Package 'scHSQ'

January 7, 2025

Type Package

Title Indentification of evolutionarily conserved genes enriched for cell-type information

Version 1.0.0

Date 2025-01-02

Author UnJin Lee

Maintainer UnJin Lee <ulee@rockefeller.edu>

Depends R (>= 3.2.0), methods

Imports Biobase, monocle3, ggplot2

Suggests

Description Implementation of HSQ criteria identification using ANOVA F-statistics, allowing for projection of evolutionarily divergent single cell-RNA-Seq data on unified manifolds.

biocViews

License GPL version 3

R topics documented:

`applyMarkGenes` *Apply marker genes to generate cross-species UMAP and associated clusterings*

Description

Apply marker genes to generate cross-species UMAP and associated clusterings. Marker genes are applied by subsetting read count data to marker gene set contained in in_scHSQ object, then follows standard monocle3 processing pipeline. Specifically, process_cds, align_cds (aligning over original species labels), reduce_dimension, and cluster_cells are executed. Resulting processed cell_data_set object is stored in in_scHSQ and returned.

Usage

```
applyMarkGenes(in_scHSQ, leiden_res=3e-4, new_clu="HSQ_clu")
```

Arguments

`in_scHSQ`	a scHSQ object
`leiden_res`	a numerical value corresponding to the resolution for leiden clustering. Arbitrarily defaults to 3e-4 for historical reasons.
`new_clu`	a character that will be used to label the new cluster. Defaults to "HSQ_clu".

Value

A scHSQ object with newly processed cell_data_set containing cross-species UMAP.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Apply newly identified marker genes passing HSQ criteria

dros_testis <- genMarkGenes(dros_testis)
dros_testis <- applyMarkGenes(dros_testis)

## End(Not run)
```

`calcANOVA` *Calculates ANOVA statistics for original input cell_data_set*

Description

Calculates ANOVA statistics for both species and cell type labels to explain variance in read count on a per gene basis.

Usage

```
calcANOVA(in_scHSQ, verbose)
```

Arguments

`in_scHSQ`	a scHSQ object
`verbose`	a logical for printing progress of computations. Defaults to TRUE.

Value

A scHSQ objected with updated ANOVA F-statistic values for further processing.

Author(s)

UnJin Lee

Examples

```
## Not run:
## calculate ANOVA F-statistic values
dros_testis <- calcANOVA(dros_testis)

## End(Not run)
```

calcANOVACor *Calculates degree of cell type-specific evolutionary divergence*

Description

Calculates degree of cell type-specific evolutionary divergence. This is done by calculating the Spearman correlation between log(F-statistic) value for species-of-origin labels and cell type assignment labels.

Usage

```
calcANOVACor(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A numerical value computing the degree of cell type-specific evolutionary divergence.

Author(s)

UnJin Lee

Examples

```
## Not run:
## calculate degree of correlation between species-of-origin log(F-statistic) and cell type
anova_cor <- calcANOVACor(in_scHSQ)

## End(Not run)
```

findPercentile *Identification of empirical cutoff for probability thresholds*

Description

Identification of empirical cutoff for probability thresholds. Performed through simple sorting. Equivalent to the inverse of the empirical cumulative distribution function.

Usage

```
findPercentile(dat, p)
```

Arguments

dat	numeric vector of empirical data
p	a numeric value between 0 and 1 specifying the threshold probability.

Value

A numeric value representing the empirical distribution correlating to the threshold probability.

Author(s)

UnJin Lee

Examples

```
## Not run:
## find empirical value corresponding to probability threshold
dist <- rnorm(1000, 0, 1)
median_thresh <- findPercentile(dist, 0.5)

## End(Not run)
```

---

genMarkGenes *Generate Marker Genes*

Description

Applies HSQ criteria to generate a list of marker genes with high cell type specificity and low species specificity.

Usage

```
genMarkGenes(in_scHSQ, run_ANOVA=TRUE)
```

Arguments

in_scHSQ	a scHSQ object
run_ANOVA	a logical for calculating new ANOVA values if they are not present. Defaults to TRUE.

Value

A scHSQ object that may be used for further processing.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Generate new marker gene set
dros_testis <- genMarkGenes(dros_testis)

## End(Not run)
```

`getANOVAStats` *Queries ANOVA statistics in processed cell_data_set*

Description

Queries ANOVA statistics in processed cell_data_set slot. Useful for testing.

Usage

```
getANOVAStats(in_scHSQ)
```

Arguments

`in_scHSQ`	a scHSQ object

Value

A matrix with ANOVA F-statistic values for each gene.

Author(s)

UnJin Lee

Examples

```
## Not run:
## changes ANOVA F-statistic values
anova_vals <- getANOVAStats(dros_testis)
anova_vals[,1] <- sample(anova_vals[,1], replace=FALSE)
anova_vals[,2] <- sample(anova_vals[,2], replace=FALSE)

dros_testis <- setANOVAStats(dros_testis, anova_vals)

## End(Not run)
```

getCTLabs *Queries metadata label for cell type*

Description

Queries metadata label for cell type. This is a column name of pData in the contained cell_data_set objects.

Usage

```
getCTLabs(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A character corresponding to metadata label for cell type.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query metadata label for cell type
ct_labs <- getCTLabs(dros_testis)

## End(Not run)
```

getFullCDS *Query original cell_data_set*

Description

Queries original cell_data_set.

Usage

```
getFullCDS(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A cell_data_set that was used for analysis.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query original cell_data_set
original_cds <- getFullCDS(dros_testis)

## End(Not run)
```

`getFullExprs` *Query read counts for original cell_data_set*

Description

Queries read counts for original cell_data_set saved in its exprs slot.

Usage

```
getFullExprs(in_scHSQ)
```

Arguments

`in_scHSQ`	a scHSQ object

Value

A sparse matrix containing the read counts for the original cell_data_set.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query read counts for original cell_data_set
dros_exprs <- getFullExprs(dros_testis)

## End(Not run)
```

getFullPData *Query metadata for original cell_data_set*

Description

Queries metadata for original cell_data_set saved in its pData slot.

Usage

```
getFullPData(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A data.frame containing the pData for the original cell_data_set.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query metadata for original cell_data_set
dros_pdata <- getFullPData(dros_testis)

## End(Not run)
```

getMarkGenes *Query Marker Genes*

Description

Queries marker gene set.

Usage

```
getMarkGenes(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A character vector of marker genes.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query marker gene set
marker_genes <- getMarkGenes(dros_testis)

## End(Not run)
```

`getReducedCDS` *Query processed cell_data_set*

Description

Queries processed cell_data_set after application of HSQ criteria.

Usage

```
getReducedCDS(in_scHSQ)
```

Arguments

`in_scHSQ`	a scHSQ object

Value

A cell_data_set containing data used for generating cross-species UMAP.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query newly processed cell_data_set
new_cds <- getReducedCDS(dros_testis)

## End(Not run)
```

getRefCellType *Queries cell type information for reference species*

Description

Queries cell type information for reference species.

Usage

```
getRefCellType(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A character vector corresponding to the cell type information for the reference species.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query reference cell type information
mel_celltypes <- getRefCellType(dros_testis)

## End(Not run)
```

getRefIdx *Query indices for cells in the reference species*

Description

Queries indices for cells in the reference species.

Usage

```
getRefIdx(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A numeric vector corresponding to the indices for cells derived from the original reference species.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query indices for reference species' cells
mel_idx <- getRefIdx(dros_testis)

## End(Not run)
```

---

getRefSpecExprs *Queries read counts from cells in the reference species*

Description

Queries read counts from cells in the reference species.

Usage

```
getRefSpecExprs(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A sparse matrix with read counts from reference species cells.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query read counts from reference species cells
mel_exprs <- getRefSpecExprs(dros_testis)

## End(Not run)
```

getSpp *Queries metadata for species of origin*

Description

Queries metadata species of origin for each cell. This is the species information contained in the pData of contained cell_data_set objects.

Usage

```
getSpp(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A character vector corresponding to species of origin for each cell.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query species of origin metadata
spp_info <- getSpp(dros_testis)

## End(Not run)
```

getSppLabs *Queries metadata label for species*

Description

Queries metadata label for species. This is a column name of pData in the contained cell_data_set objects.

Usage

```
getSppLabs(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

A character corresponding to metadata label for species.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Query metadata label for species
spp_labs <- getSppLabs(dros_testis)

## End(Not run)
```

`new_scHSQ` *Initialize new scHSQ object*

Description

Allows for the generation of a new scHSQ object containing relevant information and slots for applying the HSQ criteria on monocle3 cell_data_set objects.

Usage

```
new_scHSQ(in_CDS, ref_species, mdat_lab_spec, mdat_lab_ct, color_scheme="default",
```

Arguments

`in_CDS`	monocle3 cell_data_set object with pData labels for species and cell type
`ref_species`	a character reflecting the reference species. ref_species should be a level of the pData label for species-of-origin
`mdat_lab_spec`	a character reflecting the metadata label for species information. mdat_lab_spec should be a column name of the pData of in_CDS
`mdat_lab_ct`	a character reflecting the metadata label for cell type information. mdat_lab_ct should be a column name of the pData of in_CDS
`color_scheme`	a character vector specifying a valid ggplot2 color scheme for visualization of cell type classifications. Leave blank for default color schemes.
`p_spec`	a numeric value between 0 and 1 specifying the threshold probability for species. Defaults to 0.5.
`p_ct`	a numeric value between 0 and 1 specifying the threshold probability for cell type. Defaults to 0.5.

Value

A new scHSQ object that may be used for further processing.

Author(s)

UnJin Lee

Examples

```
## Not run:
require(Biobase, monocle3)
## Generate new scHSQ object
data(testis_3sp)
dros_testis <- new_scHSQ(testis_3sp, "mel", "spec", "known_type")

## End(Not run)
```

---

plotANOVA *Generate plots summarizing ANOVA analysis*

Description

Generate plots summarizing ANOVA analysis. Plots will show log(F-statistic) values for species-of-origin and cell type assignment on a per gene basis. Plots will also contain lines reflecting HSQ criteria probability thresholds in addition to correlation between log(F-statistic) values. Rasterized plots are recommended to have width=1500, height=1500, res=300, pointsize=5.

Usage

```
plotANOVA(in_scHSQ)
```

Arguments

in_scHSQ	a scHSQ object

Value

NULL. Execution of function will output a plot to current R graphics device.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Generate ANOVA plot

png("scHSQ_ANOVA.png", width=1500, height=1500, res=300, pointsize=5)
plotANOVA(dros_testis)
dev.off()

## End(Not run)
```

`plotCTAssign` *Plot cell type assignments in cross-species UMAP*

Description

Plot cell type assignments in cross-species UMAP. Returned object is a list of UMAP projections containing: 1) known cell type labels for the reference species, 2) new clusters for all cells across all species, and 3) clusters for cells in each respective species. Note that arguments for monocle3's plot_cells function can be passed to plotCTAssign function and will automatically be applied to all elements of output. Rasterized figures are recommended to use width=1500, height=1500, res=300, pointsize=5.

Usage

```
plotCTAssign(in_scHSQ, new_labels, group_labs=TRUE, verbose=TRUE, ...)
```

Arguments

`in_scHSQ`	a scHSQ object
`new_labels`	a character corresponding to a metadata field in the processed cell_data_set's pData slot. Defaults to "HSQ_clu".
`group_labs`	a logical for whether or not group labels are included in resulting plots. Defaults to TRUE.
`verbose`	a logical for whether or not extra messages are outputted to the console. Defaults to TRUE.

Value

A list of ggplot2 objects.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Generate figures on cross-species UMAP

ctplots <- plotCTAssign(dros_testis)

png("scHSQ_UMAP_ref.png", width=1500, height=1500, res=300, pointsize=5)
ctplots[[1]]
dev.off()

png("scHSQ_UMAP_all.png", width=1500, height=1500, res=300, pointsize=5)
ctplots[[2]]
dev.off()
```

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

`plotGeneExprs` *Plot gene expression in cross-species UMAP*

Description

Plot gene expression in cross-species UMAP. Returned object is a UMAP projection showing expression for a single gene. Note that arguments for monocle3's plot_cells function can be passed to plotCTAssign function and will automatically be applied to all elements of output. Rasterized figures are recommended to use width=1500, height=1500, res=300, pointsize=5.

Usage

```
plotGeneExprs(in_scHSQ, gene, species=NULL, label_cell_groups=FALSE, alpha=0.35, ..
```

Arguments

`in_scHSQ`	a scHSQ object
`gene`	a character specifying which gene to plot
`species`	a character specifying which species to plot. Defaults to NULL for all species.
`label_cell_groups`	a logical parameter passed to plot_cells. Defaults to FALSE.
`alpha`	a numerical value passed to plot_cells. Defaults to 0.35.

Value

A ggplot2 object.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Generate figures on cross-species UMAP

gene_exprs <- plotGeneExprs(dros_testis, "Rbp4", alpha=0.5, norm_method="size_only", min_exp

png("scHSQ_UMAP_Rbp4.png", width=1500, height=1500, res=300, pointsize=5)
gene_exprs
dev.off()


## End(Not run)
```

`reclusterCDS` *Re-cluster over processed cell_data_set*

Description

Re-cluster over processed cell_data_set. Useful for adjusting resolution of clustering to find better cell type mappings.

Usage

```
reclusterCDS(in_scHSQ, leiden_res=3e-4, new_clu="HSQ_clu1")
```

Arguments

`in_scHSQ`	a scHSQ object
`leiden_res`	a numerical value corresponding to the resolution for leiden clustering. Arbitrarily defaults to 3e-4 for historical reasons.
`new_clu`	a character that will be used to label the new cluster. Defaults to "HSQ_clu1".

Value

A scHSQ object with newly processed cell_data_set containing cross-species UMAP.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Apply newly identified marker genes passing HSQ criteria

dros_testis <- genMarkGenes(dros_testis)
dros_testis <- applyMarkGenes(dros_testis)
dros_testis <- reclusterCDS(dros_testis, leiden_res=1e-4)

## End(Not run)
```

`remapCTAssign` *Remaps cell type assignments in cell type metadata for processed cell_data_set*

Description

Remaps cell type assignments in cell type metadata for processed cell_data_set. This is useful when transferring known cell type labels onto newly identified clusters across species. ct_map is assumed to have the original cell type labels in the first column, and new labels in the second column. The first colname of ct_map must correlated to a metadata label in pData of the processed cell_data_set in in_scHSQ. The second colname of ct_map will be used to assign a new metadata label in pData of the processed cell_data_set.

Usage

```
remapCTAssign(in_scHSQ, ct_map)
```

Arguments

`in_scHSQ`	a scHSQ object
`ct_map`	a data.frame that maps cell type labels in the processed cell_data_set's pData onto new cell type labels.

Value

A scHSQ objected with updated cell type labels.

Author(s)

UnJin Lee

Examples

```
## Not run:
## reassign newly generated clusters onto known cell types
old_clu <- as.character(1:10)
new_clu <- c("Late spermatocytes", "Early spermatocytes", "Early spermatids", "Early spermat

cell_types <- data.frame(HSQ_clu=old_clu, HSQ_clu1=new_clu)

dros_testis <- remapCTAssign(dros_testis, cell_types)

## End(Not run)
```

scHSQ *The scHSQ package*

Description

The scHSQ package provides three primary functions. First, it allows for the identification of genes that fulfill the HSQ criteria using single cell RNA-Seq data. It also utilizes newly identified gene set to project cross-species data onto a single UMAP projection, allowing for the application of known cell type assignments across tissues that have undergone cell type-specific evolutionary divergence. Finally, it allows for visualization of a single gene's expression patterns on a this cross-species UMAP projection.

scHSQ-class *Class* "scHSQ"

Description

The scHSQ object contains the necessary elements defining the data set on which the HSQ criteria will be applied.

Objects from the Class

Objects can be created by calls of the form `new_scHSQ("cell_data_set", "character", "character", "character", ...)`. Objects all contain 14 slots - fullCDS, n_species, n_celltype, ref_spp, spp_labs, idx_ref, celltype_labs, col_scheme, p_species, p_celltype, anova_stats, marker_list, reducedCDS, hsq_status.

Slots

`fullCDS`: Object of class "`cell_data_set`" ~~

`n_species`: Object of class "`numeric`" ~~

`n_celltype`: Object of class "`numeric`" ~~

`ref_spp`: Object of class "`character`" ~~

`spp_labs`: Object of class "`character`" ~~

`idx_ref`: Object of class "`numeric`" ~~

`celltype_labs`: Object of class "`character`" ~~

`col_scheme`: Object of class "`character`" ~~

`p_species`: Object of class "`numeric`" ~~

`p_celltype`: Object of class "`numeric`" ~~

`anova_stats`: Object of class "`matrix`" ~~

`marker_list`: Object of class "`character`" ~~

`reducedCDS`: Object of class "`cell_data_set`" ~~

`hsq_status`: Object of class "`logical`" ~~

Methods

new_scHSQ `signature(in_CDS = "cell_data_set", ref_species = "character", mdat_lab_spec = "character", mdat_lab_ct = "character")`: ...

genMarkGenes `signature(in_scHSQ = "scHSQ")`: ...

getMarkGenes `signature(in_scHSQ = "scHSQ")`: ...

getFullCDS `signature(in_scHSQ = "scHSQ")`: ...

getReducedCDS `signature(in_scHSQ = "scHSQ")`: ...

setReducedCDS `signature(in_scHSQ = "scHSQ", in_CDS = "cell_data_set")`: ...

getFullPData `signature(in_scHSQ = "scHSQ")`: ...

getFullExprs `signature(in_scHSQ = "scHSQ")`: ...

getRefIdx `signature(in_scHSQ = "scHSQ")`: ...

setRefIdx `signature(in_scHSQ = "scHSQ", idx = "numeric")`: ...

getRefSpecExprs `signature(in_scHSQ = "scHSQ")`: ...

getSppLabs `signature(in_scHSQ = "scHSQ")`: ...

getCTLabs `signature(in_scHSQ = "scHSQ")`: ...

setCellType `signature(in_scHSQ = "scHSQ", ct_labels = "character", ct_field_new = "character")`: ...

getRefCellType `signature(in_scHSQ = "scHSQ")`: ...

getSpp `signature(in_scHSQ = "scHSQ")`: ...

setPThresh `signature(in_scHSQ = "scHSQ")`: ...

findPercentile `signature(dat = "numeric", p = "numeric")`: ...

setANOVAStats `signature(in_scHSQ = "scHSQ", in_anova="matrix")`: ...

getANOVAStats `signature(in_scHSQ = "scHSQ")`: ...

calcANOVA `signature(in_scHSQ = "scHSQ")`: ...

remapCTAssign `signature(in_scHSQ = "scHSQ", ct_map = "data.frame")`: ...

calcANOVACor `signature(in_scHSQ = "scHSQ")`: ...

plotANOVA `signature(in_scHSQ = "scHSQ")`: ...

applyMarkGenes `signature(in_scHSQ = "scHSQ")`: ...

reclusterCDS `signature(in_scHSQ = "scHSQ")`: ...

plotCTAssign `signature(in_scHSQ = "scHSQ")`: ...

plotGeneExprs `signature(in_scHSQ = "scHSQ", gene = "character")`: ...

Author(s)

UnJin Lee

References

Lee U, Li C, Langer CB, Svetec N, Zhao L (2025) Comparative Single Cell Analysis of Transcriptional Bursting Reveals the Role of Genome Organization on de novo Transcript Origination. bioRxiv doi:10.1101/2024.04.29.591771

Examples

```
showClass("scHSQ")
```

`setANOVAStats` *Sets ANOVA statistics in processed cell_data_set*

Description

Sets ANOVA statistics in processed cell_data_set slot. Useful for testing.

Usage

```
setANOVAStats(in_scHSQ, in_anova)
```

Arguments

`in_scHSQ`	a scHSQ object
`in_anova`	numeric matrix of ANOVA F-statistic values for each gene

Value

A scHSQ object with updated ANOVA F-statistic values.

Author(s)

UnJin Lee

Examples

```
## Not run:
## changes ANOVA F-statistic values
anova_vals <- getANOVAStats(dros_testis)
anova_vals[,1] <- sample(anova_vals[,1], replace=FALSE)
anova_vals[,2] <- sample(anova_vals[,2], replace=FALSE)

dros_testis <- setANOVAStats(dros_testis, anova_vals)

## End(Not run)
```

`setCellType` *Sets new metadata label for updated cell type information*

Description

Sets new metadata label for updated cell type information. Updated information is a new metadata label in contained processed cell_data_set object using the ct_field_new label. Names of ct_labels should correspond to UMIs.

Usage

```
setCellType(in_scHSQ, ct_labels, ct_field_new)
```

Arguments

`in_scHSQ`	a scHSQ object
`ct_labels`	a character vector of updated cell type labels
`ct_field_new`	a character representing the metadata field for updated cell type labels

Value

A scHSQ object with updated cell type labels in the processed cell_data_set.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Set new cell type information
new_ct_labs <- rep(as.character(1:5000), 3)
names(new_ct_labs) <- rownames(getFullPData(dros_testis))

dros_testis <- setCellType(dros_testis, new_ct_labs, "new_labs")

## End(Not run)
```

`setPThresh` *Sets probability thresholds for HSQ criteria*

Description

Sets probability thresholds for HSQ criteria. Useful if ANOVA values are already calculated and thresholds are adjusted.

Usage

```
setPThresh(in_scHSQ, p_spec=0.5, p_ct=0.5, gen_list=TRUE)
```

Arguments

`in_scHSQ`	a scHSQ object
`p_spec`	a numeric value between 0 and 1 specifying the threshold probability for species. Defaults to 0.5.
`p_ct`	a numeric value between 0 and 1 specifying the threshold probability for cell type. Defaults to 0.5.
`gen_list`	a logical for regenerating marker gene list after adjusting thresholds. Defaults to TRUE.

Value

A scHSQ object with updated probability thresholds.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Update probability thresholds to change total number of marker genes
dros_testis <- setPThresh(dros_testis, p_spec=0.75, p_ct=0.75)

## End(Not run)
```

setReducedCDS *Set new processed cell_data_set*

Description

Sets processed cell_data_set.

Usage

```
setReducedCDS(in_scHSQ, in_CDS)
```

Arguments

in_scHSQ	a scHSQ object
in_CDS	a cell_data_set object

Value

A scHSQ object with updated cell_data_set.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Set processed cell_data_set
dros_testis <- setReducedCDS(dros_testis, testis_3sp)

## End(Not run)
```

setRefIdx *Sets indices for cells in the reference species*

Description

Sets indices for cells in the reference species. Useful for excluding particular cells or cell type classifications for applying HSQ criteria.

Usage

```
setRefIdx(in_scHSQ, idx)
```

Arguments

in_scHSQ	a scHSQ object
idx	numeric vector of indices

Value

A scHSQ object with updated reference species indices.

Author(s)

UnJin Lee

Examples

```
## Not run:
## Set indices for reference species' cells
new_idx <- 1:2500
dros_testis <- setRefIdx(dros_testis, new_idx)

## End(Not run)
```

---

testis_3sp *Breast Cancer 443 Data Set*

Description

testis_3sp is an monocle3 cell_data_set object that contains scRNA-Seq counts for testis tissue obtained from D. melanogaster, D. yakuba, and D. ananassae. Only genes with one-to-one and reciprocal best orthology were retained. Data was merged using monocle3's combine_cds function. Minimal processing occurred, as monocle3's preprocess_cds function was applied. Additionally, while monocle3's reduce_dimension and cluster_cells functions were applied to this data set, there is no impact on downstream analysis performed in scHSQ.

Usage

```
data(testis_3sp)
```

Format

cell_data_set with expression data and phenotypic meta-data