

# Package: scGOclust (via r-universe)

September 17, 2024

**Type** Package

**Title** Measuring Cell Type Similarity with Gene Ontology in Single-Cell RNA-Seq

**Version** 0.2.1

**Description** Traditional methods for analyzing single cell RNA-seq datasets focus solely on gene expression, but this package introduces a novel approach that goes beyond this limitation. Using Gene Ontology terms as features, the package allows for the functional profile of cell populations, and comparison within and between datasets from the same or different species. Our approach enables the discovery of previously unrecognized functional similarities and differences between cell types and has demonstrated success in identifying cell types' functional correspondence even between evolutionarily distant species.

**URL** <https://github.com/Papatheodorou-Group/scGOclust>

**BugReports** <https://github.com/Papatheodorou-Group/scGOclust/issues>

**License** GPL (>= 3)

**Encoding** UTF-8

**LazyData** true

**LazyDataCompression** bzip2

**RoxygenNote** 7.2.3

**Imports** limma, Seurat(>= 5.0.0), biomaRt, dplyr, magrittr, stats, tibble, tidyr, Matrix, utils, networkD3, slanter

**Suggests** knitr, devtools, pheatmap, rmarkdown, httr

**VignetteBuilder** knitr

**Depends** R (>= 2.10)

**Repository** <https://papatheodorou-group.r-universe.dev>

**RemoteUrl** <https://github.com/papatheodorou-group/scgoclust>

**RemoteRef** HEAD

**RemoteSha** d014f3690313796a9ccd85157806aa4041d8de51

## Contents

analyzeGOSeurat . . . . .	2
cellTypeGOCorr . . . . .	3
crossSpeciesCellTypeGOCorr . . . . .	4
dme_subset . . . . .	5
dme_tbl . . . . .	6
ensemblToGo . . . . .	6
getCellTypeGO . . . . .	7
getCellTypeSharedGO . . . . .	8
getCellTypeSharedTerms . . . . .	9
globalvariables . . . . .	11
makeGOSeurat . . . . .	11
mmu_subset . . . . .	12
mmu_tbl . . . . .	12
plotCellTypeCorrHeatmap . . . . .	13
plotCellTypeSankey . . . . .	14
<b>Index</b>	<b>15</b>

---

analyzeGOSeurat	<i>standard seurat analysis on GO_seurat object</i>
-----------------	---

---

### Description

standard seurat analysis on GO\_seurat object

### Usage

```
analyzeGOSeurat(
  go_seurat_obj,
  cell_type_col,
  norm_log1p = TRUE,
  scale.factor = 10000,
  nfeatures = 2000,
  cluster_res = 1,
  min.dist = 0.3,
  ...
)
```

### Arguments

go_seurat_obj	go seurat object created by makeGOSeurat
cell_type_col	column name in mera.data storing cell type classes
norm_log1p	whether or not to perform data normalisation and log1p transformation, default TRUE
scale.factor	param for Seurat NormalizeData

nfeatures	param for Seurat FindVariableFeatures
cluster_res	resolution for Seurat FindClusters
min.dist	param for Seurat RunUMAP
...	additional params for all Seurat functions involved in this function

**Value**

standard analyzed GO seurat object until UMAP

**Examples**

```
library(scGOClust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
go_seurat_obj = makeGOSeurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")

analyzeGOSeurat(go_seurat_obj = go_seurat_obj, cell_type_col = "cell_type_annotation")
```

---

cellTypeGOCorr	<i>calculate correlation between cell types represented by scaled GO, per-species</i>
----------------	---

---

**Description**

calculate correlation between cell types represented by scaled GO, per-species

**Usage**

```
cellTypeGOCorr(cell_type_go, corr_method = "pearson")
```

**Arguments**

cell_type_go	cell type GO table calculated via getCellTypeGO
corr_method	correlation method, choose among "pearson", "kendall", "spearman", default 'pearson'

**Value**

a dataframe with correlation between cell types

**Examples**

```

library(scG0clust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
go_seurat_obj = makeG0Seurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")

cell_type_go = getCellTypeG0(go_seurat_obj = go_seurat_obj, cell_type_co = "cell_type_annotation")

cellTypeG0Corr(cell_type_go = cell_type_go, corr_method = "pearson")

```

---

crossSpeciesCellTypeG0Corr

*calculate cross-species correlation between cell types represented by scaled GO*

---

**Description**

calculate cross-species correlation between cell types represented by scaled GO

**Usage**

```

crossSpeciesCellTypeG0Corr(
  species_1,
  species_2,
  cell_type_go_sp1,
  cell_type_go_sp2,
  corr_method = "pearson"
)

```

**Arguments**

species_1	name of species one
species_2	name of species two
cell_type_go_sp1	cell type GO table of species one calculated via getCellTypeGO
cell_type_go_sp2	cell type GO table of species two calculated via getCellTypeGO
corr_method	correlation method, choose among "pearson", "kendall", "spearman", default 'pearson'

**Value**

correlation between cell types

**Examples**

```
library(scG0clust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
data(dme_tbl)
data(dme_subset)
mmu_go_obj = makeG0Seurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")
dme_go_obj = makeG0Seurat(ensembl_to_GO = dme_tbl,
  seurat_obj = dme_subset,
  feature_type = "external_gene_name")

mmu_cell_type_go = getCellTypeG0(go_seurat_obj = mmu_go_obj, cell_type_co = "cell_type_annotation")
dme_cell_type_go = getCellTypeG0(go_seurat_obj = dme_go_obj, cell_type_co = "annotation")

crossSpeciesCellTypeG0Corr(species_1 = 'mmusculus',
  species_2 = 'dmelanogaster',
  cell_type_go_sp1 = mmu_cell_type_go,
  cell_type_go_sp2 = dme_cell_type_go)
```

---

dme_subset	<i>Drosophila gut scRNA-seq data, 10X Chromium Subset to 45 cells per cell type as an example data</i>
------------	--

---

**Description**

Drosophila gut scRNA-seq data, 10X Chromium Subset to 45 cells per cell type as an example data

**Usage**

```
dme_subset
```

**Format**

a 'Seurat' object

**Source**

<<https://flycellatlas.org/>>

---

dme_tbl	<i>Drosophila EMSEMBL gene and GO annotation, subset to genes present in 'dme_subset'</i>
---------	---

---

**Description**

Drosophila EMSEMBL gene and GO annotation, subset to genes present in 'dme\_subset'

**Usage**

```
dme_tbl
```

**Format**

a 'data.frame' object

**Source**

<<http://www.ensembl.org/>>

---

ensemblToGo	<i>get requested ensembl ID to GO mapping table</i>
-------------	---

---

**Description**

get requested ensembl ID to GO mapping table

**Usage**

```
ensemblToGo(
  species,
  GO_type = "biological_process",
  GO_linkage_type = c("standard"),
  ...
)
```

**Arguments**

species	species name matching ensembl biomaRt naming, such as hsapiens, mmusculus
GO_type	GO term type, choose among 'biological_process', 'molecular_function', 'cellular_component', default 'biological_process'

GO\_linkage\_type GO annotation evidence codes to include. Default is 'standard', which means only including manually checked records (excluding IEA) and excluding those inferred from gene expression experiments to maximally suffice the species expression independence assumption. 'Stringent' means only including those with experimental evidence, also not from gene expression experiment, or from manual curation with evidence (excluding those from mass-annotation pipelines). Choose among 'experimental', 'phylogenetic', 'computational', 'author', 'curator', 'electronic', 'standard', 'stringent'

... additional params for useEnsembl function called in this function

**Value**

a table with ensembl to GO terms mapping including requested linkage type

**Examples**

```
library(scGOclust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
ensemblToGo("mmusculus", GO_type = "biological_process", GO_linkage_type = 'stringent')
```

---

<code>getCellTypeGO</code>	<i>get per cell type average scaled vector of GO terms</i>
----------------------------	--

---

**Description**

get per cell type average scaled vector of GO terms

**Usage**

```
getCellTypeGO(go_seurat_obj, cell_type_col, norm_log1p = TRUE)
```

**Arguments**

`go_seurat_obj` go seurat object created by `makeGOSeurat`

`cell_type_col` column name in `mera.data` storing cell type classes

`norm_log1p` whether or not to perform data normalisation and log1p transformation, default TRUE

**Value**

a table of scaled GO representation per cell type (averaged)

**Examples**

```

library(scGOclust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
go_seurat_obj = makeGOSeurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")
getCellTypeGO(go_seurat_obj = go_seurat_obj, cell_type_co = "cell_type_annotation")

```

---

```

getCellTypeSharedGO  get shared up and down regulated GO terms for all pairs of cell types

```

---

**Description**

get shared up and down regulated GO terms for all pairs of cell types

**Usage**

```

getCellTypeSharedGO(
  species_1,
  species_2,
  analyzed_go_seurat_sp1,
  analyzed_go_seurat_sp2,
  cell_type_col_sp1,
  cell_type_col_sp2,
  layer_use = "data",
  p_val_threshold = 0.01
)

```

**Arguments**

species_1	name of species one
species_2	name of species two
analyzed_go_seurat_sp1	analyzed GO seurat object of species one
analyzed_go_seurat_sp2	analyzed GO seurat object of species two
cell_type_col_sp1	cell type column name for species 1 data
cell_type_col_sp2	cell type column name for species 2 data
layer_use	layer to use for marker computation, default 'data' which after NormalizeData will be log1p normalized data.
p_val_threshold	p value threshold for selecting DEG (p_adjust)



**Value**

a list with sp1 raw, sp2 raw and shared, significant up and down regulated GO terms per cell type (pair)

**Examples**

```
library(scGOclust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
data(dme_tbl)
data(dme_subset)

mmu_go_obj = makeGOSeurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")
dme_go_obj = makeGOSeurat(ensembl_to_GO = dme_tbl,
  seurat_obj = dme_subset,
  feature_type = "external_gene_name")

mmu_go_obj_analyzed = analyzeGOSeurat(mmu_go_obj, "cell_type_annotation")
dme_go_obj_analyzed = analyzeGOSeurat(dme_go_obj, "annotation")

getCellTypeSharedGO(species_1 = 'mmusculus',
  species_2 = 'dmelanogaster',
  analyzed_go_seurat_sp1 = mmu_go_obj_analyzed,
  analyzed_go_seurat_sp2 = dme_go_obj_analyzed,
  cell_type_col_sp1 = 'cell_type_annotation',
  cell_type_col_sp2 = 'annotation',
  layer_use = "data",
  p_val_threshold = 0.01)
```

---

getCellTypeSharedTerms

*query co-up and co-down regulated GO terms from certain cell type pairs*

---

**Description**

query co-up and co-down regulated GO terms from certain cell type pairs

**Usage**

```
getCellTypeSharedTerms(
  shared_go,
  cell_type_sp1,
```

```

    cell_type_sp2,
    return_full = FALSE,
    arrange_avg_log2FC = TRUE
  )

```

### Arguments

```

shared_go      cell type shared GO table from getCellTypeSharedGO
cell_type_sp1  cell type from sp1 to query
cell_type_sp2  cell type from sp2 to query
return_full    if return also pvals and logfc info, default FALSE
arrange_avg_log2FC
                arrange result by decreasing mean avg_log2FC, default TRUE

```

### Value

a dataframe displaying co-up or co-down regulated GO terms for the queried cell type pair

### Examples

```

library(scGOclust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
data(dme_tbl)
data(dme_subset)

mmu_go_obj = makeGOSeurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")
dme_go_obj = makeGOSeurat(ensembl_to_GO = dme_tbl,
  seurat_obj = dme_subset,
  feature_type = "external_gene_name")

mmu_go_obj_analyzed = analyzeGOSeurat(mmu_go_obj, "cell_type_annotation")
dme_go_obj_analyzed = analyzeGOSeurat(dme_go_obj, "annotation")

shared_go = getCellTypeSharedGO(species_1 = 'mmusculus',
  species_2 = 'dmelanogaster',
  analyzed_go_seurat_sp1 = mmu_go_obj_analyzed,
  analyzed_go_seurat_sp2 = dme_go_obj_analyzed,
  cell_type_col_sp1 = 'cell_type_annotation',
  cell_type_col_sp2 = 'annotation',
  layer_use = "data",
  p_val_threshold = 0.01)

getCellTypeSharedTerms(shared_go = shared_go,
  cell_type_sp1 = 'intestine_Enterendocrine cell',

```

```
cell_type_sp2 = 'enteroendocrine cell',
return_full = FALSE)
```

---

globalvariables	<i>record some global variables: pre-defined column name in biomaRt query and markers</i>
-----------------	---

---

### Description

record some global variables: pre-defined column name in biomaRt query and markers

---

makeGOSeurat	<i>create a seurat object with GO terms</i>
--------------	---

---

### Description

create a seurat object with GO terms

### Usage

```
makeGOSeurat(ensembl_to_GO, seurat_obj, feature_type = "ensembl_gene_id")
```

### Arguments

ensembl_to_GO	ensembl_to_go mapping table from function ensemblToGo
seurat_obj	count matrix with genes to cells
feature_type	feature type of count matrix, choose from ensembl_gene_id, external_gene_name, default ensembl_gene_id

### Value

a seurat object with GO terms as features

### Examples

```
library(scGOclust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
makeGOSeurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")
```

---

mmu_subset	<i>Mouse stomach and intestine scRNA-seq data, microwell-seq Subset to 50 cells per cell type as an example data</i>
------------	--

---

**Description**

Mouse stomach and intestine scRNA-seq data, microwell-seq Subset to 50 cells per cell type as an example data

**Usage**

mmu\_subset

**Format**

a 'Seurat' object

**Source**

<<https://bis.zju.edu.cn/MCA/>>

---

mmu_tbl	<i>Mouse EMSEMBL gene and GO annotation, subset to genes present in 'mmu_subset'</i>
---------	--

---

**Description**

Mouse EMSEMBL gene and GO annotation, subset to genes present in 'mmu\_subset'

**Usage**

mmu\_tbl

**Format**

a 'data.frame' object

**Source**

<<http://www.ensembl.org/>>

---

`plotCellTypeCorrHeatmap`*plot clustered heatmap for cell type corr*

---

**Description**

plot clustered heatmap for cell type corr

**Usage**

```
plotCellTypeCorrHeatmap(corr_matrix, scale = NA, ...)
```

**Arguments**

<code>corr_matrix</code>	correlation matrix from <code>cellTypeGOCorr</code> or <code>crossSpeciesCellTypeGOCorr</code>
<code>scale</code>	scale value by column, row, or default no scaling (NA)
<code>...</code>	params to pass to <code>slanter::sheatmap</code>

**Value**

a sheatmap heatmap

**Examples**

```
library(scGOclust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)

go_seurat_obj = makeGOSeurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")

cell_type_go = getCellTypeGO(go_seurat_obj = go_seurat_obj, cell_type_co = "cell_type_annotation")

corr_matrix = cellTypeGOCorr(cell_type_go = cell_type_go, corr_method = "pearson")

plotCellTypeCorrHeatmap(corr_matrix = corr_matrix, scale = "column")
```

---

plotCellTypeSankey     *plot Sankey diagram for cell type links above a certain threshold*

---

**Description**

plot Sankey diagram for cell type links above a certain threshold

**Usage**

```
plotCellTypeSankey(corr_matrix, corr_threshold = 0.1, ...)
```

**Arguments**

corr\_matrix     cell type corr matrix from crossSpeciesCellTypeGOCorr  
corr\_threshold     minimum corr value for positively related cell types, default 0.6  
...     additional params for sankeyNetwork

**Value**

a Sankey plot showing related cell types

**Examples**

```
library(scGOclust)
library(httr)
httr::set_config(httr::config(ssl_verifypeer = FALSE))
data(mmu_tbl)
data(mmu_subset)
go_seurat_obj = makeGOSeurat(ensembl_to_GO = mmu_tbl,
  seurat_obj = mmu_subset,
  feature_type = "external_gene_name")

cell_type_go = getCellTypeGO(go_seurat_obj = go_seurat_obj, cell_type_co = "cell_type_annotation")
corr_matrix = cellTypeGOCorr(cell_type_go = cell_type_go, corr_method = "pearson")

plotCellTypeSankey(corr_matrix = corr_matrix, 0.1)
```

# Index

## \* datasets

- dme\_subset, [5](#)
- dme\_tbl, [6](#)
- mmu\_subset, [12](#)
- mmu\_tbl, [12](#)

analyzeGOSeurat, [2](#)

cellTypeGOCorr, [3](#)  
crossSpeciesCellTypeGOCorr, [4](#)

dme\_subset, [5](#)  
dme\_tbl, [6](#)

ensemblToGo, [6](#)

getCellTypeGO, [7](#)  
getCellTypeSharedGO, [8](#)  
getCellTypeSharedTerms, [9](#)  
globalvariables, [11](#)

makeGOSeurat, [11](#)  
mmu\_subset, [12](#)  
mmu\_tbl, [12](#)

plotCellTypeCorrHeatmap, [13](#)  
plotCellTypeSankey, [14](#)