

# Package ‘scOntoMatch’

June 27, 2022

**Type** Package

**Title** Aligning Ontology Annotation Across Single Cell Datasets with 'scOntoMatch'

**Version** 0.1.0

**Author** Yuyao Song [aut, cre] (<<https://orcid.org/0000-0001-9738-9726>>)

**Maintainer** Yuyao Song <[ysong@ebi.ac.uk](mailto:ysong@ebi.ac.uk)>

**Description** Unequal granularity of cell type annotation makes it difficult to compare scRNA-seq datasets at scale. Leveraging the ontology system for defining cell type hierarchy, 'scOntoMatch' aims to align cell type annotations to make them comparable across studies. The alignment involves two core steps: first is to trim the cell type tree within each dataset so each cell type does not have descendants, and then map cell type labels cross-studies by direct matching and mapping descendants to ancestors. Various functions for plotting cell type trees and manipulating ontology terms are also provided. In the Single Cell Expression Atlas hosted at EBI, a compendium of datasets with curated ontology labels are great inputs to this package.

**Imports** ontologyIndex, ontologyPlot, purrr

**URL** <https://github.com/YY-SONG0718/scOntoMatch>

**BugReports** <https://github.com/YY-SONG0718/scOntoMatch/issues>

**License** MIT + file LICENSE

**Encoding** UTF-8

**RoxygenNote** 7.2.0

**Depends** R(>= 3.5)

**Suggests** knitr, devtools, SeuratObject

**VignetteBuilder** knitr

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-06-27 15:00:08 UTC

**R topics documented:**

check_ontology_translate . . . . .	2
fill_query . . . . .	3
getOntologyId . . . . .	3
getOntologyName . . . . .	4
getOntoMapping . . . . .	4
getOntoMinimal . . . . .	5
getOntoMultiMapping . . . . .	6
getSeuratRds . . . . .	6
ontoMinimal . . . . .	7
ontoMultiMatch . . . . .	8
ontoMultiMinimal . . . . .	8
ontoTranslate . . . . .	9
plotMatchedOntoTree . . . . .	10
plotOntoTree . . . . .	11

<b>Index</b>	<b>12</b>
--------------	-----------

---

check\_ontology\_translate

*make sure ontology names are all translated to ontology ids while warning, consider manual reannotation*

---

**Description**

make sure ontology names are all translated to ontology ids while warning, consider manual reannotation

**Usage**

```
check_ontology_translate(obj, onts, ont, anno_col)
```

**Arguments**

obj	seurat rds object
onts	ontology ids from translate
ont	ontologyIndex object
anno_col	annotation column in obj@meta.data that is translated to onts ids

**Value**

do not return a value but output messages

---

fill_query	<i>Get a names list of ontology and id by id</i>
------------	--

---

**Description**

Get a names list of ontology and id by id

**Usage**

```
fill_query(all, query)
```

**Arguments**

all	all ontology id to plot tree
query	query ontology id to fill

**Value**

a color object to fill query in onto\_plot

**Examples**

```
## Not run:  
fill_query(all = c("CL0000548", "CL:0000066", "CL:0000082"), query = c("CL:0000082"))  
  
## End(Not run)
```

---

getOntologyId	<i>Get a names list of ontology and id by name</i>
---------------	--

---

**Description**

Get a names list of ontology and id by name

**Usage**

```
getOntologyId(ont, onto_name)
```

**Arguments**

ont	ontology object
onto_name	character vector of ontology names

**Value**

a named list mapping ontology id and ontology name

**Examples**

```
## Not run:
getOntologyId(onto_name = "epithelial cell of lung", ont = ont)

## End(Not run)
```

---

getOntologyName	<i>Get a names list of ontology and id by id</i>
-----------------	--

---

**Description**

Get a names list of ontology and id by id

**Usage**

```
getOntologyName(ont, onto_id)
```

**Arguments**

ont	ontology object
onto_id	character vector of ontology ids

**Value**

a named list mapping ontology id and ontology name

**Examples**

```
## Not run:
getOntologyName(onto_id = "epithelial cell of lung", ont = ont)

## End(Not run)
```

---

getOntoMapping	<i>match ontology terms by direct mapping and mapping descendants to ancestors</i>
----------------	--

---

**Description**

match ontology terms by direct mapping and mapping descendants to ancestors

**Usage**

```
getOntoMapping(ont, onts1, onts2)
```

**Arguments**

ont                    the ontology object from get\_OBO  
onts1                  a character vector of ontology id  
onts2                  a character vector of ontology id

**Value**

a named list for ontology id mapping looks like ontology\_id:ontology\_id

**Examples**

```
## Not run:  
getOntoMapping(ont = ont, onts1 = "CL:0000548", onts2 = c("CL0000548", "CL:0000066"))  
  
## End(Not run)
```

---

getOntoMinimal                    *match descendant terms to ancestor terms within a dataset*

---

**Description**

match descendant terms to ancestor terms within a dataset

**Usage**

```
getOntoMinimal(ont, onts)
```

**Arguments**

ont                    the ontology object from get\_OBO  
onts                   a character vector of ontology id

**Value**

a named list for ontology id mapping looks like ontology\_id:ontology\_id

**Examples**

```
## Not run:  
getOntoMinimal(ont = ont, onts = c("CL0000548", "CL:0000066", "CL:0000082"))  
  
## End(Not run)
```

---

getOntoMultiMapping     *Match descendants to ancestors in multiple ontology id lists*

---

### Description

Match descendants to ancestors in multiple ontology id lists

### Usage

```
getOntoMultiMapping(ont, onts)
```

### Arguments

ont	ontologyIndex object
onts	named list of ontology ids

### Value

a named character of mapping from:mapping to

### Examples

```
## Not run:
getOntoMultiMapping(ont = ont, onts = c("CL0000548", "CL:0000066", "CL:0000082"))

## End(Not run)
```

---

getSeuratRds     *read in seurat object .rds files as a named list of seurat object*

---

### Description

read in seurat object .rds files as a named list of seurat object

### Usage

```
getSeuratRds(metadata, sep)
```

### Arguments

metadata	a metadata file indicating name, path to 'seurat' rds file
sep	sep of the metadata file

### Value

a named list contains data name and the corresponding 'seurat' object

**Examples**

```
## Not run:
getSeuratRds(metadata = 'metadata.tsv', sep = '\t')

## End(Not run)
```

---

ontoMinimal	<i>get the minimal ontology tree of a dataset by reducing descendant terms to ancestor terms return obj meta.data[["cell_ontology_base"]] storing the reduced ontology annotation</i>
-------------	---

---

**Description**

get the minimal ontology tree of a dataset by reducing descendant terms to ancestor terms return obj meta.data[["cell\_ontology\_base"]] storing the reduced ontology annotation

**Usage**

```
ontoMinimal(obj, ont, anno_col, onto_id_col)
```

**Arguments**

- obj                    the seurat object
- ont                    ontologyIndex object
- anno\_col              the cell ontology text annotation column name
- onto\_id\_col          if also have ontology id column for direct mapping

**Value**

an seurat object with meta.data[["cell\_ontology\_base"]]

**Examples**

```
## Not run:
ontoMinimal(obj = seurat_obj, ont = ont, anno_col = 'ontology_name', onto_id_col = 'ontology_id')

## End(Not run)
```

---

ontoMultiMatch	<i>Core function of scOntoMatch Match the ontology annotation of multiple seurat objects</i>
----------------	--

---

**Description**

Core function of scOntoMatch Match the ontology annotation of multiple seurat objects

**Usage**

```
ontoMultiMatch(obj_list, anno_col, onto_id_col, ont)
```

**Arguments**

obj_list	a named list of seurat objects to match
anno_col	the cell ontology text annotation column name
onto_id_col	if also have ontology id column for direct mapping
ont	ontologyIndex object

**Value**

a list of seurat objects with annotation ontology mapped to each-other in obs[['cell\_ontology\_mapped']]

**Examples**

```
## Not run:
ontoMultiMatch(seurat_obj_list, ont, "ontology_name", 'ontology_id')

## End(Not run)
```

---

ontoMultiMinimal	<i>get the minimal ontology tree of a list of seurat objects by reducing descendant terms to ancestor terms return a named list of seurat objects with meta.data[["cell_ontology_base"]] storing the reduced ontology annotation</i>
------------------	--

---

**Description**

get the minimal ontology tree of a list of seurat objects by reducing descendant terms to ancestor terms return a named list of seurat objects with meta.data[["cell\_ontology\_base"]] storing the reduced ontology annotation

**Usage**

```
ontoMultiMinimal(obj_list, ont, anno_col = "cell_ontology_base", onto_id_col)
```



**Arguments**

obj\_list            a named list of seurat objects  
 ont                ontologyIndex object  
 anno\_col          the cell ontology text annotation column name  
 onto\_id\_col        if also have ontology id column for direct mapping

**Value**

a named list of seurat objects with meta.data[["cell\_ontology\_base"]]

**Examples**

```
## Not run:
ontoMultiMinimal(seurat_obj_list, ont, "cell_ontology_base", 'ontology_id')

## End(Not run)
```

---

ontoTranslate	<i>translate named list of obj_list to named list of cell ontology ids per obj</i>
---------------	--

---

**Description**

translate named list of obj\_list to named list of cell ontology ids per obj

**Usage**

```
ontoTranslate(obj_list, ont, onto_id_col, anno_col)
```

**Arguments**

obj\_list            a named list of seurat object  
 ont                ontologyIndex object  
 onto\_id\_col        if also have ontology id column for direct mapping  
 anno\_col          the cell ontology text annotation column name

**Value**

a named list of cell ontology ids

**Examples**

```
## Not run:
ontoTranslate(seurat_obj_list, ont, 'ontology_name', 'ontology_id')

## End(Not run)
```

---

plotMatchedOntoTree *Plot a ontology tree with matched ontology from ontoMatch*

---

### Description

Plot a ontology tree with matched ontology from ontoMatch

### Usage

```
plotMatchedOntoTree(  
  obj_list,  
  ont,  
  anno_col = "cell_ontology_mapped",  
  onto_id_col,  
  roots = c("CL:0000548"),  
  ...  
)
```

### Arguments

obj_list	a list of seurat obj files as the output of ontoMatch
ont	ontology object
anno_col	the cell ontology text annotation column name
onto_id_col	if also have ontology id column for direct mapping
roots	root ontology in tree to plot, default "animal cells" in cell ontology
...	additional parameters for ontologyPlot::onto_plot

### Value

a list of matched ontology tree plot

### Examples

```
## Not run:  
plotMatchedOntoTree(seurat_obj_list, ont, 'cell_ontology_mapped', 'ontology_id')  
  
## End(Not run)
```

---

plotOntoTree	<i>Plot a tree representation of ontology terms</i>
--------------	---

---

**Description**

Plot a tree representation of ontology terms

**Usage**

```
plotOntoTree(  
  ont,  
  onts,  
  plot_ancestors = TRUE,  
  ont_query = NULL,  
  roots = c("CL:0000548"),  
  ...  
)
```

**Arguments**

ont	ontology object
onts	ontology ids to plot
plot_ancestors	if plot ancestors or not
ont_query	query ontology to highlight in the tree
roots	root ontology in tree, default "animal cells" in cell ontology
...	additional parameters for ontologyPlot::onto_plot

**Value**

an ontology tree plot

**Examples**

```
## Not run:  
plotOntoTree(ont = ont, onts = c("CL:0000066", "CL:0000082"), ont_query = c("CL:0000082"))  
  
## End(Not run)
```

# Index

[check\\_ontology\\_translate](#), 2

[fill\\_query](#), 3

[getOntologyId](#), 3

[getOntologyName](#), 4

[getOntoMapping](#), 4

[getOntoMinimal](#), 5

[getOntoMultiMapping](#), 6

[getSeuratRds](#), 6

[ontoMinimal](#), 7

[ontoMultiMatch](#), 8

[ontoMultiMinimal](#), 8

[ontoTranslate](#), 9

[plotMatchedOntoTree](#), 10

[plotOntoTree](#), 11