whether to collect the legends/guides when plotting with split.by. Default is TRUE if one value is provided to features otherwise is set to FALSE.
aspect_ratio	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
figure_plot	logical. Whether to remove the axes and plot with legend on left of plot denoting axes labels. (Default is FALSE). Requires split_seurat = TRUE.
num_columns	Number of columns in plot layout.
layer	Which layer to pull expression data from? Default is "data".
alpha_exp	new alpha level to apply to expressing cell color palette (colors_use). Must be value between 0-1.
alpha_na_exp	new alpha level to apply to non-expressing cell color palette (na_color). Must be value between 0-1.
label	logical, whether to label the clusters. Default is FALSE.
label_feature_yaxis	logical, whether to place feature labels on secondary y-axis as opposed to above legend key. Default is FALSE. When setting label_feature_yaxis = TRUE the number of columns in plot output will automatically be set to the number of levels in split.by'.
min.cutoff, max.cutoff	Vector of minimum and maximum cutoff values for each feature, may specify quantile in the form of 'q##' where '##' is the quantile (eg, 'q1', 'q10').
combine	Combine plots into a single patchworked ggplot object. If FALSE, return a list of ggplot objects.
...	Extra parameters passed to FeaturePlot.

Value

A ggplot object

Examples

```
library(Seurat)
FeaturePlot_scCustom(seurat_object = pbmc_small, features = "CD3E",
colors_use = viridis_plasma_dark_high, na_color = "lightgray")
```

Features.liger *Extract Features from LIGER Object*

Description

Extract all unique features from LIGER object

Usage

```
## S3 method for class 'liger'
Features(x, by_dataset = FALSE, ...)
```

Arguments

x	LIGER object name.
by_dataset	logical, whether to return list with vector of features for each dataset in LIGER object or to return single vector of unique features across all datasets in object (default is FALSE; return vector of unique features)
...	Arguments passed to other methods

Value

vector or list depending on by_dataset parameter

Examples

```
## Not run:
# return single vector of all unique features
all_features <- Features(x = object, by_dataset = FALSE)

# return list of vectors containing features from each individual dataset in object
dataset_features <- Features(x = object, by_dataset = TRUE)

## End(Not run)
```

FeatureScatter_scCustom

Modified version of FeatureScatter

Description

Create customized FeatureScatter plots with scCustomize defaults.

Usage

```
FeatureScatter_scCustom(
  seurat_object,
  feature1 = NULL,
  feature2 = NULL,
  cells = NULL,
  colors_use = NULL,
  pt.size = NULL,
  group.by = NULL,
  split.by = NULL,
```

```
  split_seurat = FALSE,
  shuffle = TRUE,
  aspect_ratio = NULL,
  title_size = 15,
  plot.cor = TRUE,
  num_columns = NULL,
  raster = NULL,
  raster.dpi = c(512, 512),
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
feature1	First feature to plot.
feature2	Second feature to plot.
cells	Cells to include on the scatter plot.
colors_use	color for the points on plot.
pt.size	Adjust point size for plotting.
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident). Default is active ident.
split.by	Feature to split plots by (i.e. "orig.ident").
split_seurat	logical. Whether or not to display split plots like Seurat (shared y axis) or as individual plots in layout. Default is FALSE.
shuffle	logical, whether to randomly shuffle the order of points. This can be useful for crowded plots if points of interest are being buried. Default is TRUE.
aspect_ratio	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
title_size	size for plot title labels. Does NOT apply if split_seurat = TRUE.
plot.cor	Display correlation in plot subtitle (or title if split_seurat = TRUE).
num_columns	number of columns in final layout plot.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to FeatureScatter.

Value

A ggplot object

Examples

```
library(Seurat)

pbmc_small$sample_id <- sample(c("sample1", "sample2"), size = ncol(pbmc_small), replace = TRUE)

FeatureScatter_scCustom(seurat_object = pbmc_small, feature1 = "nCount_RNA",
feature2 = "nFeature_RNA", split.by = "sample_id")
```

Feature_Present *Check if genes/features are present*

Description

Check if genes are present in object and return vector of found genes. Return warning messages for genes not found.

Usage

```
Feature_Present(
  data,
  features,
  case_check = TRUE,
  case_check_msg = TRUE,
  print_msg = TRUE,
  omit_warn = TRUE,
  return_none = FALSE,
  seurat_assay = NULL
)
```

Arguments

data	Name of input data. Currently only data of classes: Seurat, liger, data.frame, dgCMatrix, dgTMatrix, tibble are accepted. Gene_IDs must be present in row-names of the data.
features	vector of features to check.
case_check	logical. Whether or not to check if features are found if the case is changed from the input list (Sentence case to Upper and vice versa). Default is TRUE.
case_check_msg	logical. Whether to print message to console if alternate case features are found in addition to inclusion in returned list. Default is TRUE.
print_msg	logical. Whether message should be printed if all features are found. Default is TRUE.

omit_warn	logical. Whether to print message about features that are not found in current object. Default is TRUE.
return_none	logical. Whether list of found vs. bad features should still be returned if no features are found. Default is FALSE.
seurat_assay	Name of assay to pull feature names from if data is Seurat Object. Default is NULL which will check against features from all assays present.

Value

A list of length 3 containing 1) found features, 2) not found features, 3) features found if case was modified.

Examples

```
## Not run:
features <- Feature_Present(data = obj_name, features = DEG_list, print_msg = TRUE,
case_check = TRUE)
found_features <- features[[1]]

## End(Not run)
```

Fetch_Meta *Get meta data from object*

Description

Quick function to properly pull meta.data from objects.

Usage

```
Fetch_Meta(object, ...)

## S3 method for class 'liger'
Fetch_Meta(object, ...)

## S3 method for class 'Seurat'
Fetch_Meta(object, ...)
```

Arguments

object	Object of class Seurat or liger.
...	Arguments passed to other methods

Value

A data.frame containing cell-level meta data

Examples

```
library(Seurat)
meta_data <- Fetch_Meta(object = pbmc_small)
head(meta_data, 5)
```

`Find_Factor_Cor` *Find Factor Correlations*

Description

Calculate correlations between gene loadings for all factors in liger or Seurat object.

Usage

```
Find_Factor_Cor(object, reduction = NULL)
```

Arguments

`object`	LIGER/Seurat object name.
`reduction`	reduction name to pull loadings for. Only valid if supplying a Seurat object.

Value

correlation matrix

Examples

```
## Not run:
factor_correlations <- Find_Factor_Cor(object = object)

## End(Not run)
```

`Get_Reference_LIGER` *Get Reference Dataset*

Description

Function to select reference dataset to use in liger based on meta data information

Usage

```
Get_Reference_LIGER(liger_object, meta_data_column, value)
```

Arguments

liger_object	LIGER object name.
meta_data_column	meta data column to use for selecting largest dataset.
value	value from column meta_data_column to use for selecting largest dataset.

Value

dataset name as character

Examples

```
## Not run:
# standalone use
ref_dataset <- Get_Reference_LIGER(liger_object = object, meta_data_column = "Treatment",
value = "Ctrl")

# use within `quantileNorm`
object <- quantileNorm(object = object, reference = Get_Reference_LIGER(liger_object = object,
meta_data_column = "Treatment", value = "Ctrl"))

## End(Not run)
```

Hue_Pal *Hue_Pal*

Description

Shortcut to hue_pal to return to ggplot2 defaults if user desires, from scales package.

Usage

```
Hue_Pal(num_colors)
```

Arguments

num_colors	number of colors to return in palette.

Value

hue color palette (as many colors as desired)

Examples

```
cols <- Hue_Pal(num_colors = 8)
PalettePlot(pal= cols)
```

Idents.liger *Extract or set default identities from object*

Description

Extract default identities from object in factor form.

Usage

```
## S3 method for class 'liger'
Idents(object, ...)

## S3 replacement method for class 'liger'
Idents(object, ...) <- value
```

Arguments

object	LIGER object name.
...	Arguments passed to other methods
value	name of column in cellMeta slot to set as new default cluster/ident

Value

factor

object

Note

Use of Idents<- is only for setting new default ident/cluster from column already present in cellMeta. To add new column with new cluster values to cellMeta and set as default see Rename_Clusters.

Examples

```
## Not run:
# Extract idents
object_idents <- Idents(object = liger_object)

## End(Not run)

## Not run:
# Set idents
Idents(object = liger_object) <- "new_annotation"

## End(Not run)
```

ieg_gene_list *Immediate Early Gene (IEG) gene lists*

Description

Gene symbols for immediate early genes

Usage

```
ieg_gene_list
```

Format

A list of seven vectors

Mus_musculus_IEGs Gene symbols for IEGs from source publication (see below)

Homo_sapiens_IEGs Human gene symbols for homologous genes from mouse gene list

Source

Mouse gene list is from: SI Table 4 from doi:10.1016/j.neuron.2017.09.026. Human gene list was compiled by first creating homologous gene list using biomaRt and then adding some manually curated homologs according to HGNC. See data-raw directory for scripts used to create gene list.

Iterate_Barcode_Rank_Plot *Iterative Barcode Rank Plots*

Description

Read data, calculate `DropletUtils::barcodeRanks`, create barcode rank plots, and outout single PDF output.

Usage

```
Iterate_Barcode_Rank_Plot(
  dir_path_h5,
  multi_directory = TRUE,
  h5_filename = "raw_feature_bc_matrix.h5",
  cellranger_multi = FALSE,
  parallel = FALSE,
  num_cores = NULL,
  file_path = NULL,
  file_name = NULL,
  pt.size = 6,
  raster_dpi = c(1024, 1024),
```

```
  plateau = NULL,
  ...
)
```

Arguments

dir_path_h5	path to parent directory (if multi_directory = TRUE) or directory containing all h5 files (if multi_directory = FALSE).
multi_directory	logical, whether or not all h5 files are in their own subdirectories or in a single directory (default is TRUE; each in own subdirectory (e.g. output from Cell Ranger)).
h5_filename	Either the file name of h5 file (if multi_directory = TRUE) or the shared suffix (if multi_directory = FALSE)
cellranger_multi	logical, whether the outputs to be read are from Cell Ranger multi as opposed to Cell Ranger count (default is FALSE). Only valid if multi_directory = FALSE.
parallel	logical, should files be read in parallel (default is FALSE).
num_cores	Number of cores to use in parallel if parallel = TRUE.
file_path	file path to use for saving PDF output.
file_name	Name of PDF output file.
pt.size	point size for plotting, default is 6.
raster_dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(1024, 1024).
plateau	numerical values at which to add vertical line designating estimated empty droplet plateau (default is NULL). Must be vector equal in length to number of samples.
...	Additional parameters passed to Read10X_h5_Multi_Directory or Read10X_h5_GEO.

Value

pdf document

Examples

```
## Not run:
Iterate_Barcode_Rank_Plot(dir_path_h5 = "H5_PATH/", multi_directory = TRUE,
h5_filename = "raw_feature_bc_matrix", parallel = TRUE, num_cores = 12, file_path = "OUTPUT_PATH",
file_name = "Barcode_Rank_Plots")

## End(Not run)
```

`Iterate_Cluster_Highlight_Plot` *Iterate Cluster Highlight Plot*

Description

Iterate the create plots with cluster of interest highlighted across all cluster (active.idents) in given Seurat Object

Usage

```
Iterate_Cluster_Highlight_Plot(
  seurat_object,
  highlight_color = "dodgerblue",
  background_color = "lightgray",
  pt.size = NULL,
  reduction = NULL,
  file_path = NULL,
  file_name = NULL,
  file_type = NULL,
  single_pdf = FALSE,
  output_width = NULL,
  output_height = NULL,
  dpi = 600,
  raster = NULL,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`highlight_color`	Color to highlight cells (default "navy"). Can provide either single color to use for all clusters/plots or a vector of colors equal to the number of clusters to use (in order) for the clusters/plots.
`background_color`	non-highlighted cell colors.
`pt.size`	point size for both highlighted cluster and background.
`reduction`	Dimensionality Reduction to use (if NULL then defaults to Object default).
`file_path`	directory file path and/or file name prefix. Defaults to current wd.
`file_name`	name suffix to append after sample name.
`file_type`	File type to save output as. Must be one of following: ".pdf", ".png", ".tiff", ".jpeg", or ".svg".
`single_pdf`	saves all plots to single PDF file (default = FALSE). ‘file_type‘ must be .pdf.

output_width	the width (in inches) for output page size. Default is NULL.
output_height	the height (in inches) for output page size. Default is NULL.
dpi	dpi for image saving.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
...	Extra parameters passed toDimPlot.

Value

Saved plots

Examples

```
## Not run:
Iterate_Cluster_Highlight_Plot(seurat_object = object, highlight_color = "navy",
background_color = "lightgray", file_path = "path/", file_name = "name", file_type = ".pdf",
single_pdf = TRUE)

## End(Not run)
```

Iterate_DimPlot_bySample

Iterate DimPlot By Sample

Description

Iterate DimPlot by orig.ident column from Seurat object metadata

Usage

```
Iterate_DimPlot_bySample(
  seurat_object,
  sample_column = "orig.ident",
  file_path = NULL,
  file_name = NULL,
  file_type = NULL,
  single_pdf = FALSE,
  output_width = NULL,
  output_height = NULL,
  dpi = 600,
  color = "black",
  no_legend = TRUE,
  title_prefix = NULL,
  reduction = NULL,
  dims = c(1, 2),
  pt.size = NULL,
```

```
  raster = NULL,
  ...
)
```

Arguments

seurat_object	Seurat object name.
sample_column	name of meta.data column containing sample names/ids (default is "orig.ident").
file_path	directory file path and/or file name prefix. Defaults to current wd.
file_name	name suffix to append after sample name.
file_type	File type to save output as. Must be one of following: ".pdf", ".png", ".tiff", ".jpeg", or ".svg".
single_pdf	saves all plots to single PDF file (default = FALSE). 'file_type" must be .pdf
output_width	the width (in inches) for output page size. Default is NULL.
output_height	the height (in inches) for output page size. Default is NULL.
dpi	dpi for image saving.
color	color scheme to use.
no_legend	logical, whether or not to include plot legend, default is TRUE.
title_prefix	Value that should be used for plot title prefix if no_legend = TRUE. If NULL the value of meta_data_column will be used. Default is NULL.
reduction	Dimensionality Reduction to use (default is object default).
dims	Dimensions to plot.
pt.size	Adjust point size for plotting.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
...	Extra parameters passed to DimPlot.

Value

A ggplot object

Examples

```
## Not run:
Iterate_DimPlot_bySample(seurat_object = object, file_path = "plots/", file_name = "tsne",
file_type = ".jpg", dpi = 600, color = "black")

## End(Not run)
```

`Iterate_FeaturePlot_scCustom` *Iterative Plotting of Gene Lists using Custom FeaturePlots*

Description

Create and Save plots for Gene list with Single Command

Usage

```
Iterate_FeaturePlot_scCustom(
  seurat_object,
  features,
  colors_use = viridis_plasma_dark_high,
  na_color = "lightgray",
  na_cutoff = 1e-09,
  split.by = NULL,
  order = TRUE,
  return_plots = FALSE,
  file_path = NULL,
  file_name = NULL,
  file_type = NULL,
  single_pdf = FALSE,
  output_width = NULL,
  output_height = NULL,
  features_per_page = 1,
  num_columns = NULL,
  landscape = TRUE,
  dpi = 600,
  pt.size = NULL,
  reduction = NULL,
  raster = NULL,
  alpha_exp = NULL,
  alpha_na_exp = NULL,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`features`	vector of features to plot. If a named vector is provided then the names for each gene will be incorporated into plot title if `single_pdf = TRUE` or into file name if `FALSE`.
`colors_use`	color scheme to use.
`na_color`	color for non-expressed cells.
`na_cutoff`	Value to use as minimum expression cutoff. To set no cutoff set to `NA`.

split.by	Variable in @meta.data to split the plot by.
order	whether to move positive cells to the top (default = TRUE).
return_plots	logical. Whether to return plots to list instead of saving them to file(s). Default is FALSE.
file_path	directory file path and/or file name prefix. Defaults to current wd.
file_name	name suffix and file extension.
file_type	File type to save output as. Must be one of following: ".pdf", ".png", ".tiff", ".jpeg", or ".svg".
single_pdf	saves all plots to single PDF file (default = FALSE).
output_width	the width (in inches) for output page size. Default is NULL.
output_height	the height (in inches) for output page size. Default is NULL.
features_per_page	numeric, number of features to plot on single page if single_pdf = TRUE. Default is 1.
num_columns	Number of columns in plot layout (only applicable if single_pdf = TRUE AND features_per_page > 1).
landscape	logical, when plotting multiple features per page in single PDF whether to use landscape or portrait page dimensions (default is TRUE).
dpi	dpi for image saving.
pt.size	Adjust point size for plotting.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
alpha_exp	new alpha level to apply to expressing cell color palette (colors_use). Must be value between 0-1.
alpha_na_exp	new alpha level to apply to non-expressing cell color palette (na_color). Must be value between 0-1.
...	Extra parameters passed to FeaturePlot.

Value

Saved plots

Examples

```
## Not run:
Iterate_FeaturePlot_scCustom(seurat_object = object, features = DEG_list,
colors_use = viridis_plasma_dark_high, na_color = "lightgray", file_path = "plots/",
file_name = "tsne", file_type = ".jpg", dpi = 600)

## End(Not run)
```

`Iterate_Meta_Highlight_Plot` *Iterate Meta Highlight Plot*

Description

Iterate the create plots with meta data variable of interest highlighted.

Usage

```
Iterate_Meta_Highlight_Plot(
  seurat_object,
  meta_data_column,
  new_meta_order = NULL,
  meta_data_sort = TRUE,
  highlight_color = "navy",
  background_color = "lightgray",
  pt.size = NULL,
  no_legend = FALSE,
  title_prefix = NULL,
  reduction = NULL,
  file_path = NULL,
  file_name = NULL,
  file_type = NULL,
  single_pdf = FALSE,
  output_width = NULL,
  output_height = NULL,
  dpi = 600,
  raster = NULL,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`meta_data_column`	Name of the column in `seurat_object@meta.data` slot to pull value from for highlighting.
`new_meta_order`	The order in which to plot each level within `meta_data_column` if `single_PDF` is TRUE.
`meta_data_sort`	logical. Whether or not to sort and relevel the levels in `meta_data_column` if `single_PDF` is TRUE. Default is TRUE.
`highlight_color`	Color to highlight cells (default "navy"). Can provide either single color to use for all clusters/plots or a vector of colors equal to the number of clusters to use (in order) for the clusters/plots.

background_color	non-highlighted cell colors.
pt.size	point size for both highlighted cluster and background.
no_legend	logical, whether or not to remove plot legend and move to plot title. Default is FALSE.
title_prefix	Value that should be used for plot title prefix if no_legend = TRUE. If NULL the value of meta_data_column will be used. Default is NULL.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
file_path	directory file path and/or file name prefix. Defaults to current wd.
file_name	name suffix to append after sample name.
file_type	File type to save output as. Must be one of following: ".pdf", ".png", ".tiff", ".jpeg", or ".svg".
single_pdf	saves all plots to single PDF file (default = FALSE). 'file_type" must be .pdf.
output_width	the width (in inches) for output page size. Default is NULL.
output_height	the height (in inches) for output page size. Default is NULL.
dpi	dpi for image saving.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
...	Extra parameters passed toDimPlot.

Value

Saved plots

Examples

```
## Not run:
Iterate_Meta_Highlight_Plot(seurat_object = object, meta_data_column = "sample_id",
highlight_color = "navy", background_color = "lightgray", file_path = "path/",
file_name = "name", file_type = ".pdf", single_pdf = TRUE)

## End(Not run)
```

Iterate_PC_Loading_Plots

Iterate PC Loading Plots

Description

Plot PC Heatmaps and Dim Loadings for exploratory analysis

Usage

```
Iterate_PC_Loading_Plots(
  seurat_object,
  dims_plot = NULL,
  file_path = NULL,
  name_prefix = NULL,
  file_name = "PC_Loading_Plots",
  return_plots = FALSE
)
```

Arguments

seurat_object	Seurat object name.
dims_plot	number of PCs to plot (integer). Default is all dims present in PCA.
file_path	directory file path to save file.
name_prefix	prefix for file name (optional).
file_name	suffix for file name. Default is "PC_Loading_Plots".
return_plots	Whether to return the plot list (Default is FALSE). Must assign to environment to save plot list.

Value

A list of plots outputted as pdf

See Also

PCHeatmap and VizDimLoadings

Examples

```
## Not run:
Iterate_PC_Loading_Plots(seurat_object = seurat, dims_plot = 25, file_path = "plots/")

## End(Not run)
```

Iterate_Plot_Density_Custom

Iterative Plotting of Gene Lists using Custom Density Plots

Description

Create and save plots for gene list with single command. Requires Nebulosa package from Bioconductor.

Usage

```
Iterate_Plot_Density_Custom(
  seurat_object,
  gene_list,
  viridis_palette = "magma",
  custom_palette = NULL,
  pt.size = 1,
  file_path = NULL,
  file_name = NULL,
  file_type = NULL,
  single_pdf = FALSE,
  output_width = NULL,
  output_height = NULL,
  dpi = 600,
  reduction = NULL,
  combine = TRUE,
  joint = FALSE,
  ...
)
```

Arguments

seurat_object	Seurat object name.
gene_list	vector of genes to plot. If a named vector is provided then the names for each gene will be incorporated into plot title if single_pdf = TRUE or into file name if FALSE.
viridis_palette	color scheme to use.
custom_palette	color for non-expressed cells.
pt.size	Adjust point size for plotting.
file_path	directory file path and/or file name prefix. Defaults to current wd.
file_name	name suffix and file extension.
file_type	File type to save output as. Must be one of following: ".pdf", ".png", ".tiff", ".jpeg", or ".svg".
single_pdf	saves all plots to single PDF file (default = FALSE). 'file_type" must be .pdf.
output_width	the width (in inches) for output page size. Default is NULL.
output_height	the height (in inches) for output page size. Default is NULL.
dpi	dpi for image saving.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default)
combine	Create a single plot? If FALSE, a list with ggplot objects is returned.
joint	NULL. This function only supports joint = FALSE. Leave as NULL to generate plots. To iterate joint plots see function: Iterate_Plot_Density_Joint.
...	Extra parameters passed to plot_density.

Value

Saved plots

Examples

```
## Not run:
Iterate_Plot_Density_Custom(seurat_object = object, gene_list = DEG_list, viridis_palette = "magma",
file_path = "plots/", file_name = "_density_plots", file_type = ".jpg", dpi = 600)

## End(Not run)
```

Iterate_Plot_Density_Joint

Iterative Plotting of Gene Lists using Custom Joint Density Plots

---

Description

Create and save plots for gene list with single command. Requires Nebulosa package from Bioconductor.

Usage

```
Iterate_Plot_Density_Joint(
  seurat_object,
  gene_list,
  viridis_palette = "magma",
  custom_palette = NULL,
  pt.size = 1,
  file_path = NULL,
  file_name = NULL,
  file_type = NULL,
  single_pdf = FALSE,
  output_width = NULL,
  output_height = NULL,
  dpi = 600,
  reduction = NULL,
  combine = TRUE,
  joint = NULL,
  ...
)
```

Arguments

seurat_object	Seurat object name.

gene_list	a list of vectors of genes to plot jointly. Each entry in the list will be plotted for the joint density. All entries in list must be greater than 2 features. If a named list is provided then the names for each list entry will be incorporated into plot title if single_pdf = TRUE or into file name if FALSE.
viridis_palette	color scheme to use.
custom_palette	color for non-expressed cells.
pt.size	Adjust point size for plotting.
file_path	directory file path and/or file name prefix. Defaults to current wd.
file_name	name suffix and file extension.
file_type	File type to save output as. Must be one of following: ".pdf", ".png", ".tiff", ".jpeg", or ".svg".
single_pdf	saves all plots to single PDF file (default = FALSE). 'file_type" must be .pdf.
output_width	the width (in inches) for output page size. Default is NULL.
output_height	the height (in inches) for output page size. Default is NULL.
dpi	dpi for image saving.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default)
combine	Create a single plot? If FALSE, a list with ggplot objects is returned.
joint	NULL. This function only supports joint = FALSE. Leave as NULL to generate plots. To iterate joint plots see function: Iterate_Plot_Density_Joint.
...	Extra parameters passed to plot_density.

Value

Saved plots

Examples

```
## Not run:
Iterate_Plot_Density_Joint(seurat_object = object, gene_list = DEG_list, viridis_palette = "magma",
file_path = "plots/", file_name = "joint_plots", file_type = ".jpg", dpi = 600)

## End(Not run)
```

Iterate_VlnPlot_scCustom

Iterative Plotting of Gene Lists using VlnPlot_scCustom

Description

Create and Save plots for Gene list with Single Command

Usage

```
Iterate_VlnPlot_scCustom(
  seurat_object,
  features,
  colors_use = NULL,
  pt.size = NULL,
  group.by = NULL,
  split.by = NULL,
  file_path = NULL,
  file_name = NULL,
  file_type = NULL,
  single_pdf = FALSE,
  output_width = NULL,
  output_height = NULL,
  raster = NULL,
  dpi = 600,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
features	vector of features to plot.
colors_use	color palette to use for plotting. By default if number of levels plotted is less than or equal to 36 it will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from DiscretePalette_scCustomize.
pt.size	point size for plotting.
group.by	Name of one or more metadata columns to group (color) plot by (for example, orig.ident); default is the current active.ident of the object.
split.by	Feature to split plots by (i.e. "orig.ident").
file_path	directory file path and/or file name prefix. Defaults to current wd.
file_name	name suffix and file extension.
file_type	File type to save output as. Must be one of following: ".pdf", ".png", ".tiff", ".jpeg", or ".svg".
single_pdf	saves all plots to single PDF file (default = FALSE). 'file_type" must be .pdf.
output_width	the width (in inches) for output page size. Default is NULL.
output_height	the height (in inches) for output page size. Default is NULL.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
dpi	dpi for image saving.
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.

color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

Saved plots

Examples

```
## Not run:
Iterate_VlnPlot_scCustom(seurat_object = object, features = DEG_list, colors = color_list,
file_path = "plots/", file_name = "_vln", file_type = ".jpg", dpi = 600)

## End(Not run)
```

JCO_Four *Four Color Palette (JCO)*

Description

Shortcut to a specific JCO 4 color palette from ggsci package.

Usage

```
JCO_Four()
```

Value

4 color palette from the JCO ggsci palette

References

Selection of colors from the JCO palette from ggsci being called through paletteer. See ggsci for more info on palettes https://CRAN.R-project.org/package=ggsci

Examples

```
cols <- JCO_Four()
PalettePlot(pal= cols)
```

`Liger_to_Seurat` *Create a Seurat object containing the data from a liger object* **[Soft-deprecated]**

Description

Merges raw.data and scale.data of object, and creates Seurat object with these values along with tsne.coords, iNMF factorization, and cluster assignments. Supports Seurat V2 and V3.

Usage

```
Liger_to_Seurat(
  liger_object,
  nms = names(liger_object@H),
  renormalize = TRUE,
  use.liger.genes = TRUE,
  by.dataset = FALSE,
  keep_meta = TRUE,
  reduction_label = "UMAP",
  seurat_assay = "RNA",
  assay_type = NULL,
  add_barcode_names = FALSE,
  barcode_prefix = TRUE,
  barcode_cell_id_delimiter = "_"
)
```

Arguments

`liger_object`	`liger` object.
`nms`	By default, labels cell names with dataset of origin (this is to account for cells in different datasets which may have same name). Other names can be passed here as vector, must have same length as the number of datasets. (default names(H)).
`renormalize`	Whether to log-normalize raw data using Seurat defaults (default TRUE).
`use.liger.genes`	Whether to carry over variable genes (default TRUE).
`by.dataset`	Include dataset of origin in cluster identity in Seurat object (default FALSE).
`keep_meta`	logical. Whether to transfer additional metadata (nGene/nUMI/dataset already transferred) to new Seurat Object. Default is TRUE.
`reduction_label`	Name of dimensionality reduction technique used. Enables accurate transfer or name to Seurat object instead of defaulting to "tSNE".
`seurat_assay`	Name to set for assay in Seurat Object. Default is "RNA".
`assay_type`	what type of Seurat assay to create in new object (Assay vs Assay5). Default is NULL which will default to the current user settings. See `Convert_Assay` parameter `convert_to` for acceptable values.

`add_barcode_names`
: logical, whether to add dataset names to the cell barcodes when creating Seurat object, default is FALSE.

`barcode_prefix`
: logical, if `add_barcode_names = TRUE` should the names be added as prefix to current cell barcodes/names or a suffix (default is TRUE; prefix).

`barcode_cell_id_delimiter`
: The delimiter to use when adding dataset id to barcode prefix/suffix. Default is "_".

Details

Stores original dataset identity by default in new object metadata if dataset names are passed in nms. iNMF factorization is stored in dim.reduction object with key "iNMF".

Value

Seurat object with raw.data, scale.data, reduction_label, iNMF, and ident slots set.

References

Original function is part of LIGER package https://github.com/welch-lab/liger (Licence: GPL-3). Function was slightly modified for use in scCustomize with keep.meta parameter. Also posted as PR to liger GitHub.

Examples

```
## Not run:
seurat_object <- Liger_to_Seurat(liger_object = LIGER_OBJ, reduction_label = "UMAP")

## End(Not run)
```

---

lncRNA_gene_list *lncRNA gene list*

Description

A list of gene symbol ids for lncRNA genes (Ensembl version 113; 04/08/2025)

Usage

```
lncRNA_gene_list
```

Format

A list of six vectors

Mus_musculus_lncRNA Ensembl IDs for mouse lncRNA genes

Homo_sapiens_lncRNA Ensembl IDs for human lncRNA genes

Danio_rerio_lncRNA Ensembl IDs for zebrafish lncRNA genes

Rattus_norvegicus_lncRNA Ensembl IDs for rat lncRNA genes

Macaca_mulatta_lncRNA Ensembl IDs for macaque lncRNA genes

Gallus_gallus_lncRNA Ensembl IDs for chicken lncRNA genes

Source

See data-raw directory for scripts used to create gene list.

MAD_Stats *Median Absolute Deviation Statistics*

Description

Get quick values for X x median absolute deviation for Genes, UMIs, %mito per cell grouped by meta.data variable.

Usage

```
MAD_Stats(
  seurat_object,
  group_by_var = deprecated(),
  group.by = "orig.ident",
  default_var = TRUE,
  mad_var = NULL,
  mad_num = 2
)
```

Arguments

seurat_object	Seurat object name.
group_by_var	**[Deprecated]** soft-deprecated. See group.by.
group.by	meta data column to classify samples (default = "orig.ident").
default_var	logical. Whether to include the default meta.data variables of: "nCount_RNA", "nFeature_RNA", "percent_mito", "percent_ribo", "percent_mito_ribo", and "log10GenesPerUMI" in addition to variables supplied to mad_var.
mad_var	Column(s) in @meta.data to calculate medians for in addition to defaults. Must be of class() integer or numeric.
mad_num	integer value to multiply the MAD in returned data.frame (default is 2). Often helpful when calculating a outlier range to base of of median + (X*MAD).

Value

A data.frame.

Examples

```
## Not run:
mad_stats <- MAD_Stats(seurat_object = obj, group.by = "orig.ident")

## End(Not run)
```

`Median_Stats` *Median Statistics*

Description

Get quick values for median Genes, UMIs, %mito per cell grouped by meta.data variable.

Usage

```
Median_Stats(
  seurat_object,
  group_by_var = deprecated(),
  group.by = "orig.ident",
  default_var = TRUE,
  median_var = NULL
)
```

Arguments

`seurat_object`	Seurat object name.
`group_by_var`	**[Deprecated]** soft-deprecated. See `group.by`.
`group.by`	meta data column to classify samples (default = "orig.ident").
`default_var`	logical. Whether to include the default meta.data variables of: "nCount_RNA", "nFeature_RNA", "percent_mito", "percent_ribo", "percent_mito_ribo", and "log10GenesPerUMI" in addition to variables supplied to `median_var`.
`median_var`	Column(s) in `@meta.data` to calculate medians for in addition to defaults. Must be of `class()` integer or numeric.

Value

A data.frame.

Examples

```
## Not run:
med_stats <- Median_Stats(seurat_object - obj, group.by = "orig.ident")

## End(Not run)
```

`Merge_Seurat_List` *Merge a list of Seurat Objects*

Description

Enables easy merge of a list of Seurat Objects. See See `merge` for more information,

Usage

```
Merge_Seurat_List(
  list_seurat,
  add.cell.ids = NULL,
  merge.data = TRUE,
  project = "SeuratProject"
)
```

Arguments

`list_seurat`	list composed of multiple Seurat Objects.
`add.cell.ids`	A character vector of equal length to the number of objects in `list_seurat`. Appends the corresponding values to the start of each objects' cell names. See `merge`.
`merge.data`	Merge the data slots instead of just merging the counts (which requires renormalization). This is recommended if the same normalization approach was applied to all objects. See `merge`.
`project`	Project name for the Seurat object. See `merge`.

Value

A Seurat Object

Examples

```
## Not run:
object_list <- list(obj1, obj2, obj3, ...)
merged_object <- Merge_Seurat_List(list_seurat = object_list)

## End(Not run)
```

`Merge_Sparse_Data_All` *Merge a list of Sparse Matrices*

Description

Enables easy merge of a list of sparse matrices

Usage

```
Merge_Sparse_Data_All(
  matrix_list,
  add_cell_ids = NULL,
  prefix = TRUE,
  cell_id_delimiter = "_"
)
```

Arguments

`matrix_list`	list of matrices to merge.
`add_cell_ids`	a vector of sample ids to add as prefix to cell barcode during merge.
`prefix`	logical. Whether `add_cell_ids` should be added as prefix to current cell barcodes/names or as suffix to current cell barcodes/names. Default is TRUE, add as prefix.
`cell_id_delimiter`	The delimiter to use when adding cell id prefix/suffix. Default is "_".

Value

A sparse Matrix

References

Original function is part of LIGER package https://github.com/welch-lab/liger/blob/master/R/mergeObject.R (License: GPL-3). Function was modified for use in scCustomize (add progress bar, prefix vs. suffix, and delimiter options).

Examples

```
## Not run:
data_list <- Read10X_GEO(...)
merged <- Merge_Sparse_Data_All(matrix_list = data_list, add_cell_ids = names(data_list),
prefix = TRUE, cell_id_delimiter = "_")

## End(Not run)
```

Merge_Sparse_Multimodal_All *Merge a list of Sparse Matrices contain multi-modal data.*

Description

Enables easy merge of a list of sparse matrices for multi-modal data.

Usage

```
Merge_Sparse_Multimodal_All(
  matrix_list,
  add_cell_ids = NULL,
  prefix = TRUE,
  cell_id_delimiter = "_"
)
```

Arguments

matrix_list	list of matrices to merge.
add_cell_ids	a vector of sample ids to add as prefix to cell barcode during merge.
prefix	logical. Whether add_cell_ids should be added as prefix to current cell barcodes/names or as suffix to current cell barcodes/names. Default is TRUE, add as prefix.
cell_id_delimiter	The delimiter to use when adding cell id prefix/suffix. Default is "_".

Value

A list containing one sparse matrix for each modality

Examples

```
## Not run:
data_list <- Read10X_GEO(...)
merged_list <- Merge_Sparse_Multimodal_All(matrix_list = data_list, add_cell_ids = names(data_list),
prefix = TRUE, cell_id_delimiter = "_")

## End(Not run)
```

`Meta_Highlight_Plot` *Meta Highlight Plot*

Description

Create Plot with meta data variable of interest highlighted

Usage

```
Meta_Highlight_Plot(
  seurat_object,
  meta_data_column,
  meta_data_highlight,
  highlight_color = NULL,
  background_color = "lightgray",
  pt.size = NULL,
  aspect_ratio = NULL,
  figure_plot = FALSE,
  raster = NULL,
  raster.dpi = c(512, 512),
  label = FALSE,
  split.by = NULL,
  split_seurat = FALSE,
  reduction = NULL,
  ggplot_default_colors = FALSE,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`meta_data_column`	Name of the column in `seurat_object@meta.data` slot to pull value from for highlighting.
`meta_data_highlight`	Name of variable(s) within `meta_data_name` to highlight in the plot.
`highlight_color`	Color to highlight cells (default "navy").
`background_color`	non-highlighted cell colors.
`pt.size`	point size for both highlighted cluster and background.
`aspect_ratio`	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
`figure_plot`	logical. Whether to remove the axes and plot with legend on left of plot denoting axes labels. (Default is FALSE). Requires `split_seurat = TRUE`.

raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
label	Whether to label the highlighted meta data variable(s). Default is FALSE.
split.by	Variable in @meta.data to split the plot by.
split_seurat	logical. Whether or not to display split plots like Seurat (shared y axis) or as individual plots in layout. Default is FALSE.
reduction	Dimensionality Reduction to use (if NULL then defaults to Object default).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
...	Extra parameters passed toDimPlot.

Value

A ggplot object

Examples

```
library(Seurat)
pbmc_small$sample_id <- sample(c("sample1", "sample2"), size = ncol(pbmc_small), replace = TRUE)

Meta_Highlight_Plot(seurat_object = pbmc_small, meta_data_column = "sample_id",
meta_data_highlight = "sample1", highlight_color = "gold", background_color = "lightgray",
pt.size = 2)
```

Meta_Numeric *Check if meta data columns are numeric*

Description

Check if any present meta data columns are numeric and returns vector of valid numeric columns. Issues warning message if any columns not in numeric form.

Usage

```
Meta_Numeric(data)
```

Arguments

data	a data.frame contain meta.data.

Value

vector of meta data columns that are numeric.

Examples

```
## Not run:
numeric_meta_columns <- Meta_Numeric(data = meta_data)

## End(Not run)
```

`Meta_Present` *Check if meta data are present*

Description

Check if meta data columns are present in object and return vector of found columns Return warning messages for meta data columns not found.

Usage

```
Meta_Present(
  object,
  meta_col_names,
  print_msg = TRUE,
  omit_warn = TRUE,
  return_none = FALSE
)
```

Arguments

`object`	Seurat or Liger object name.
`meta_col_names`	vector of column names to check.
`print_msg`	logical. Whether message should be printed if all features are found. Default is TRUE.
`omit_warn`	logical. Whether to print message about features that are not found in current object. Default is TRUE.
`return_none`	logical. Whether list of found vs. bad features should still be returned if no `meta_col_names` are found. Default is FALSE.

Value

vector of meta data columns that are present

Examples

```
## Not run:
meta_variables <- Meta_Present(object = obj_name, meta_col_names = "percent_mito", print_msg = TRUE)

## End(Not run)
```

`Meta_Remove_Seurat` *Remove meta data columns containing Seurat Defaults*

Description

Remove any columns from new meta_data data.frame in preparation for adding back to Seurat Object

Usage

```
Meta_Remove_Seurat(
  meta_data,
  seurat_object,
  barcodes_to_rownames = FALSE,
  barcodes_colname = "barcodes"
)
```

Arguments

`meta_data`	data.frame containing meta data.
`seurat_object`	object name.
`barcodes_to_rownames`	logical, are barcodes present as column and should they be moved to rownames (to be compatible with `Seurat::AddMetaData`). Default is FALSE.
`barcodes_colname`	name of barcodes column in meta_data. Required if `barcodes_to_rownames = TRUE`.

Value

data.frame with only new columns.

Examples

```
## Not run:
new_meta <- Meta_Remove_Seurat(meta_data = meta_data_df, seurat_object = object)
object <- AddMetaData(object = object, metadata = new_meta)

## End(Not run)
```

Move_Legend *Move Legend Position*

Description

Shortcut for thematic modification to move legend position.

Usage

```
Move_Legend(position = "right", ...)
```

Arguments

position	valid position to move legend. Default is "right".
...	extra arguments passed to ggplot2::theme().

Value

Returns a list-like object of class *theme*.

Examples

```
# Generate a plot and customize theme
library(ggplot2)
df <- data.frame(x = rnorm(n = 100, mean = 20, sd = 2), y = rbinom(n = 100, size = 100, prob = 0.2))
p <- ggplot(data = df, mapping = aes(x = x, y = y)) + geom_point(mapping = aes(color = 'red'))
p + Move_Legend("left")
```

msigdb_qc_ensembl_list *QC Gene Lists*

Description

Ensembl IDs for qc percentages from MSigDB database. The gene sets are from 3 MSigDB lists: "HALLMARK_OXIDATIVE_PHOSPHORYLATION", "HALLMARK_APOPTOSIS", and "HALLMARK_DNA_REPAIR". (Ensembl version 112; 4/29/2024)

Usage

```
msigdb_qc_ensembl_list
```

Format

A list of 21 vectors

Homo_sapiens_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for human

Homo_sapiens_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for human

Homo_sapiens_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for human

Mus_musculus_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for mouse

Mus_musculus_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for mouse

Mus_musculus_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for mouse

Rattus_norvegicus_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for rat

Rattus_norvegicus_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for rat

Rattus_norvegicus_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for rat

Drosophila_melanogaster_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for fly

Drosophila_melanogaster_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for fly

Drosophila_melanogaster_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for fly

Dario_rerio_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for zebrafish

Dario_rerio_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for zebrafish

Dario_rerio_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for zebrafish

Macaca_mulatta_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for macaque

Macaca_mulatta_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for macaque

Macaca_mulatta_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for macaque

Gallus_gallus_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for chicken

Gallus_gallus_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for chicken

Gallus_gallus_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for chicken

Source

MSigDB gene sets (ensembl IDs) via msigdbr package https://cran.r-project.org/package=msigdbr. See data-raw directory for scripts used to create gene list.

`msigdb_qc_gene_list` *QC Gene Lists*

Description

Gene symbols for qc percentages from MSigDB database. The gene sets are from 3 MSigDB lists: "HALLMARK_OXIDATIVE_PHOSPHORYLATION", "HALLMARK_APOPTOSIS", and "HALLMARK_DNA_REPAIR".

Usage

```
msigdb_qc_gene_list
```

Format

A list of 21 vectors

Homo_sapiens_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for human

Homo_sapiens_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for human

Homo_sapiens_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for human

Mus_musculus_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for mouse

Mus_musculus_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for mouse

Mus_musculus_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for mouse

Rattus_norvegicus_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for rat

Rattus_norvegicus_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for rat

Rattus_norvegicus_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for rat

Drosophila_melanogaster_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for fly

Drosophila_melanogaster_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for fly

Drosophila_melanogaster_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for fly

Dario_rerio_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for zebrafish

Dario_rerio_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for zebrafish

Dario_rerio_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for zebrafish

Macaca_mulatta_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for macaque

Macaca_mulatta_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for macaque

Macaca_mulatta_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for macaque

Gallus_gallus_msigdb_oxphos Genes in msigdb "HALLMARK_OXIDATIVE_PHOSPHORYLATION" list for chicken

Gallus_gallus_msigdb_apop Genes in msigdb "HALLMARK_APOPTOSIS" list for chicken

Gallus_gallus_msigdb_dna_repair Genes in msigdb "HALLMARK_DNA_REPAIR" list for chicken

Source

MSigDB gene sets (gene symbols) via msigdbr package https://cran.r-project.org/package=msigdbr. See data-raw directory for scripts used to create gene list.

NavyAndOrange *Navy and Orange Dual Color Palette*

Description

Shortcut to navy orange color plot

Usage

```
NavyAndOrange(flip_order = FALSE)
```

Arguments

flip_order	Whether to flip the order of colors.

Value

Navy orange palette

Examples

```
cols <- NavyAndOrange()
PalettePlot(pal= cols)
```

PalettePlot *Plot color palette in viewer*

Description

Plots given color vector/palette in viewer to evaluate palette before plotting on data.

Usage

```
PalettePlot(pal = NULL, label_color_num = NULL)
```

Arguments

pal	a vector of colors (either named colors of hex codes).
label_color_num	logical, whether or not to numerically label the colors in output plot. Default is TRUE is number of colors in pal is less than 75 and FALSE is greater than 75.

Value

Plot of all colors in supplied palette/vector

References

Adapted from colorway package build_palette internals (License: GPL-3). https://github.com/hypercompetent/colorway.

Examples

```
pal <- DiscretePalette_scCustomize(num_colors = 36, palette = "varibow")
PalettePlot(pal = pal)
```

PC_Plotting *PC Plots*

Description

Plot PC Heatmaps and Dim Loadings for exploratory analysis. Plots a single Heatmap and Gene Loading Plot. Used for PC_Loading_Plots function.

Usage

```
PC_Plotting(seurat_object, dim_number)
```

Arguments

seurat_object	Seurat Object.
dim_number	A single dim to plot (integer).

Value

A plot of PC heatmap and gene loadings for single

See Also

PCHeatmap and VizDimLoadings

Examples

```
library(Seurat)
PC_Plotting(seurat_object = pbmc_small, dim_number = 1)
```

Percent_Expressing *Calculate percent of expressing cells*

Description

Calculates the percent of cells that express a given set of features by various grouping factors

Usage

```
Percent_Expressing(
  seurat_object,
  features,
  threshold = 0,
  group_by = deprecated(),
  group.by = NULL,
  split_by = deprecated(),
  split.by = NULL,
  entire_object = FALSE,
  layer = "data",
  assay = NULL
)
```

Arguments

seurat_object	Seurat object name.
features	Feature(s) to plot.
threshold	Expression threshold to use for calculation of percent expressing (default is 0).
group_by	**[Deprecated]** soft-deprecated. See group.by.

group.by	Factor to group the cells by.
split_by	**[Deprecated]** soft-deprecated. See split.by.
split.by	Factor to split the groups by.
entire_object	logical (default = FALSE). Whether to calculate percent of expressing cells across the entire object as opposed to by cluster or by group.by variable.
layer	Which layer to pull expression data from? Default is "data".
assay	Assay to pull feature data from. Default is active assay.

Value

A data.frame

References

Part of code is modified from Seurat package as used by DotPlot to generate values to use for plotting. Source code can be found here: https://github.com/satijalab/seurat/blob/4e868fcde49dc0a3df47f94f5fb54R/visualization.R#L3391 (License: GPL-3).

Examples

```
## Not run:
percent_stats <- Percent_Expressing(seurat_object = object, features = "Cx3cr1", threshold = 0)

## End(Not run)
```

---

plotFactors_scCustom *Customized version of plotFactors*

Description

Modified and optimized version of plotFactors function from LIGER package.

Usage

```
plotFactors_scCustom(
  liger_object,
  num_genes = 8,
  colors_use_factors = NULL,
  colors_use_dimreduc = c("lemonchiffon", "red"),
  pt.size_factors = 1,
  pt.size_dimreduc = 1,
  reduction = "UMAP",
  reduction_label = "UMAP",
  plot_legend = TRUE,
  raster = TRUE,
```

```
  raster.dpi = c(512, 512),
  order = FALSE,
  plot_dimreduc = TRUE,
  save_plots = TRUE,
  file_path = NULL,
  file_name = NULL,
  return_plots = FALSE,
  cells.highlight = NULL,
  reorder_datasets = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123
)
```

Arguments

liger_object	liger liger_object. Need to perform clustering and factorization before calling this function
num_genes	Number of genes to display for each factor (Default 8).
colors_use_factors	colors to use for plotting factor loadings By default datasets will be plotted using "varibow" with shuffle = TRUE from both from DiscretePalette_scCustomize.
colors_use_dimreduc	colors to use for plotting factor loadings on dimensionality reduction coordinates (tSNE/UMAP). Default is c('lemonchiffon', 'red'),
pt.size_factors	Adjust point size for plotting in the factor plots.
pt.size_dimreduc	Adjust point size for plotting in dimensionality reduction plots.
reduction	Name of dimensionality reduction to use for plotting. Default is "UMAP". Only for newer style liger objects.
reduction_label	What to label the x and y axes of resulting plots. LIGER does not store name of technique and therefore needs to be set manually. Default is "UMAP". Only for older style liger objects.
plot_legend	logical, whether to plot the legend on factor loading plots, default is TRUE. Helpful if number of datasets is large to avoid crowding the plot with legend.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 200,000 cells.
raster.dpi	Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).
order	logical. Whether to plot higher loading cells on top of cells with lower loading values in the dimensionality reduction plots (Default = FALSE).
plot_dimreduc	logical. Whether to plot factor loadings on dimensionality reduction coordinates. Default is TRUE.
save_plots	logical. Whether to save plots. Default is TRUE

file_path	directory file path and/or file name prefix. Defaults to current wd.
file_name	name suffix to append after sample name.
return_plots	logical. Whether or not to return plots to the environment. (Default is FALSE)
cells.highlight	Names of specific cells to highlight in plot (black) (default NULL).
reorder_datasets	New order to plot datasets in for the factor plots if different from current factor level order in cell.data slot. Only for older style liger objects.
ggplot_default_colors	logical. If colors_use_factors = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "varibow" palette.
color_seed	random seed for the palette shuffle if colors_use_factors = NULL. Default = 123.

Value

A list of ggplot/patchwork objects and/or PDF file.

Author(s)

Velina Kozareva (Original code for modified function), Sam Marsh (Added/modified functionality)

References

Based on plotFactors functionality from original LIGER package.

Examples

```
## Not run:
plotFactors_scCustom(liger_object = liger_obj, return_plots = FALSE, plot_dimreduc = TRUE,
raster = FALSE, save_plots = TRUE)

## End(Not run)
```

Plot_Cells_per_Sample *Plot Number of Cells/Nuclei per Sample*

Description

Plot of total cell or nuclei number per sample grouped by another meta data variable.

Usage

```
Plot_Cells_per_Sample(
  seurat_object,
  sample_col = "orig.ident",
  group_by = deprecated(),
  group.by = NULL,
  colors_use = NULL,
  dot_size = 1,
  plot_title = "Cells/Nuclei per Sample",
  y_axis_label = "Number of Cells",
  x_axis_label = NULL,
  legend_title = NULL,
  x_lab_rotate = TRUE,
  color_seed = 123
)
```

Arguments

seurat_object	Seurat object name.
sample_col	Specify which column in meta.data specifies sample ID (i.e. orig.ident).
group_by	**[Deprecated]** soft-deprecated. See group.by.
group.by	Column in meta.data slot to group results by (i.e. "Treatment").
colors_use	List of colors or color palette to use.
dot_size	size of the dots plotted if group.by is not NULL. Default is 1.
plot_title	Plot title.
y_axis_label	Label for y axis.
x_axis_label	Label for x axis.
legend_title	Label for plot legend.
x_lab_rotate	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Value

A ggplot object

Examples

```
## Not run:
Plot_Cells_per_Sample(seurat_object = obj, sample_col = "orig.ident", group.by = "Treatment")

## End(Not run)
```

`Plot_Density_Custom` *Nebulosa Density Plot*

Description

Allow for customization of Nebulosa plot_density. Requires Nebulosa package from Bioconductor.

Usage

```
Plot_Density_Custom(
  seurat_object,
  features,
  joint = FALSE,
  viridis_palette = "magma",
  custom_palette = NULL,
  pt.size = 1,
  aspect_ratio = NULL,
  reduction = NULL,
  combine = TRUE,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`features`	Features to plot.
`joint`	logical. Whether to return joint density plot. Default is FALSE.
`viridis_palette`	default viridis palette to use (must be one of: "viridis", "magma", "cividis", "inferno", "plasma"). Default is "magma".
`custom_palette`	non-default color palette to be used in place of default viridis options.
`pt.size`	Adjust point size for plotting.
`aspect_ratio`	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
`reduction`	Dimensionality Reduction to use (if NULL then defaults to Object default).
`combine`	Create a single plot? If FALSE, a list with ggplot objects is returned.
`...`	Extra parameters passed to `plot_density`.

Value

A ggplot object

Examples

```
## Not run:
library(Seurat)
Plot_Density_Custom(seurat_object = pbmc_small, features = "CD3E")

## End(Not run)
```

`Plot_Density_Joint_Only` *Nebulosa Joint Density Plot*

Description

Return only the joint density plot from Nebulosa plot_density function. Requires Nebulosa package from Bioconductor.

Usage

```
Plot_Density_Joint_Only(
  seurat_object,
  features,
  viridis_palette = "magma",
  custom_palette = NULL,
  pt.size = 1,
  aspect_ratio = NULL,
  reduction = NULL,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`features`	Features to plot.
`viridis_palette`	default viridis palette to use (must be one of: "viridis", "magma", "cividis", "inferno", "plasma"). Default is "magma".
`custom_palette`	non-default color palette to be used in place of default viridis options.
`pt.size`	Adjust point size for plotting.
`aspect_ratio`	Control the aspect ratio (y:x axes ratio length). Must be numeric value; Default is NULL.
`reduction`	Dimensionality Reduction to use (if NULL then defaults to Object default).
`...`	Extra parameters passed to `plot_density`.

Value

A ggplot object

Examples

```
## Not run:
library(Seurat)
Plot_Density_Joint_Only(seurat_object = pbmc_small, features = c("CD8A", "CD3E"))

## End(Not run)
```

`Plot_Median_Genes` *Plot Median Genes per Cell per Sample*

Description

Plot of median genes per cell per sample grouped by desired meta data variable.

Usage

```
Plot_Median_Genes(
  seurat_object,
  sample_col = "orig.ident",
  group_by = deprecated(),
  group.by = NULL,
  colors_use = NULL,
  dot_size = 1,
  plot_title = "Median Genes/Cell per Sample",
  y_axis_label = "Median Genes",
  x_axis_label = NULL,
  legend_title = NULL,
  x_lab_rotate = TRUE,
  color_seed = 123
)
```

Arguments

`seurat_object`	Seurat object name.
`sample_col`	Specify which column in meta.data specifies sample ID (i.e. orig.ident).
`group_by`	**[Deprecated]** soft-deprecated. See `group.by`.
`group.by`	Column in meta.data slot to group results by (i.e. "Treatment").
`colors_use`	List of colors or color palette to use. Only applicable if `group.by` is not NULL.
`dot_size`	size of the dots plotted if `group.by` is not NULL. Default is 1.
`plot_title`	Plot title.

y_axis_label	Label for y axis.
x_axis_label	Label for x axis.
legend_title	Label for plot legend.
x_lab_rotate	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Value

A ggplot object

Examples

```
library(Seurat)
# Create example groups
pbmc_small$sample_id <- sample(c("sample1", "sample2"), size = ncol(pbmc_small), replace = TRUE)

# Plot
Plot_Median_Genes(seurat_object = pbmc_small, sample_col = "orig.ident", group.by = "sample_id")
```

Plot_Median_Mito *Plot Median Percent Mito per Cell per Sample*

Description

Plot of median percent mito per cell per sample grouped by desired meta data variable.

Usage

```
Plot_Median_Mito(
  seurat_object,
  sample_col = "orig.ident",
  group_by = deprecated(),
  group.by = NULL,
  colors_use = NULL,
  dot_size = 1,
  plot_title = "Median % Mito per Sample",
  y_axis_label = "Percent Mitochondrial Reads",
  x_axis_label = NULL,
  legend_title = NULL,
  x_lab_rotate = TRUE,
  color_seed = 123
)
```

Arguments

seurat_object	Seurat object name.
sample_col	Specify which column in meta.data specifies sample ID (i.e. orig.ident).
group_by	**[Deprecated]** soft-deprecated. See group.by.
group.by	Column in meta.data slot to group results by (i.e. "Treatment").
colors_use	List of colors or color palette to use. Only applicable if group.by is not NULL.
dot_size	size of the dots plotted if group.by is not NULL. Default is 1.
plot_title	Plot title.
y_axis_label	Label for y axis.
x_axis_label	Label for x axis.
legend_title	Label for plot legend.
x_lab_rotate	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Value

A ggplot object

Examples

```
## Not run:
# Add mito
obj <- Add_Mito_Ribo_Seurat(seurat_object = obj, species = "human")

# Plot
Plot_Median_Mito(seurat_object = obj, sample_col = "orig.ident",  group.by = "sample_id")

## End(Not run)
```

Plot_Median_Other *Plot Median other variable per Cell per Sample*

Description

Plot of median other variable per cell per sample grouped by desired meta data variable.

Usage

```
Plot_Median_Other(
  seurat_object,
  median_var,
  sample_col = "orig.ident",
  group_by = deprecated(),
  group.by = NULL,
  colors_use = NULL,
  dot_size = 1,
  plot_title = NULL,
  y_axis_label = NULL,
  x_axis_label = NULL,
  legend_title = NULL,
  x_lab_rotate = TRUE,
  color_seed = 123
)
```

Arguments

seurat_object	Seurat object name.
median_var	Variable in meta.data slot to calculate and plot median values for.
sample_col	Specify which column in meta.data specifies sample ID (i.e. orig.ident).
group_by	**[Deprecated]** soft-deprecated. See group.by.
group.by	Column in meta.data slot to group results by (i.e. "Treatment").
colors_use	List of colors or color palette to use. Only applicable if group.by is not NULL.
dot_size	size of the dots plotted if group.by is not NULL. Default is 1.
plot_title	Plot title.
y_axis_label	Label for y axis.
x_axis_label	Label for x axis.
legend_title	Label for plot legend.
x_lab_rotate	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Value

A ggplot object

Examples

```
## Not run:
library(Seurat)
cd_features <- list(c('CD79B', 'CD79A', 'CD19', 'CD180', 'CD200', 'CD3D', 'CD2','CD3E',
'CD7','CD8A', 'CD14', 'CD1C', 'CD68', 'CD9', 'CD247'))
```

```
pbmc_small <- AddModuleScore(object = pbmc_small, features = cd_features, ctrl = 5,
name = 'CD_Features')

Plot_Median_Other(seurat_object = pbmc_small, median_var = "CD_Features1",
sample_col = "orig.ident", group.by = "Treatment")

## End(Not run)
```

`Plot_Median_UMIs` *Plot Median UMIs per Cell per Sample*

Description

Plot of median UMIs per cell per sample grouped by desired meta data variable.

Usage

```
Plot_Median_UMIs(
  seurat_object,
  sample_col = "orig.ident",
  group_by = deprecated(),
  group.by = NULL,
  colors_use = NULL,
  dot_size = 1,
  plot_title = "Median UMIs/Cell per Sample",
  y_axis_label = "Median UMIs",
  x_axis_label = NULL,
  legend_title = NULL,
  x_lab_rotate = TRUE,
  color_seed = 123
)
```

Arguments

`seurat_object`	Seurat object name.
`sample_col`	Specify which column in meta.data specifies sample ID (i.e. orig.ident).
`group_by`	**[Deprecated]** soft-deprecated. See `group.by`.
`group.by`	Column in meta.data slot to group results by (i.e. "Treatment").
`colors_use`	List of colors or color palette to use. Only applicable if `group.by` is not NULL.
`dot_size`	size of the dots plotted if `group.by` is not NULL. Default is 1.
`plot_title`	Plot title.
`y_axis_label`	Label for y axis.
`x_axis_label`	Label for x axis.
`legend_title`	Label for plot legend.

x_lab_rotate	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Value

A ggplot object

Examples

```
library(Seurat)
# Create example groups
pbmc_small$sample_id <- sample(c("sample1", "sample2"), size = ncol(pbmc_small), replace = TRUE)

# Plot
Plot_Median_UMIs(seurat_object = pbmc_small, sample_col = "orig.ident",  group.by = "sample_id")
```

Proportion_Plot *Cell Proportion Plot*

Description

Plots the proportion of cells belonging to each identity in active.ident of Seurat object. Can plot either the totals or split by a variable in meta.data.

Usage

```
Proportion_Plot(
  seurat_object,
  plot_type = "bar",
  plot_scale = "percent",
  group_by_var = deprecated(),
  group.by = "ident",
  split.by = NULL,
  num_columns = NULL,
  x_lab_rotate = TRUE,
  colors_use = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123
)
```

Arguments

seurat_object	Seurat object name.
plot_type	whether to plot a pie chart or bar chart; value must be one of "bar" or "pie". Default is "bar"

plot_scale	whether to plot bar chart as total cell counts or percents, value must be one of "percent" or "count". Default is "percent".
group_by_var	**[Deprecated]** soft-deprecated. See group.by.
group.by	meta data column to classify samples (default = "ident" and will use active.ident).
split.by	meta data variable to use to split plots. Default is NULL which will plot across entire object.
num_columns	number of columns in plot. Only valid if split.by is not NULL.
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is FALSE). Only valid if plot_type = "bar".
colors_use	color palette to use for plotting.
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Value

ggplot2 or patchwork object

Examples

```
#' library(Seurat)
Proportion_Plot(seurat_object = pbmc_small)
```

Proportion_Plot_per_Sample

Cell Proportion Plot per Sample

Description

Plots the proportion of cells belonging to each identity per sample split by grouping variable/condition.

Usage

```
Proportion_Plot_per_Sample(
  seurat_object,
  cluster = "ident",
  split.by,
  sample_col,
  pt.size = 1.5,
  x_lab_rotate = TRUE,
  colors_use = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123
)
```

Arguments

seurat_object	Seurat object name.
cluster	name of meta.data column containing cluster values. Default is ident which defaults to current active.ident.
split.by	name of meta.data column containing sample group/condition variable.
sample_col	name of meta.data column that contains sample ID information.
pt.size	the size of points in plot (default is 1.5).
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is FALSE). Only valid if plot_type = "bar".
colors_use	color palette to use for plotting.
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

Examples

```
## Not run:
Proportion_Plot_per_Sample(seurat_object = obj, split.by = "Diagnosis",
sample_col = "orig.ident")

## End(Not run)
```

Pull_Cluster_Annotation

Pull cluster information from annotation csv file.

Description

shortcut filter and pull function compatible with annotation files created by Create_Cluster_Annotation_File by default but also any other csv file.

Usage

```
Pull_Cluster_Annotation(
  annotation = NULL,
  cluster_name_col = "cluster",
  cell_type_col = "cell_type"
)
```

Arguments

annotation	name of the data.frame/tibble or path to CSV file containing cluster annotation.
cluster_name_col	name of column containing cluster names/numbers (default is "cluster").
cell_type_col	name of column contain the cell type annotation (default is "cell_type").

Value

a list of named vectors for every cell type in the `cell_type_col` column of the annotation table and vectors new cluster names (for use with `Rename_Clusters` function or manual identity renaming).

Examples

```
## Not run:
# If pulling from a data.frame/tibble
cluster_annotation <- Pull_Cluster_Annotation(annotation = annotation_df,
cluster_name_col = "cluster", cell_type_col = "cell_type")

# If pulling from csv file
cluster_annotation <- Pull_Cluster_Annotation(annotation = "file_path/file_name.csv",
cluster_name_col = "cluster", cell_type_col = "cell_type")

## End(Not run)
```

Pull_Directory_List *Pull Directory List*

Description

Enables easy listing of all sub-directories for use as input library lists in Read10X multi functions.

Usage

```
Pull_Directory_List(base_path)
```

Arguments

base_path	path to the parent directory which contains all of the subdirectories of interest.

Value

A vector of sub-directories within `base_path`.

Examples

```
## Not run:
data_dir <- 'path/to/data/directory'
library_list <- Pull_Directory_List(base_path = data_dir)

## End(Not run)
```

QC_Histogram *QC Histogram Plots*

Description

Custom histogram for initial QC checks including lines for thresholding

Usage

```
QC_Histogram(
  seurat_object,
  features,
  low_cutoff = NULL,
  high_cutoff = NULL,
  cutoff_line_width = NULL,
  split.by = NULL,
  bins = 250,
  colors_use = "dodgerblue",
  num_columns = NULL,
  plot_title = NULL,
  assay = NULL,
  print_defaults = FALSE
)
```

Arguments

seurat_object	Seurat object name.
features	Feature from meta.data, assay features, or feature name shortcut to plot.
low_cutoff	Plot line a potential low threshold for filtering.
high_cutoff	Plot line a potential high threshold for filtering.
cutoff_line_width	numerical value for thickness of cutoff lines, default is NULL.
split.by	Feature to split plots by (i.e. "orig.ident").
bins	number of bins to plot default is 250.
colors_use	color to fill histogram bars, default is "dodgerblue".
num_columns	Number of columns in plot layout.

plot_title	optional, vector to use for plot title. Default is the name of the variable being plotted.
assay	assay to pull features from, default is active assay.
print_defaults	return list of accepted default shortcuts to provide to features instead of full name.

Value

A patchwork object

Examples

```
## Not run:
QC_Histogram(seurat_object = object, features = "nFeature_RNA")

## End(Not run)
```

QC_Plots_Combined_Vln *QC Plots Genes, UMIs, & % Mito*

Description

Custom VlnPlot for initial QC checks including lines for thresholding

Usage

```
QC_Plots_Combined_Vln(
  seurat_object,
  group.by = NULL,
  feature_cutoffs = NULL,
  UMI_cutoffs = NULL,
  mito_cutoffs = NULL,
  mito_name = "percent_mito",
  cutoff_line_width = NULL,
  pt.size = NULL,
  plot_median = FALSE,
  median_size = 15,
  plot_boxplot = FALSE,
  colors_use = NULL,
  x_lab_rotate = TRUE,
  y_axis_log = FALSE,
  raster = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
feature_cutoffs	Numeric vector of length 1 or 2 to plot lines for potential low/high threshold for filtering.
UMI_cutoffs	Numeric vector of length 1 or 2 to plot lines for potential low/high threshold for filtering.
mito_cutoffs	Numeric vector of length 1 or 2 to plot lines for potential low/high threshold for filtering.
mito_name	The column name containing percent mitochondrial counts information. Default value is "percent_mito" which is default value created when using Add_Mito_Ribo().
cutoff_line_width	numerical value for thickness of cutoff lines, default is NULL.
pt.size	Point size for plotting
plot_median	logical, whether to plot median for each ident on the plot (Default is FALSE).
median_size	Shape size for the median is plotted.
plot_boxplot	logical, whether to plot boxplot inside of violin (Default is FALSE).
colors_use	vector of colors to use for plot.
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is TRUE).
y_axis_log	logical. Whether to change y axis to log10 scale (Default is FALSE).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

A ggplot object

Examples

```
## Not run:
QC_Plots_Combined_Vln(seurat_object = object)

## End(Not run)
```

`QC_Plots_Complexity` *QC Plots Cell "Complexity"*

Description

Custom VlnPlot for initial QC checks including lines for thresholding

Usage

```
QC_Plots_Complexity(
  seurat_object,
  feature = "log10GenesPerUMI",
  group.by = NULL,
  x_axis_label = NULL,
  y_axis_label = "log10(Genes) / log10(UMIs)",
  plot_title = "Cell Complexity",
  low_cutoff = NULL,
  high_cutoff = NULL,
  cutoff_line_width = NULL,
  pt.size = NULL,
  plot_median = FALSE,
  plot_boxplot = FALSE,
  median_size = 15,
  colors_use = NULL,
  x_lab_rotate = TRUE,
  y_axis_log = FALSE,
  raster = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`feature`	Feature from Meta Data to plot.
`group.by`	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
`x_axis_label`	Label for x axis.
`y_axis_label`	Label for y axis.
`plot_title`	Plot Title.
`low_cutoff`	Plot line a potential low threshold for filtering.
`high_cutoff`	Plot line a potential high threshold for filtering.
`cutoff_line_width`	numerical value for thickness of cutoff lines, default is NULL.

pt.size	Point size for plotting
plot_median	logical, whether to plot median for each ident on the plot (Default is FALSE).
plot_boxplot	logical, whether to plot boxplot inside of violin (Default is FALSE).
median_size	Shape size for the median is plotted.
colors_use	vector of colors to use for plot.
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is TRUE).
y_axis_log	logical. Whether to change y axis to log10 scale (Default is FALSE).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

A ggplot object

Examples

```
library(Seurat)
pbmc_small <- Add_Cell_Complexity(pbmc_small)

QC_Plots_Complexity(seurat_object = pbmc_small)
```

QC_Plots_Feature *QC Plots Feature*

Description

Custom VlnPlot for initial QC checks including lines for thresholding

Usage

```
QC_Plots_Feature(
  seurat_object,
  feature,
  group.by = NULL,
  x_axis_label = NULL,
  y_axis_label = NULL,
  plot_title = NULL,
  low_cutoff = NULL,
```

```
  high_cutoff = NULL,
  cutoff_line_width = NULL,
  pt.size = NULL,
  plot_median = FALSE,
  median_size = 15,
  plot_boxplot = FALSE,
  colors_use = NULL,
  x_lab_rotate = TRUE,
  y_axis_log = FALSE,
  raster = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
feature	Feature from Meta Data to plot.
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
x_axis_label	Label for x axis.
y_axis_label	Label for y axis.
plot_title	Plot Title.
low_cutoff	Plot line a potential low threshold for filtering.
high_cutoff	Plot line a potential high threshold for filtering.
cutoff_line_width	numerical value for thickness of cutoff lines, default is NULL.
pt.size	Point size for plotting.
plot_median	logical, whether to plot median for each ident on the plot (Default is FALSE).
median_size	Shape size for the median is plotted.
plot_boxplot	logical, whether to plot boxplot inside of violin (Default is FALSE).
colors_use	vector of colors to use for plot.
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is TRUE).
y_axis_log	logical. Whether to change y axis to log10 scale (Default is FALSE).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

A ggplot object

Examples

```
## Not run:
QC_Plots_Feature(seurat_object = object, feature = "FEATURE_NAME",
y_axis_label = "FEATURE per Cell", plot_title = "FEATURE per Cell", high_cutoff = 10,
low_cutoff = 2)

## End(Not run)
```

QC_Plots_Genes *QC Plots Genes*

Description

Custom VlnPlot for initial QC checks including lines for thresholding

Usage

```
QC_Plots_Genes(
  seurat_object,
  plot_title = "Genes Per Cell/Nucleus",
  group.by = NULL,
  x_axis_label = NULL,
  y_axis_label = "Features",
  low_cutoff = NULL,
  high_cutoff = NULL,
  cutoff_line_width = NULL,
  pt.size = NULL,
  plot_median = FALSE,
  plot_boxplot = FALSE,
  median_size = 15,
  colors_use = NULL,
  x_lab_rotate = TRUE,
  y_axis_log = FALSE,
  raster = NULL,
  assay = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
plot_title	Plot Title.
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
x_axis_label	Label for x axis.
y_axis_label	Label for y axis.
low_cutoff	Plot line a potential low threshold for filtering.
high_cutoff	Plot line a potential high threshold for filtering.
cutoff_line_width	numerical value for thickness of cutoff lines, default is NULL.
pt.size	Point size for plotting.
plot_median	logical, whether to plot median for each ident on the plot (Default is FALSE).
plot_boxplot	logical, whether to plot boxplot inside of violin (Default is FALSE).
median_size	Shape size for the median is plotted.
colors_use	vector of colors to use for plot.
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is TRUE).
y_axis_log	logical. Whether to change y axis to log10 scale (Default is FALSE).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
assay	Name of assay to use, defaults to the active assay.
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

A ggplot object

Examples

```
library(Seurat)
QC_Plots_Genes(seurat_object = pbmc_small, plot_title = "Genes per Cell", low_cutoff = 40,
high_cutoff = 85)
```

QC_Plots_Mito *QC Plots Mito*

Description

#' Custom VlnPlot for initial QC checks including lines for thresholding

Usage

```
QC_Plots_Mito(
  seurat_object,
  mito_name = "percent_mito",
  plot_title = "Mito Gene % per Cell/Nucleus",
  group.by = NULL,
  x_axis_label = NULL,
  y_axis_label = "% Mitochondrial Gene Counts",
  low_cutoff = NULL,
  high_cutoff = NULL,
  cutoff_line_width = NULL,
  pt.size = NULL,
  plot_median = FALSE,
  median_size = 15,
  plot_boxplot = FALSE,
  colors_use = NULL,
  x_lab_rotate = TRUE,
  y_axis_log = FALSE,
  raster = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
mito_name	The column name containing percent mitochondrial counts information. Default value is "percent_mito" which is default value created when using `Add_Mito_Ribo()`.
plot_title	Plot Title.
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
x_axis_label	Label for x axis.
y_axis_label	Label for y axis.
low_cutoff	Plot line a potential low threshold for filtering.
high_cutoff	Plot line a potential high threshold for filtering.

cutoff_line_width	numerical value for thickness of cutoff lines, default is NULL.
pt.size	Point size for plotting.
plot_median	logical, whether to plot median for each ident on the plot (Default is FALSE).
median_size	Shape size for the median is plotted.
plot_boxplot	logical, whether to plot boxplot inside of violin (Default is FALSE).
colors_use	vector of colors to use for plot.
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is TRUE).
y_axis_log	logical. Whether to change y axis to log10 scale (Default is FALSE).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

A ggplot object

Examples

```
## Not run:
QC_Plots_Mito(seurat_object = object, plot_title = "Percent Mito per Cell", high_cutoff = 10)

## End(Not run)
```

QC_Plots_UMIs *QC Plots UMIs*

Description

#' Custom VlnPlot for initial QC checks including lines for thresholding

Usage

```
QC_Plots_UMIs(
  seurat_object,
  plot_title = "UMIs per Cell/Nucleus",
  group.by = NULL,
  x_axis_label = NULL,
  y_axis_label = "UMIs",
```

```
  low_cutoff = NULL,
  high_cutoff = NULL,
  cutoff_line_width = NULL,
  pt.size = NULL,
  plot_median = FALSE,
  median_size = 15,
  plot_boxplot = FALSE,
  colors_use = NULL,
  x_lab_rotate = TRUE,
  y_axis_log = FALSE,
  raster = NULL,
  assay = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
plot_title	Plot Title.
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
x_axis_label	Label for x axis.
y_axis_label	Label for y axis.
low_cutoff	Plot line a potential low threshold for filtering.
high_cutoff	Plot line a potential high threshold for filtering.
cutoff_line_width	numerical value for thickness of cutoff lines, default is NULL.
pt.size	Point size for plotting.
plot_median	logical, whether to plot median for each ident on the plot (Default is FALSE).
median_size	Shape size for the median is plotted.
plot_boxplot	logical, whether to plot boxplot inside of violin (Default is FALSE).
colors_use	vector of colors to use for plot.
x_lab_rotate	Rotate x-axis labels 45 degrees (Default is TRUE).
y_axis_log	logical. Whether to change y axis to log10 scale (Default is FALSE).
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
assay	Name of assay to use, defaults to the active assay.
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

A ggplot object

Examples

```
library(Seurat)
QC_Plots_UMIs(seurat_object = pbmc_small, plot_title = "UMIs per Cell", low_cutoff = 75,
high_cutoff = 600)
```

QC_Plot_GenevsFeature *QC Plots Genes vs Misc*

Description

Custom FeatureScatter for initial QC checks including lines for thresholding

Usage

```
QC_Plot_GenevsFeature(
  seurat_object,
  feature1,
  x_axis_label = NULL,
  y_axis_label = "Genes per Cell/Nucleus",
  low_cutoff_gene = NULL,
  high_cutoff_gene = NULL,
  low_cutoff_feature = NULL,
  high_cutoff_feature = NULL,
  cutoff_line_width = NULL,
  colors_use = NULL,
  pt.size = 1,
  group.by = NULL,
  raster = NULL,
  raster.dpi = c(512, 512),
  assay = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  shuffle_seed = 1,
  ...
)
```

Arguments

seurat_object	Seurat object name.
feature1	First feature to plot.
x_axis_label	Label for x axis.

`y_axis_label` Label for y axis.

`low_cutoff_gene`
Plot line a potential low threshold for filtering genes per cell.

`high_cutoff_gene`
Plot line a potential high threshold for filtering genes per cell.

`low_cutoff_feature`
Plot line a potential low threshold for filtering feature1 per cell.

`high_cutoff_feature`
Plot line a potential high threshold for filtering feature1 per cell.

`cutoff_line_width`
numerical value for thickness of cutoff lines, default is NULL.

`colors_use` vector of colors to use for plotting by identity.

`pt.size` Adjust point size for plotting.

`group.by` Name of one or more metadata columns to group (color) cells by (for example, orig.ident). Default is `@active.ident`.

`raster` Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 cells.

`raster.dpi` Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).

`assay` Name of assay to use, defaults to the active assay.

`ggplot_default_colors`
logical. If `colors_use = NULL`, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.

`color_seed` random seed for the "varibow" palette shuffle if `colors_use = NULL` and number of groups plotted is greater than 36. Default = 123.

`shuffle_seed` Sets the seed if randomly shuffling the order of points (Default is 1).

`...` Extra parameters passed to `FeatureScatter`.

Value

A ggplot object

Examples

```
## Not run:
QC_Plot_GenevsFeature(seurat_object = obj, y_axis_label = "Feature per Cell")

## End(Not run)
```

QC_Plot_UMIvsFeature *QC Plots UMI vs Misc*

Description

Custom FeatureScatter for initial QC checks including lines for thresholding

Usage

```
QC_Plot_UMIvsFeature(
  seurat_object,
  feature1,
  x_axis_label = NULL,
  y_axis_label = "UMIs per Cell/Nucleus",
  low_cutoff_UMI = NULL,
  high_cutoff_UMI = NULL,
  low_cutoff_feature = NULL,
  high_cutoff_feature = NULL,
  cutoff_line_width = NULL,
  colors_use = NULL,
  pt.size = 1,
  group.by = NULL,
  raster = NULL,
  raster.dpi = c(512, 512),
  assay = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  shuffle_seed = 1,
  ...
)
```

Arguments

seurat_object	Seurat object name.
feature1	First feature to plot.
x_axis_label	Label for x axis.
y_axis_label	Label for y axis.
low_cutoff_UMI	Plot line a potential low threshold for filtering UMI per cell.
high_cutoff_UMI	Plot line a potential high threshold for filtering UMI per cell.
low_cutoff_feature	Plot line a potential low threshold for filtering feature1 per cell.
high_cutoff_feature	Plot line a potential high threshold for filtering feature1 per cell.

cutoff_line_width
: numerical value for thickness of cutoff lines, default is NULL.

colors_use
: vector of colors to use for plotting by identity.

pt.size
: Adjust point size for plotting.

group.by
: Name of one or more metadata columns to group (color) cells by (for example, orig.ident). Default is `@active.ident`.

raster
: Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 cells.

raster.dpi
: Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).

assay
: Name of assay to use, defaults to the active assay.

ggplot_default_colors
: logical. If `colors_use = NULL`, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.

color_seed
: random seed for the "varibow" palette shuffle if `colors_use = NULL` and number of groups plotted is greater than 36. Default = 123.

shuffle_seed
: Sets the seed if randomly shuffling the order of points (Default is 1).

...
: Extra parameters passed to `FeatureScatter`.

Value

A ggplot object

Examples

```
## Not run:
QC_Plot_UMIvsFeature(seurat_object = obj, y_axis_label = "Feature per Cell")

## End(Not run)
```

QC_Plot_UMIvsGene *QC Plots Genes vs UMIs*

Description

Custom FeatureScatter for initial QC checks including lines for thresholding

Usage

```
QC_Plot_UMIvsGene(
  seurat_object,
  x_axis_label = "UMIs per Cell/Nucleus",
  y_axis_label = "Genes per Cell/Nucleus",
  low_cutoff_gene = -Inf,
```

```
  high_cutoff_gene = Inf,
  low_cutoff_UMI = -Inf,
  high_cutoff_UMI = Inf,
  cutoff_line_width = NULL,
  colors_use = NULL,
  meta_gradient_name = NULL,
  meta_gradient_color = viridis_plasma_dark_high,
  meta_gradient_na_color = "lightgray",
  meta_gradient_low_cutoff = NULL,
  cells = NULL,
  combination = FALSE,
  ident_legend = TRUE,
  pt.size = 1,
  group.by = NULL,
  raster = NULL,
  raster.dpi = c(512, 512),
  assay = NULL,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  shuffle_seed = 1,
  ...
)
```

Arguments

seurat_object	Seurat object name.
x_axis_label	Label for x axis.
y_axis_label	Label for y axis.
low_cutoff_gene	Plot line a potential low threshold for filtering genes per cell.
high_cutoff_gene	Plot line a potential high threshold for filtering genes per cell.
low_cutoff_UMI	Plot line a potential low threshold for filtering UMIs per cell.
high_cutoff_UMI	Plot line a potential high threshold for filtering UMIs per cell.
cutoff_line_width	numerical value for thickness of cutoff lines, default is NULL.
colors_use	vector of colors to use for plotting by identity.
meta_gradient_name	Name of continuous meta data variable to color points in plot by. (MUST be continuous variable i.e. "percent_mito").
meta_gradient_color	The gradient color palette to use for plotting of meta variable (default is viridis "Plasma" palette with dark colors high).
meta_gradient_na_color	Color to use for plotting values when a `meta_gradient_low_cutoff` is set (default is "lightgray").

meta_gradient_low_cutoff
: Value to use as threshold for plotting. meta_gradient_name values below this value will be plotted using meta_gradient_na_color.

cells
: Cells to include on the scatter plot (default is all cells).

combination
: logical (default FALSE). Whether or not to return a plot layout with both the plot colored by identity and the meta data gradient plot.

ident_legend
: logical, whether to plot the legend containing identities (left plot) when combination = TRUE. Default is TRUE.

pt.size
: Passes size of points to both FeatureScatter and geom_point.

group.by
: Name of one or more metadata columns to group (color) cells by (for example, orig.ident). Default is @active.ident.

raster
: Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 cells.

raster.dpi
: Pixel resolution for rasterized plots, passed to geom_scattermore(). Default is c(512, 512).

assay
: Name of assay to use, defaults to the active assay.

ggplot_default_colors
: logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.

color_seed
: Random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.

shuffle_seed
: Sets the seed if randomly shuffling the order of points (Default is 1).

...
: Extra parameters passed to FeatureScatter.

Value

A ggplot object

Examples

```
library(Seurat)
QC_Plot_UMIvsGene(seurat_object = pbmc_small, x_axis_label = "UMIs per Cell/Nucleus",
y_axis_label = "Genes per Cell/Nucleus")
```

Random_Cells_Downsample

Randomly downsample by identity

Description

Get a randomly downsampled set of cell barcodes with even numbers of cells for each identity class. Can return either as a list (1 entry per identity class) or vector of barcodes.

Usage

```
Random_Cells_Downsample(
  seurat_object,
  num_cells,
  group.by = NULL,
  return_list = FALSE,
  allow_lower = FALSE,
  seed = 123
)
```

Arguments

seurat_object	Seurat object
num_cells	number of cells per ident to use in down-sampling. This value must be less than or equal to the size of ident with fewest cells. Alternatively, can set to "min" which will use the maximum number of barcodes based on size of smallest group.
group.by	The ident to use to group cells. Default is NULL which use current active.ident. .
return_list	logical, whether or not to return the results as list instead of vector, default is FALSE.
allow_lower	logical, if number of cells in identity is lower than num_cells keep the maximum number of cells, default is FALSE. If FALSE will report error message if num_cells is too high, if TRUE will subset cells with more than num_cells to that value and those with less than num_cells will not be downsampled.
seed	random seed to use for downsampling. Default is 123.

Value

either a vector or list of cell barcodes

Examples

```
library(Seurat)

# return vector of barcodes
random_cells <- Random_Cells_Downsample(seurat_object = pbmc_small, num_cells = 10)
head(random_cells)

# return list
random_cells_list <- Random_Cells_Downsample(seurat_object = pbmc_small, return_list = TRUE,
num_cells = 10)
head(random_cells_list)

# return max total number of cells (setting `num_cells = "min`)
random_cells_max <- Random_Cells_Downsample(seurat_object = pbmc_small, num_cells = "min")
```

Read10X_GEO *Load in NCBI GEO data from 10X*

Description

Enables easy loading of sparse data matrices provided by 10X genomics. That have file prefixes added to them by NCBI GEO or other repos.

Usage

```
Read10X_GEO(
  data_dir = NULL,
  sample_list = NULL,
  sample_names = NULL,
  gene.column = 2,
  cell.column = 1,
  unique.features = TRUE,
  strip.suffix = FALSE,
  parallel = FALSE,
  num_cores = NULL,
  merge = FALSE
)
```

Arguments

data_dir	Directory containing the matrix.mtx, genes.tsv (or features.tsv), and barcodes.tsv files provided by 10X.
sample_list	A vector of file prefixes/names if specific samples are desired. Default is NULL and will load all samples in given directory.
sample_names	a set of sample names to use for each sample entry in returned list. If NULL will set names to the file name of each sample.
gene.column	Specify which column of genes.tsv or features.tsv to use for gene names; default is 2.
cell.column	Specify which column of barcodes.tsv to use for cell names; default is 1.
unique.features	Make feature names unique (default TRUE).
strip.suffix	Remove trailing "-1" if present in all cell barcodes.
parallel	logical (default FALSE). Whether to use multiple cores when reading in data. Only possible on Linux based systems.
num_cores	if parallel = TRUE indicates the number of cores to use for multicore processing.
merge	logical (default FALSE) whether or not to merge samples into a single matrix or return list of matrices. If TRUE each sample entry in list will have cell barcode prefix added. The prefix will be taken from sample_names.

Value

If features.csv indicates the data has multiple data types, a list containing a sparse matrix of the data from each type will be returned. Otherwise a sparse matrix containing the expression data will be returned.

References

Code used in function has been slightly modified from `Seurat::Read10X` function of Seurat package https://github.com/satijalab/seurat (License: GPL-3). Function was modified to support file prefixes and altered loop by Samuel Marsh for scCustomize (also previously posted as potential PR to Seurat GitHub).

Examples

```
## Not run:
data_dir <- 'path/to/data/directory'
expression_matrices <- Read10X_GEO(data_dir = data_dir)
# To create object from single file
seurat_object = CreateSeuratObject(counts = expression_matrices[[1]])

## End(Not run)
```

Read10X_h5_GEO *Load in NCBI GEO data from 10X in HDF5 file format*

Description

Enables easy loading of HDF5 data matrices provided by 10X genomics. That have file prefixes added to them by NCBI GEO or other repos or programs (i.e. Cell Bender)

Usage

```
Read10X_h5_GEO(
  data_dir = NULL,
  sample_list = NULL,
  sample_names = NULL,
  shared_suffix = NULL,
  parallel = FALSE,
  num_cores = NULL,
  merge = FALSE,
  ...
)
```

Arguments

`data_dir`	Directory containing the .h5 files provided by 10X.
`sample_list`	A vector of file prefixes/names if specific samples are desired. Default is `NULL` and will load all samples in given directory.
`sample_names`	a set of sample names to use for each sample entry in returned list. If `NULL` will set names to the file name of each sample.
`shared_suffix`	a suffix and file extension shared by all samples.
`parallel`	logical (default FALSE). Whether to use multiple cores when reading in data. Only possible on Linux based systems.
`num_cores`	if `parallel = TRUE` indicates the number of cores to use for multicore processing.
`merge`	logical (default FALSE) whether or not to merge samples into a single matrix or return list of matrices. If TRUE each sample entry in list will have cell barcode prefix added. The prefix will be taken from `sample_names`.
`...`	Additional arguments passed to `Read10X_h5`

Value

If the data has multiple data types, a list containing a sparse matrix of the data from each type will be returned. Otherwise a sparse matrix containing the expression data will be returned.

Examples

```
## Not run:
data_dir <- 'path/to/data/directory'
expression_matrices <- Read10X_h5_GEO(data_dir = data_dir)
# To create object from single file
seurat_object = CreateSeuratObject(counts = expression_matrices[[1]])

## End(Not run)
```

`Read10X_h5_Multi_Directory`

Load 10X h5 count matrices from multiple directories

Description

Enables easy loading of sparse data matrices provided by 10X genomics that are present in multiple subdirectories. Can function with either default output directory structure of Cell Ranger or custom directory structure.

Usage

```
Read10X_h5_Multi_Directory(
  base_path,
  secondary_path = NULL,
  default_10X_path = TRUE,
  cellranger_multi = FALSE,
  h5_filename = "filtered_feature_bc_matrix.h5",
  sample_list = NULL,
  sample_names = NULL,
  replace_suffix = FALSE,
  new_suffix_list = NULL,
  parallel = FALSE,
  num_cores = NULL,
  merge = FALSE,
  ...
)
```

Arguments

base_path	path to the parent directory which contains all of the subdirectories of interest.
secondary_path	path from the parent directory to count matrix files for each sample.
default_10X_path	logical (default TRUE) sets the secondary path variable to the default 10X directory structure.
cellranger_multi	logical, whether samples were processed with Cell Ranger `multi`, default is FALSE.
h5_filename	name of h5 file (including .h5 suffix). If all h5 files have same name (i.e. Cell Ranger output) then use full file name. By default function uses Cell Ranger name: "filtered_feature_bc_matrix.h5". If h5 files have sample specific prefixes (i.e. from Cell Bender) then use only the shared part of file name (e.g., "_filtered_out.h5").
sample_list	a vector of sample directory names if only specific samples are desired. If `NULL` will read in subdirectories in parent directory.
sample_names	a set of sample names to use for each sample entry in returned list. If `NULL` will set names to the subdirectory name of each sample.
replace_suffix	logical (default FALSE). Whether or not to replace the barcode suffixes of matrices using `Replace_Suffix`.
new_suffix_list	a vector of new suffixes to replace existing suffixes if `replace_suffix = TRUE`. See `Replace_Suffix` for more information. To remove all suffixes set `new_suffix_list = ""`.
parallel	logical (default FALSE) whether or not to use multi core processing to read in matrices.
num_cores	how many cores to use for parallel processing.

merge	logical (default FALSE) whether or not to merge samples into a single matrix or return list of matrices. If TRUE each sample entry in list will have cell barcode prefix added. The prefix will be taken from sample_names.
...	Extra parameters passed to Read10X_h5.

Value

a list of sparse matrices (merge = FALSE) or a single sparse matrix (merge = TRUE).

Examples

```
## Not run:
base_path <- 'path/to/data/directory'
expression_matrices <- Read10X_h5_Multi_Directory(base_path = base_path)

## End(Not run)
```

Read10X_Multi_Directory

Load 10X count matrices from multiple directories

Description

Enables easy loading of sparse data matrices provided by 10X genomics that are present in multiple subdirectories. Can function with either default output directory structure of Cell Ranger or custom directory structure.

Usage

```
Read10X_Multi_Directory(
  base_path,
  secondary_path = NULL,
  default_10X_path = TRUE,
  cellranger_multi = FALSE,
  sample_list = NULL,
  sample_names = NULL,
  parallel = FALSE,
  num_cores = NULL,
  merge = FALSE,
  ...
)
```

Arguments

base_path	path to the parent directory which contains all of the subdirectories of interest.
secondary_path	path from the parent directory to count matrix files for each sample.
default_10X_path	logical (default TRUE) sets the secondary path variable to the default 10X directory structure.
cellranger_multi	logical, whether samples were processed with Cell Ranger multi, default is FALSE.
sample_list	a vector of sample directory names if only specific samples are desired. If NULL will read in subdirectories in parent directory.
sample_names	a set of sample names to use for each sample entry in returned list. If NULL will set names to the subdirectory name of each sample.
parallel	logical (default FALSE) whether or not to use multi core processing to read in matrices.
num_cores	how many cores to use for parallel processing.
merge	logical (default FALSE) whether or not to merge samples into a single matrix or return list of matrices. If TRUE each sample entry in list will have cell barcode prefix added. The prefix will be taken from sample_names.
...	Extra parameters passed to Read10X.

Value

a list of sparse matrices (merge = FALSE) or a single sparse matrix (merge = TRUE).

Examples

```
## Not run:
base_path <- 'path/to/data/directory'
expression_matrices <- Read10X_Multi_Directory(base_path = base_path)

## End(Not run)
```

Read_Add_cNMF *Read and add results from cNMF*

Description

Reads the usage and spectra files from cNMF results and adds them as dimensionality reduction to seurat object.

Usage

```
Read_Add_cNMF(
  seurat_object,
  usage_file,
  spectra_file,
  normalize = TRUE,
  assay = NULL
)
```

Arguments

seurat_object	Seurat object name to add cNMF reduction
usage_file	path and name of cNMF usage file
spectra_file	path and name of cNMF spectra file
normalize	logical, whether to normalize the cNMF usage data, default is TRUE
assay	assay to add reduction. Default is NULL and will use current active assay.

Value

Seurat object with new dimensionality reduction "cnmf"

References

For more information about cNMF and usage see https://github.com/dylkot/cNMF

Examples

```
## Not run:
object <- Read_cNMF(seurat_object = object,
usage_file = "example_cNMF/example_cNMF.usages.k_27.dt_0_01.consensus.txt",
spectra_file = "example_cNMF/example_cNMF.gene_spectra_score.k_27.dt_0_01.txt")

## End(Not run)
```

Read_CellBender_h5_Mat

Load CellBender h5 matrices (corrected)

Description

Extract sparse matrix with corrected counts from CellBender h5 output file.

Usage

```
Read_CellBender_h5_Mat(
  file_name,
  use.names = TRUE,
  unique.features = TRUE,
  h5_group_name = NULL,
  feature_slot_name = "features"
)
```

Arguments

file_name	Path to h5 file.
use.names	Label row names with feature names rather than ID numbers (default TRUE).
unique.features	Make feature names unique (default TRUE).
h5_group_name	Name of the group within H5 file that contains count data. This is only required if H5 file contains multiple subgroups and non-default names. Default is NULL.
feature_slot_name	Name of the slot contain feature names/ids. Must be one of: "features"(Cell Ranger v3+) or "genes" (Cell Ranger v1/v2 or STARsolo). Default is "features".

Value

sparse matrix

References

Code used in function has been modified from Seurat::Read10X_h5 function of Seurat package https://github.com/satijalab/seurat (License: GPL-3).

Examples

```
## Not run:
mat <- Read_CellBender_h5_Mat(file_name = "/SampleA_out_filtered.h5")

## End(Not run)
```

Read_CellBender_h5_Multi_Directory

Load CellBender h5 matrices (corrected) from multiple directories

---

Description

Extract sparse matrix with corrected counts from CellBender h5 output file across multiple sample subdirectories.

Usage

```
Read_CellBender_h5_Multi_Directory(
  base_path,
  secondary_path = NULL,
  filtered_h5 = TRUE,
  custom_name = NULL,
  sample_list = NULL,
  sample_names = NULL,
  h5_group_name = NULL,
  feature_slot_name = "features",
  replace_suffix = FALSE,
  new_suffix_list = NULL,
  parallel = FALSE,
  num_cores = NULL,
  merge = FALSE,
  ...
)
```

Arguments

`base_path`	path to the parent directory which contains all of the subdirectories of interest.
`secondary_path`	path from the parent directory to count matrix files for each sample.
`filtered_h5`	logical (default TRUE). Will set the shared file name suffix `custom_name` is NULL.
`custom_name`	if file name was customized in CellBender then this parameter should contain the portion of file name that is shared across all samples. Must included the ".h5" extension as well.
`sample_list`	a vector of sample directory names if only specific samples are desired. If `NULL` will read in subdirectories in parent directory.
`sample_names`	a set of sample names to use for each sample entry in returned list. If `NULL` will set names to the subdirectory name of each sample. NOTE: unless `sample_list` is specified this will rename files in the order they are read which will be alphabetical.
`h5_group_name`	Name of the group within H5 file that contains count data. This is only required if H5 file contains multiple subgroups and non-default names. Default is `NULL`.
`feature_slot_name`	Name of the slot contain feature names/ids. Must be one of: "features"(Cell Ranger v3+) or "genes" (Cell Ranger v1/v2 or STARsolo). Default is "features".
`replace_suffix`	logical (default FALSE). Whether or not to replace the barcode suffixes of matrices using `Replace_Suffix`.
`new_suffix_list`	a vector of new suffixes to replace existing suffixes if `replace_suffix = TRUE`. See `Replace_Suffix` for more information. To remove all suffixes set `new_suffix_list = ""`.
`parallel`	logical (default FALSE) whether or not to use multi core processing to read in matrices.

num_cores	how many cores to use for parallel processing.
merge	logical (default FALSE) whether or not to merge samples into a single matrix or return list of matrices. If TRUE each sample entry in list will have cell barcode prefix added. The prefix will be taken from sample_names.
...	Extra parameters passed to Read_CellBender_h5_Mat.

Value

list of sparse matrices

Examples

```
## Not run:
base_path <- 'path/to/data/directory'
mat_list <- Read_CellBender_h5_Multi_Directory(base_path = base_path)

## End(Not run)
```

Read_CellBender_h5_Multi_File

Load CellBender h5 matrices (corrected) from multiple files

Description

Extract sparse matrix with corrected counts from CellBender h5 output file across multiple samples within the same directory.

Usage

```
Read_CellBender_h5_Multi_File(
  data_dir = NULL,
  filtered_h5 = TRUE,
  custom_name = NULL,
  sample_list = NULL,
  sample_names = NULL,
  h5_group_name = NULL,
  feature_slot_name = "features",
  parallel = FALSE,
  num_cores = NULL,
  merge = FALSE,
  ...
)
```

Arguments

data_dir	Directory containing the .h5 files output by CellBender.
filtered_h5	logical (default TRUE). Will set the shared file name suffix if custom_name is NULL.
custom_name	if file name was customized in CellBender then this parameter should contain the portion of file name that is shared across all samples. Must included the ".h5" extension as well.
sample_list	a vector of sample names if only specific samples are desired. If NULL will read in all files within data_dir directory.
sample_names	a set of sample names to use for each sample entry in returned list. If NULL will set names to the subdirectory name of each sample.
h5_group_name	Name of the group within H5 file that contains count data. This is only required if H5 file contains multiple subgroups and non-default names. Default is NULL.
feature_slot_name	Name of the slot contain feature names/ids. Must be one of: "features"(Cell Ranger v3+) or "genes" (Cell Ranger v1/v2 or STARsolo). Default is "features".
parallel	logical (default FALSE) whether or not to use multi core processing to read in matrices
num_cores	how many cores to use for parallel processing.
merge	logical (default FALSE) whether or not to merge samples into a single matrix or return list of matrices. If TRUE each sample entry in list will have cell barcode prefix added. The prefix will be taken from sample_names.
...	Extra parameters passed to Read_CellBender_h5_Mat.

Value

list of sparse matrices

Examples

```
## Not run:
base_path <- 'path/to/data/directory'
mat_list <- Read_CellBender_h5_Multi_File(data_dir = base_path)

## End(Not run)
```

Read_GEO_Delim *Load in NCBI GEO data formatted as single file per sample*

Description

Can read delimited file types (i.e. csv, tsv, txt)

Usage

```
Read_GEO_Delim(
  data_dir,
  file_suffix,
  move_genes_rownames = TRUE,
  sample_list = NULL,
  full_names = FALSE,
  sample_names = NULL,
  barcode_suffix_period = FALSE,
  parallel = FALSE,
  num_cores = NULL,
  merge = FALSE
)
```

Arguments

data_dir	Directory containing the files.
file_suffix	The file suffix of the individual files. Must be the same across all files being imported. This is used to detect files to import and their GEO IDs.
move_genes_rownames	logical. Whether gene IDs are present in first column or in row names of delimited file. If TRUE will move the first column to row names before creating final matrix. Default is TRUE.
sample_list	a vector of samples within directory to read in (can be either with or without file_suffix see full_names). If NULL will read in all subdirectories.
full_names	logical (default FALSE). Whether or not the sample_list vector includes the file suffix. If FALSE the function will add suffix based on file_suffix parameter.
sample_names	a set of sample names to use for each sample entry in returned list. If NULL will set names to the directory name of each sample.
barcode_suffix_period	Is the barcode suffix a period and should it be changed to "-". Default (FALSE; barcodes will be left identical to their format in input files.). If TRUE "." in barcode suffix will be changed to "-".
parallel	logical (default FALSE). Whether to use multiple cores when reading in data. Only possible on Linux based systems.
num_cores	if parallel = TRUE indicates the number of cores to use for multicore processing.
merge	logical (default FALSE) whether or not to merge samples into a single matrix or return list of matrices. If TRUE each sample entry in list will have cell barcode prefix added. The prefix will be taken from sample_names.

Value

List of gene x cell matrices in list format named by sample name.

Examples

```
## Not run:
data_dir <- 'path/to/data/directory'
expression_matrices <- Read_GEO_Delim(data_dir = data_dir)

## End(Not run)
```

`Read_Metrics_10X` *Read Overall Statistics from 10X Cell Ranger Count*

Description

Get data.frame with all metrics from the Cell Ranger count analysis (present in web_summary.html)

Usage

```
Read_Metrics_10X(
  base_path,
  secondary_path = NULL,
  default_10X = TRUE,
  cellranger_multi = FALSE,
  lib_list = NULL,
  lib_names = NULL
)
```

Arguments

`base_path`	path to the parent directory which contains all of the sub-directories of interest or alternatively can provide single csv file to read and format identically to reading multiple files.
`secondary_path`	path from the parent directory to count "outs/" folder which contains the "metrics_summary.csv" file.
`default_10X`	logical (default TRUE) sets the secondary path variable to the default 10X directory structure.
`cellranger_multi`	logical, whether or not metrics come from Cell Ranger `count` or from Cell Ranger `multi`. Default is FALSE.
`lib_list`	a list of sample names (matching directory names) to import. If `NULL` will read in all samples in parent directory.
`lib_names`	a set of sample names to use for each sample. If `NULL` will set names to the directory name of each sample.

Value

A data frame or list of data.frames with sample metrics from cell ranger.

Examples

```
## Not run:
metrics <- Read_Metrics_10X(base_path = "/path/to/directories", default_10X = TRUE)

## End(Not run)
```

`Read_Metrics_CellBender`

Read Overall Statistics from CellBender

Description

Get data.frame with all metrics from the CellBender `remove-background` analysis.

Usage

```
Read_Metrics_CellBender(base_path, lib_list = NULL, lib_names = NULL)
```

Arguments

`base_path`	path to the parent directory which contains all of the sub-directories of interest or path to single metrics csv file.
`lib_list`	a list of sample names (matching directory names) to import. If `NULL` will read in all samples in parent directory.
`lib_names`	a set of sample names to use for each sample. If `NULL` will set names to the directory name of each sample.

Value

A data frame with sample metrics from CellBender.

Examples

```
## Not run:
CB_metrics <- Read_Metrics_CellBender(base_path = "/path/to/directories")

## End(Not run)
```

Reduction_Loading_Present *Check if reduction loadings are present*

Description

Check if reduction loadings are present in object and return vector of found loading names. Return warning messages for genes not found.

Usage

```
Reduction_Loading_Present(
  seurat_object,
  reduction_names,
  print_msg = TRUE,
  omit_warn = TRUE,
  return_none = FALSE
)
```

Arguments

seurat_object	object name.
reduction_names	vector of genes to check.
print_msg	logical. Whether message should be printed if all features are found. Default is TRUE.
omit_warn	logical. Whether to print message about features that are not found in current object. Default is TRUE.
return_none	logical. Whether list of found vs. bad features should still be returned if no features are found. Default is FALSE.

Value

A list of length 3 containing 1) found features, 2) not found features.

Examples

```
## Not run:
reductions <- Reduction_Loading_Present(seurat_object = obj_name, reduction_name = "PC_1")
found_reductions <- reductions[[1]]

## End(Not run)
```

Rename_Clusters *Rename Clusters*

Description

Wrapper function to rename active cluster identity in Seurat or Liger Object with new idents.

Usage

```
Rename_Clusters(object, ...)

## S3 method for class 'liger'
Rename_Clusters(
  object,
  new_idents,
  old_ident_name = NULL,
  new_ident_name = NULL,
  overwrite = FALSE,
  ...
)

## S3 method for class 'Seurat'
Rename_Clusters(
  object,
  new_idents,
  old_ident_name = NULL,
  new_ident_name = NULL,
  meta_col_name = deprecated(),
  overwrite = FALSE,
  ...
)
```

Arguments

object	Object of class Seurat or liger.
...	Arguments passed to other methods
new_idents	vector of new cluster names. Must be equal to the length of current default identity of Object. Will accept named vector (with old idents as names) or will name the new_idents vector internally.
old_ident_name	optional, name to use for storing current object idents in object meta data slot.
new_ident_name	optional, name to use for storing new object idents in object meta data slot.
overwrite	logical, whether to overwrite columns in object meta data slot. if they have same names as old_ident_name and/or new_ident_name.
meta_col_name	**[Soft-deprecated]**. See old_ident_name.

Value

An object of the same class as `object` with updated default identities.

Examples

```
## Not run:
# Liger version
obj <- Rename_Clusters(object = obj_name, new_idents = new_idents_vec,
old_ident_name = "LIGER_Idents_Round01", new_ident_name = "LIGER_Idents_Round02")

## End(Not run)

## Not run:
obj <- Rename_Clusters(seurat_object = obj_name, new_idents = new_idents_vec,
old_ident_name = "Seurat_Idents_Round01", new_ident_name = "Round01_Res0.6_Idents")

## End(Not run)
```

`Replace_Suffix` *Replace barcode suffixes*

Description

Replace barcode suffixes in matrix, data.frame, or list of matrices/data.frames

Usage

```
Replace_Suffix(data, current_suffix, new_suffix)
```

Arguments

`data`	Either matrix/data.frame or list of matrices/data.frames with the cell barcodes in the column names.
`current_suffix`	a single value or vector of values representing current barcode suffix. If suffix is the same for all matrices/data.frames in list only single value is required.
`new_suffix`	a single value or vector of values representing new barcode suffix to be added. If desired suffix is the same for all matrices/data.frames in list only single value is required. If no suffix is desired set `new_suffix = ""`.‘

Value

matrix or data.frame with new column names.

Examples

```
## Not run:
dge_matrix <- Replace_Suffix(data = dge_matrix, current_suffix = "-1", new_suffix = "-2")

## End(Not run)
```

---

scCustomize_Palette *Color Palette Selection for scCustomize*

Description

Function to return package default discrete palettes depending on number of groups plotted.

Usage

```
scCustomize_Palette(
  num_groups,
  ggplot_default_colors = FALSE,
  color_seed = 123
)
```

Arguments

num_groups	number of groups to be plotted. If ggplot_default_colors = FALSE then by default: • If number of levels plotted equal to 2 then colors will be NavyAndOrange(). • If number of levels plotted greater than 2 but less than or equal to 36 it will use "polychrome" from DiscretePalette_scCustomize(). • If greater than 36 will use "varibow" with shuffle = TRUE from DiscretePalette_scCustomize.
ggplot_default_colors	logical. Whether to use default ggplot hue palette or not.
color_seed	random seed to use for shuffling the "varibow" palette.

Value

vector of colors to use for plotting.

Examples

```
cols <- scCustomize_Palette(num_groups = 24, ggplot_default_colors = FALSE)
PalettePlot(pal= cols)
```

`Seq_QC_Plot_Alignment_Combined` *QC Plots Sequencing metrics (Alignment) (Layout)*

Description

Plot a combined plot of the Alignment QC metrics from sequencing output.

Usage

```
Seq_QC_Plot_Alignment_Combined(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  patchwork_title = "Sequencing QC Plots: Read Alignment Metrics",
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`patchwork_title`	Title to use for the patchworked plot output.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Alignment_Combined(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Antisense` *QC Plots Sequencing metrics (Alignment)*

Description

Plot the fraction of reads mapped Antisense to Gene

Usage

```
Seq_QC_Plot_Antisense(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Antisense(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Basic_Combined`

QC Plots Sequencing metrics (Layout)

Description

Plot a combined plot of the basic QC metrics from sequencing output.

Usage

```
Seq_QC_Plot_Basic_Combined(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  patchwork_title = "Sequencing QC Plots: Basic Cell Metrics",
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`patchwork_title`	Title to use for the patchworked plot output.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Basic_Combined(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Exonic` *QC Plots Sequencing metrics (Alignment)*

Description

Plot the fraction of reads confidently mapped to Exonic regions

Usage

```
Seq_QC_Plot_Exonic(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Exonic(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Genes` *QC Plots Sequencing metrics*

Description

Plot the median genes per cell per sample

Usage

```
Seq_QC_Plot_Genes(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Genes(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Genome` *QC Plots Sequencing metrics (Alignment)*

Description

Plot the fraction of reads confidently mapped to genome

Usage

```
Seq_QC_Plot_Genome(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Genome(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Intergenic`

QC Plots Sequencing metrics (Alignment)

Description

Plot the fraction of reads confidently mapped to intergenic regions

Usage

```
Seq_QC_Plot_Intergenic(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Intergeneic(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Intronic` *QC Plots Sequencing metrics (Alignment)*

Description

Plot the fraction of reads confidently mapped to intronic regions

Usage

```
Seq_QC_Plot_Intronic(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Intronic(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Number_Cells` *QC Plots Sequencing metrics*

Description

Plot the number of cells per sample

Usage

```
Seq_QC_Plot_Number_Cells(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Number_Cells(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Reads_in_Cells` *QC Plots Sequencing metrics*

Description

Plot the fraction of reads in cells per sample

Usage

```
Seq_QC_Plot_Reads_in_Cells(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Reads_in_Cells(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Reads_per_Cell`
QC Plots Sequencing metrics

Description

Plot the mean number of reads per cell

Usage

```
Seq_QC_Plot_Reads_per_Cell(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Reads_per_Cell(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Saturation` *QC Plots Sequencing metrics*

Description

Plot the sequencing saturation percentage per sample

Usage

```
Seq_QC_Plot_Saturation(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Saturation(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Total_Genes`

QC Plots Sequencing metrics

Description

Plot the total genes detected per sample

Usage

```
Seq_QC_Plot_Total_Genes(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Total_Genes(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_Transcriptome`
QC Plots Sequencing metrics (Alignment)

Description

Plot the fraction of reads confidently mapped to transcriptome

Usage

```
Seq_QC_Plot_Transcriptome(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_Transcriptome(metrics_dataframe = metrics)

## End(Not run)
```

`Seq_QC_Plot_UMIs` *QC Plots Sequencing metrics*

Description

Plot the median UMIs per cell per sample

Usage

```
Seq_QC_Plot_UMIs(
  metrics_dataframe,
  plot_by = "sample_id",
  colors_use = NULL,
  dot_size = 1,
  x_lab_rotate = FALSE,
  significance = FALSE,
  ...
)
```

Arguments

`metrics_dataframe`	data.frame contain Cell Ranger QC Metrics (see `Read_Metrics_10X`).
`plot_by`	Grouping factor for the plot. Default is to plot as single group with single point per sample.
`colors_use`	colors to use for plot if plotting by group. Defaults to RColorBrewer Dark2 palette if less than 8 groups and `DiscretePalette_scCustomize(palette = "polychrome")` if more than 8.
`dot_size`	size of the dots plotted if `plot_by` is not `sample_id` Default is 1.
`x_lab_rotate`	logical. Whether to rotate the axes labels on the x-axis. Default is FALSE.
`significance`	logical. Whether to calculate and plot p-value comparisons when plotting by grouping factor. Default is FALSE.
`...`	Other variables to pass to `ggpubr::stat_compare_means` when doing significance testing.

Value

A ggplot object

Examples

```
## Not run:
Seq_QC_Plot_UMIs(metrics_dataframe = metrics)

## End(Not run)
```

`seq_zeros` *Create sequence with zeros*

Description

Create sequences of numbers like `seq()` or `seq_len()` but with zeros prefixed to keep numerical order

Usage

```
seq_zeros(seq_length, num_zeros = NULL)
```

Arguments

`seq_length`	a seqeunce or numbers of numbers to create sequence. Users can provide sequence (1:XX) or number of values to add in sequence (will be used as second number in `seq_len`; 1:XX).
`num_zeros`	number of zeros to prefix sequence, default is (e.g, 01, 02, 03, ...)

Value

vector of numbers in sequence

References

Base code from stackoverflow post: https://stackoverflow.com/a/38825614

Examples

```
# Using sequence
new_seq <- seq_zeros(seq_length = 1:15, num_zeros = 1)
new_seq

# Using number
new_seq <- seq_zeros(seq_length = 15, num_zeros = 1)
new_seq
```

```
# Sequence with 2 zeros
new_seq <- seq_zeros(seq_length = 1:15, num_zeros = 2)
new_seq
```

`Setup_scRNAseq_Project` *Setup project directory structure*

Description

Create reproducible project directory organization when initiating a new analysis.

Usage

```
Setup_scRNAseq_Project(
  custom_dir_file = NULL,
  cluster_annotation_path = NULL,
  cluster_annotation_file_name = "cluster_annotation.csv"
)
```

Arguments

`custom_dir_file`
: file to file containing desired directory structure. Default is NULL and will provide generic built-in directory structure.

`cluster_annotation_path`
: path to place cluster annotation file using `Create_Cluster_Annotation_File`.

`cluster_annotation_file_name`
: name to use for annotation file if created (optional).

Value

no return value. Creates system folders.

Examples

```
## Not run:
# If using built-in directory structure.
Setup_scRNAseq_Project()

## End(Not run)
```

Single_Color_Palette *Single Color Palettes for Plotting*

Description

Selects colors from modified versions of RColorBrewer single colors palettes

Usage

```
Single_Color_Palette(pal_color, num_colors = NULL, seed_use = 123)
```

Arguments

pal_color	color palette to select (Options are: 'reds', 'blues', 'greens', 'purples', 'oranges', 'grays').
num_colors	set number of colors (max = 7).
seed_use	set seed for reproducibility (default: 123).

Value

A vector of colors

References

See RColorBrewer for more info on palettes https://CRAN.R-project.org/package=RColorBrewer

Examples

```
pal <- Single_Color_Palette(pal_color = "reds", num_colors = 7)
PalettePlot(pal= pal)
```

SpatialDimPlot_scCustom *SpatialDimPlot with modified default settings*

Description

Creates SpatialDimPlot with some of the settings modified from their Seurat defaults (colors_use).

Usage

```
SpatialDimPlot_scCustom(
  seurat_object,
  group.by = NULL,
  images = NULL,
  colors_use = NULL,
  crop = TRUE,
  label = FALSE,
  label.size = 7,
  label.color = "white",
  label.box = TRUE,
  repel = FALSE,
  ncol = NULL,
  pt.size.factor = 1.6,
  alpha = c(1, 1),
  image.alpha = 1,
  stroke = 0.25,
  interactive = FALSE,
  combine = TRUE,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
group.by	Name of meta.data column to group the data by
images	Name of the images to use in the plot(s)
colors_use	color palette to use for plotting. By default if number of levels plotted is less than or equal to 36 it will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from DiscretePalette_scCustomize.
crop	Crop the plot in to focus on points plotted. Set to FALSE to show entire background image.
label	Whether to label the clusters
label.size	Sets the size of the labels
label.color	Sets the color of the label text
label.box	Whether to put a box around the label text (geom_text vs geom_label)
repel	Repels the labels to prevent overlap
ncol	Number of columns if plotting multiple plots
pt.size.factor	Scale the size of the spots.
alpha	Controls opacity of spots. Provide as a vector specifying the min and max for SpatialFeaturePlot. For SpatialDimPlot, provide a single alpha value for each plot.

image.alpha	Adjust the opacity of the background images. Set to 0 to remove.
stroke	Control the width of the border around the spots
interactive	Launch an interactive SpatialDimPlot or SpatialFeaturePlot session, see ISpatialDimPlot or ISpatialFeaturePlot for more details
combine	Combine plots into a single gg object; note that if TRUE; themeing will not work when plotting multiple features/groupings
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to DimPlot.

Value

A ggplot object

References

Many of the param names and descriptions are from Seurat to facilitate ease of use as this is simply a wrapper to alter some of the default parameters https://github.com/satijalab/seurat/blob/master/R/visualization.R (License: GPL-3).

Examples

```
## Not run:
SpatialDimPlot_scCustom(seurat_object = seurat_object)

## End(Not run)
```

Split_Layers *Split Seurat object into layers*

Description

Split Assay5 of Seurat object into layers by variable in meta.data

Usage

```
Split_Layers(seurat_object, assay = "RNA", split.by)
```

Arguments

seurat_object	Seurat object name.
assay	name(s) of assays to convert. Defaults to current active assay.
split.by	Variable in meta.data to use for splitting layers.

Examples

```
## Not run:
# Split object by "treatment"
obj <- Split_Layers(object = obj, assay = "RNA", split.by = "treatment")

## End(Not run)
```

`Split_Vector` *Split vector into list*

Description

Splits vector into chunks of x sizes

Usage

```
Split_Vector(x, chunk_size = NULL, num_chunk = NULL, verbose = FALSE)
```

Arguments

`x`	vector to split
`chunk_size`	size of chunks for vector to be split into, default is NULL. Only valid if `num_chunk` is NULL.
`num_chunk`	number of chunks to split the vector into, default is NULL. Only valid if `chunk_size` is NULL.
`verbose`	logical, print details of vector and split, default is FALSE.

Value

list with vector of X length

References

Base code from stackoverflow post: https://stackoverflow.com/a/3321659/15568251

Examples

```
vector <- c("gene1", "gene2", "gene3", "gene4", "gene5", "gene6")

vector_list <- Split_Vector(x = vector, chunk_size = 3)
```

`Stacked_VlnPlot` *Stacked Violin Plot*

Description

Code for creating stacked violin plot gene expression.

Usage

```
Stacked_VlnPlot(
  seurat_object,
  features,
  group.by = NULL,
  split.by = NULL,
  idents = NULL,
  x_lab_rotate = FALSE,
  plot_legend = FALSE,
  colors_use = NULL,
  color_seed = 123,
  ggplot_default_colors = FALSE,
  plot_spacing = 0.15,
  spacing_unit = "cm",
  vln_linewidth = NULL,
  pt.size = NULL,
  raster = NULL,
  add.noise = TRUE,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`features`	Features to plot.
`group.by`	Group (color) cells in different ways (for example, orig.ident).
`split.by`	A variable to split the violin plots by,
`idents`	Which classes to include in the plot (default is all).
`x_lab_rotate`	logical or numeric. If logical whether to rotate x-axis labels 45 degrees (Default is FALSE). If numeric must be either 45 or 90. Setting 45 is equivalent to setting TRUE.
`plot_legend`	logical. Adds plot legend containing `idents` to the returned plot.
`colors_use`	specify color palette to used in `VlnPlot`. By default if number of levels plotted is less than or equal to 36 it will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from `DiscretePalette_scCustomize`.
`color_seed`	random seed for the "varibow" palette shuffle if `colors_use` = `NULL` and number of groups plotted is greater than 36. Default = 123.

ggplot_default_colors
: logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.

plot_spacing
: Numerical value specifying the vertical spacing between each plot in the stack. Default is 0.15 ("cm"). Spacing dependent on unit provided to spacing_unit.

spacing_unit
: Unit to use in specifying vertical spacing between plots. Default is "cm".

vln_linewidth
: Adjust the linewidth of violin outline. Must be numeric.

pt.size
: Adjust point size for plotting. Default for Stacked_VlnPlot is 0 to avoid issues with rendering so many points in vector form. Alternatively, see raster parameter.

raster
: Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).

add.noise
: logical, determine if adding a small noise for plotting (Default is TRUE).

...
: Extra parameters passed to VlnPlot.

Value

A ggplot object

Author(s)

Ming Tang (Original Code), Sam Marsh (Wrap single function, added/modified functionality)

References

https://divingintogeneticsandgenomics.rbind.io/post/stacked-violin-plot-for-visualizing-single-cell

See Also

https://twitter.com/tangming2005

Examples

```
library(Seurat)
Stacked_VlnPlot(seurat_object = pbmc_small, features = c("CD3E", "CD8", "GZMB", "MS4A1"),
x_lab_rotate = TRUE)
```

Store_Misc_Info_Seurat *Store misc data in Seurat object*

Description

Wrapper function save variety of data types to the `object@misc` slot of Seurat object.

Usage

```
Store_Misc_Info_Seurat(
  seurat_object,
  data_to_store,
  data_name,
  list_as_list = FALSE,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`seurat_object`	object name.
`data_to_store`	data to be stored in `@misc` slot. Can be single piece of data or list. If list of data see `list_as_list` parameter for control over data storage.
`data_name`	name to give the entry in `@misc` slot. Must be of equal length of the number of data items being stored.
`list_as_list`	logical. If `data_to_store` is a list, this dictates whether to store in `@misc` slot as list (TRUE) or whether to store each entry in the list separately (FALSE). Default is FALSE.
`overwrite`	Logical. Whether to overwrite existing items with the same name. Default is FALSE, meaning that function will abort if item with `data_name` is present in misc slot.
`verbose`	logical, whether to print messages when running function, default is TRUE.

Value

Seurat Object with new entries in the `@misc` slot.

Examples

```
library(Seurat)
clu_pal <- c("red", "green", "blue")

pbmc_small <- Store_Misc_Info_Seurat(seurat_object = pbmc_small, data_to_store = clu_pal,
data_name = "rd1_colors")
```

`Store_Palette_Seurat` *Store color palette in Seurat object*

Description

Wrapper function around `Store_Misc_Info_Seurat` to store color palettes.

Usage

```
Store_Palette_Seurat(
  seurat_object,
  palette,
  palette_name,
  list_as_list = FALSE,
  overwrite = FALSE,
  verbose = TRUE
)
```

Arguments

`seurat_object`	object name.
`palette`	vector or list of vectors containing color palettes to store. If list of palettes see `list_as_list` parameter for control over data storage.
`palette_name`	name to give the palette(s) in `@misc` slot. Must be of equal length to the number of data items being stored.
`list_as_list`	logical. If `data_to_store` is a list, this dictates whether to store in `@misc` slot as list (TRUE) or whether to store each entry in the list separately (FALSE). Default is FALSE.
`overwrite`	Logical. Whether to overwrite existing items with the same name. Default is FALSE, meaning that function will abort if item with `data_name` is present in misc slot.
`verbose`	logical, whether to print messages when running function, default is TRUE.

Value

Seurat Object with new entries in the `@misc` slot.

Examples

```
library(Seurat)
clu_pal <- c("red", "green", "blue")

pbmc_small <- Store_Misc_Info_Seurat(seurat_object = pbmc_small, data_to_store = clu_pal,
data_name = "rd1_colors")
```

Subset_LIGER *Subset LIGER object*

Description

Subset LIGER object by cluster or other meta data variable.

Usage

```
Subset_LIGER(
  liger_object,
  cluster = NULL,
  cluster_col = "leiden_cluster",
  ident = NULL,
  ident_col = NULL,
  invert = FALSE
)
```

Arguments

liger_object	LIGER object name.
cluster	Name(s) of cluster to subset from object.
cluster_col	name of @cellMeta column containing cluster names, default is "leiden_cluster".
ident	variable within ident_col to use in sub-setting object.
ident_col	column in @cellMeta that contains values provided to ident.
invert	logical, whether to subset the inverse of the clusters or idents provided, default is FALSE.

Value

liger object

Examples

```
## Not run:
# subset clusters 3 and 5
sub_liger <- subset_liger(liger_object = liger_object, cluster = c(3, 5))

# subset control samples from column "Treatment"
sub_liger <- subset_liger(liger_object = liger_object, ident = "control",
ident_col = "Treatment")

# subset control samples from column "Treatment" in clusters 3 and 5
sub_liger <- subset_liger(liger_object = liger_object, ident = "control",
ident_col = "Treatment", cluster = c(3, 5))

# Remove cluster 9
```

```
sub_liger <- subset_liger(liger_object = liger_object, cluster = 9, invert = TRUE)

## End(Not run)
```

`theme_ggprism_mod` *Modified ggprism theme*

Description

Modified ggprism theme which restores the legend title.

Usage

```
theme_ggprism_mod(
  palette = "black_and_white",
  base_size = 14,
  base_family = "sans",
  base_fontface = "bold",
  base_line_size = base_size/20,
  base_rect_size = base_size/20,
  axis_text_angle = 0,
  border = FALSE
)
```

Arguments

`palette`	`string`. Palette name, use `names(ggprism_data$themes)` to show all valid palette names.
`base_size`	`numeric`. Base font size, given in `"pt"`.
`base_family`	`string`. Base font family, default is `"sans"`.
`base_fontface`	`string`. Base font face, default is `"bold"`.
`base_line_size`	`numeric`. Base linewidth for line elements
`base_rect_size`	`numeric`. Base linewidth for rect elements
`axis_text_angle`	`integer`. Angle of axis text in degrees. One of: `0, 45, 90, 270`.
`border`	`logical`. Should a border be drawn around the plot? Clipping will occur unless e.g. `coord_cartesian(clip = "off")` is used.

Value

Returns a list-like object of class *theme*.

References

theme is a modified version of theme_prism from ggprism package https://github.com/csdaw/ggprism (License: GPL-3). Param text is from ggprism::theme_prism() documentation theme_prism. Theme adaptation based on ggprism vignette https://csdaw.github.io/ggprism/articles/themes.html#make-your-own-ggprism-theme-1.

Examples

```
# Generate a plot and customize theme
library(ggplot2)
df <- data.frame(x = rnorm(n = 100, mean = 20, sd = 2), y = rbinom(n = 100, size = 100, prob = 0.2))
p <- ggplot(data = df, mapping = aes(x = x, y = y)) + geom_point(mapping = aes(color = 'red'))
p + theme_ggprism_mod()
```

Top_Genes_Factor *Extract top loading genes for LIGER factor*

Description

Extract vector to the top loading genes for specified LIGER iNMF factor

Usage

```
Top_Genes_Factor(object, factor = NULL, num_genes = 10, ...)

## S3 method for class 'liger'
Top_Genes_Factor(object, factor = NULL, num_genes = 10, ...)

## S3 method for class 'Seurat'
Top_Genes_Factor(object, factor = NULL, num_genes = 10, reduction, ...)
```

Arguments

object	object name.
factor	factor number to pull genes from. Set to "all" to return top loading genes from all factors
num_genes	number of top loading genes to return as vector, default is 10.
...	Arguments passed to other methods
reduction	name of reduction containing NMF/iNMF/cNMF data.

Value

vector of top genes for given factor or data.frame containing top genes across all factors

Examples

```
## Not run:
top_genes_factor10 <- Top_Genes_Factor(object = object, factor = 1, num_genes = 10)

## End(Not run)

## Not run:
top_genes_factor10 <- Top_Genes_Factor(object = object, factor = 1, num_genes = 10,
reduction = "cNMF")

## End(Not run)
```

UnRotate_X *Unrotate x axis on VlnPlot*

Description

Shortcut for thematic modification to unrotate the x axis (e.g., for Seurat VlnPlot is rotated by default).

Usage

```
UnRotate_X(...)
```

Arguments

`...`	extra arguments passed to `ggplot2::theme()`.

Value

Returns a list-like object of class *theme*.

Examples

```
library(Seurat)
p <- VlnPlot(object = pbmc_small, features = "CD3E")
p + UnRotate_X()
```

`Updated_HGNC_Symbols` *Update HGNC Gene Symbols*

Description

Update human gene symbols using data from HGNC. This function will store cached data in package directory using (BiocFileCache). Use of this function requires internet connection on first use (or if setting `update_symbol_data = TRUE`). Subsequent use does not require connection and will pull from cached data.

Usage

```
Updated_HGNC_Symbols(
  input_data,
  update_symbol_data = NULL,
  case_check_as_warn = FALSE,
  verbose = TRUE
)
```

Arguments

`input_data`	Data source containing gene names. Accepted formats are: • `charcter vector` • `Seurat Objects` • `data.frame`: genes as rownames • `dgCMatrix/dgTMatrix`: genes as rownames • `tibble`: genes in first column
`update_symbol_data`	logical, whether to update cached HGNC data, default is NULL. If `NULL` BiocFileCache will check and prompt for update if cache is stale. If `FALSE` the BiocFileCache stale check will be skipped and current cache will be used. If `TRUE` the BiocFileCache stale check will be skipped and HGNC data will be downloaded.
`case_check_as_warn`	logical, whether case checking of features should cause abort or only warn, default is FALSE (abort). Set to TRUE if atypical names (i.e. old LOC naming) are present in input_data.
`verbose`	logical, whether to print results detailing numbers of symbols, found, updated, and not found; default is TRUE.

Value

data.frame containing columns: input_features, Approved_Symbol (already approved; output unchanged), Not_Found_Symbol (symbol not in HGNC; output unchanged), Updated_Symbol (new symbol from HGNC; output updated).

Examples

```
## Not run:
new_names <- Updated_HGNC_Symbols(input_data = Seurat_Object)

## End(Not run)
```

`Updated_MGI_Symbols` *Update MGI Gene Symbols*

Description

Update mouse gene symbols using data from MGI This function will store cached data in package directory using (BiocFileCache). Use of this function requires internet connection on first use (or if setting `update_symbol_data = TRUE`). Subsequent use does not require connection and will pull from cached data.

Usage

```
Updated_MGI_Symbols(input_data, update_symbol_data = NULL, verbose = TRUE)
```

Arguments

`input_data`	Data source containing gene names. Accepted formats are: • `charcter vector` • `Seurat Objects` • `data.frame`: genes as rownames • `dgCMatrix/dgTMatrix`: genes as rownames • `tibble`: genes in first column
`update_symbol_data`	logical, whether to update cached MGI data, default is NULL. If `NULL` BiocFileCache will check and prompt for update if cache is stale. If `FALSE` the BiocFileCache stale check will be skipped and current cache will be used. If `TRUE` the BiocFileCache stale check will be skipped and MGI data will be downloaded.
`verbose`	logical, whether to print results detailing numbers of symbols, found, updated, and not found; default is TRUE.

Value

data.frame containing columns: input_features, Approved_Symbol (already approved; output unchanged), Not_Found_Symbol (symbol not in MGI; output unchanged), Updated_Symbol (new symbol from MGI; output updated).

Examples

```
## Not run:
new_names <- Updated_MGI_Symbols(input_data = Seurat_Object)

## End(Not run)
```

`VariableFeaturePlot_scCustom` *Custom Labeled Variable Features Plot*

Description

Creates variable features plot with N number of features already labeled by default.

Usage

```
VariableFeaturePlot_scCustom(
  seurat_object,
  num_features = 10,
  custom_features = NULL,
  label = TRUE,
  pt.size = 1,
  colors_use = c("black", "red"),
  repel = TRUE,
  y_axis_log = FALSE,
  assay = NULL,
  selection.method = NULL,
  ...
)
```

Arguments

`seurat_object`	Seurat object name.
`num_features`	Number of top variable features to highlight by color/label.
`custom_features`	A vector of custom feature names to label on plot instead of labeling top variable genes.
`label`	logical. Whether to label the top features. Default is TRUE.
`pt.size`	Adjust point size for plotting.
`colors_use`	colors to use for plotting. Default is "black" and "red".
`repel`	logical (default TRUE). Whether or not to repel the feature labels on plot.
`y_axis_log`	logical. Whether to change y axis to log10 scale (Default is FALSE).
`assay`	Assay to pull variable features from.

selection.method
: If more then one method use to calculate variable features specify which method to use for plotting. See selection.method parameter in VariableFeaturePlot for list of options.

...
: Extra parameters passed to VariableFeaturePlot.

Value

A ggplot object

Examples

```
library(Seurat)
VariableFeaturePlot_scCustom(seurat_object = pbmc_small, num_features = 10)
```

Variable_Features_ALL_LIGER

Perform variable gene selection over whole dataset

Description

Performs variable gene selection for LIGER object across the entire object instead of by dataset and then taking union.

Usage

```
Variable_Features_ALL_LIGER(
  liger_object,
  num_genes = NULL,
  var.thresh = 0.3,
  alpha.thresh = 0.99,
  tol = 1e-04,
  do.plot = FALSE,
  pt.size = 1.5,
  chunk = 1000
)
```

Arguments

liger_object
: LIGER object name.

num_genes
: Number of genes to find. Optimizes the value of var.thresh to get this number of genes, (Default is NULL).

var.thresh
: Variance threshold. Main threshold used to identify variable genes. Genes with expression variance greater than threshold (relative to mean) are selected. (higher threshold -> fewer selected genes).

alpha.thresh	Alpha threshold. Controls upper bound for expected mean gene expression (lower threshold -> higher upper bound). (default 0.99)
tol	Tolerance to use for optimization if num.genes values passed in (default 0.0001). Only applicable for rliger < 2.0.0.
do.plot	Display log plot of gene variance vs. gene expression. Selected genes are plotted in green. (Default FALSE)
pt.size	Point size for plot.
chunk	size of chunks in hdf5 file. (Default 1000)

Value

A LIGER Object with variable genes in correct slot.

References

Matching function parameter text descriptions are taken from `rliger::selectGenes` which is called by this function after creating new temporary object/dataset. https://github.com/welch-lab/liger. (License: GPL-3).

Examples

```
## Not run:
liger_obj <- Variable_Features_ALL_LIGER(liger_object = liger_obj, num_genes = 2000)

## End(Not run)
```

---

viridis_plasma_dark_high

Viridis Shortcuts

Description

Quick shortcuts to access viridis palettes

Usage

```
viridis_plasma_dark_high

viridis_plasma_light_high

viridis_inferno_dark_high

viridis_inferno_light_high

viridis_magma_dark_high
```

viridis_magma_light_high

viridis_dark_high

viridis_light_high

Format

An object of class character of length 250.

An object of class character of length 250.

An object of class character of length 250.

An object of class character of length 250.

An object of class character of length 250.

An object of class character of length 250.

An object of class character of length 250.

An object of class character of length 250.

Value

A color palette for plotting

Examples

```
## Not run:
FeaturePlot_scCustom(object = seurat_object, features = "Cx3cr1",
colors_use = viridis_plasma_dark_high, na_color = "lightgray")

## End(Not run)
```

VlnPlot_scCustom *VlnPlot with modified default settings*

Description

Creates DimPlot with some of the settings modified from their Seurat defaults (colors_use, shuffle, label).

Usage

```
VlnPlot_scCustom(
  seurat_object,
  features,
  colors_use = NULL,
  pt.size = NULL,
  group.by = NULL,
```

```
  split.by = NULL,
  plot_median = FALSE,
  plot_boxplot = FALSE,
  median_size = 15,
  idents = NULL,
  num_columns = NULL,
  raster = NULL,
  add.noise = TRUE,
  ggplot_default_colors = FALSE,
  color_seed = 123,
  ...
)
```

Arguments

seurat_object	Seurat object name.
features	Feature(s) to plot.
colors_use	color palette to use for plotting. By default if number of levels plotted is less than or equal to 36 it will use "polychrome" and if greater than 36 will use "varibow" with shuffle = TRUE both from DiscretePalette_scCustomize.
pt.size	Adjust point size for plotting.
group.by	Name of one or more metadata columns to group (color) cells by (for example, orig.ident); default is the current active.ident of the object.
split.by	Feature to split plots by (i.e. "orig.ident").
plot_median	logical, whether to plot median for each ident on the plot (Default is FALSE).
plot_boxplot	logical, whether to plot boxplot inside of violin (Default is FALSE).
median_size	Shape size for the median is plotted.
idents	Which classes to include in the plot (default is all).
num_columns	Number of columns in plot layout. Only valid if split.by != NULL.
raster	Convert points to raster format. Default is NULL which will rasterize by default if greater than 100,000 total points plotted (# Cells x # of features).
add.noise	logical, determine if adding a small noise for plotting (Default is TRUE).
ggplot_default_colors	logical. If colors_use = NULL, Whether or not to return plot using default ggplot2 "hue" palette instead of default "polychrome" or "varibow" palettes.
color_seed	random seed for the "varibow" palette shuffle if colors_use = NULL and number of groups plotted is greater than 36. Default = 123.
...	Extra parameters passed to VlnPlot.

Value

A ggplot object

References

Many of the param names and descriptions are from Seurat to facilitate ease of use as this is simply a wrapper to alter some of the default parameters https://github.com/satijalab/seurat/blob/master/R/visualization.R (License: GPL-3).

Examples

```
library(Seurat)
VlnPlot_scCustom(seurat_object = pbmc_small, features = "CD3E")
```

WhichCells.liger *Extract Cells for particular identity*

Description

Extract all cell barcodes for a specific identity

Usage

```
## S3 method for class 'liger'
WhichCells(
  object,
  idents = NULL,
  ident_col = NULL,
  by_dataset = FALSE,
  invert = FALSE,
  ...
)
```

Arguments

object	LIGER object name.
idents	identities to extract cell barcodes.
ident_col	name of meta data column to use when subsetting cells by identity values. Default is NULL, which will use the objects default clustering as the ident_col.
by_dataset	logical, whether to return vector with cell barcodes for all idents in or to return list (1 entry per dataset with vector of cells) (default is FALSE; return vector).
invert	logical, invert the selection of cells (default is FALSE).
...	Arguments passed to other methods

Value

vector or list depending on by_dataset parameter

Examples

```
## Not run:
# Extract cells from ident =1 in current default clustering
ident1_cells <- WhichCells(object = liger_object, idents = 1)

# Extract all cells from "stim" treatment from object
stim_cells <- WhichCells(object = liger_object, idents = "stim", ident_col = "Treatment")

## End(Not run)
```

Index