Package 'CCN'

May 6, 2022

Type Package

Title Generate the Standardized Common Cell type Nomenclature (CCN)

Version 0.1.0

Author Jeremy Miller <jeremym@alleninstitute.org>

Maintainer Jeremy Miller <jeremym@alleninstitute.org>

Description To facilitate cross-dataset comparison, the Allen Institute created the CCN for matching and tracking cell types across studies. A primary objective of the CCN was to develop a system that would be straightforward and allow designation of cell types with or without hierarchical organization. This library applies the CCN to any data set. More details can be found here: https://portal.brain-map.org/explore/classes/nomenclature

License Allen Institute Software License

Encoding UTF-8

LazyData true

Imports dendextend (>= 1.7.0),
   ggplot2 (>= 2.2.1),
   dplyr (>= 0.3.4),
   data.table (>= 1.0),
   jsonlite (>= 1.1),
   rols (>= 2.0)

RoxygenNote 7.1.2

VignetteBuilder knitr

R topics documented:

---

add_attr_to_dend   *Add an attribute to dendrogram nodes*

Description

Adds a specified attribute to labeled nodes of a dendrogram

Usage

```
add_attr_to_dend(dend, value, attribute = "label")
```

Arguments

dend	dendrogram of annotated cell types (with labeled nodes)
value	named vector of attributes to add (values = attribute value; name = dendrogram node labels)
attribute	name of attribute to add or update in "dend"

Value

The input dendrogram with the new added/updated attribute.

---

annotate_nomenclature_from_metadata
                     Automatic annotation of nomenclature table from metadata

---

Description

Automatically annotate existing cell sets and add new cell sets as needed based on existing metadata columns, for metadata that represent groups of cell types (e.g., something like "sex" would not be appropriate, but "broad class" would). More specifically, this function will: 1. Identify all values corresponding to that column 2. For each value it (i) finds all relevant dendrogram labels, (ii) generates the corresponding cell set label, and (iii) cross-references this label with the existing table 3. If the label exists, the new metadata is added to the requested column in the nomenclature table, and if not it will generate a new entree in the table and then add the requested metadata 4. Repeat this processes for all relevant metadata columns and associated values

Usage

```
annotate_nomenclature_from_metadata(
  cell_set_information,
  metadata,
  metadata_columns,
  metadata_order = NULL,
 annotation_columns = rep("cell_set_preferred_alias", length(metadata_columns)),
  cluster_column = "cluster_label",
  append = TRUE
)
```

Arguments

cell_set_information
: the nomenclature table output from 'build_nomenclature_table' or related/downstream functions

metadata
: cell or cell type metadata table that includes the columns to annotate

metadata_columns
: a character vector of column names corresponding to the metadata fields to add annotations

metadata_order
: optional character vector of column names indicating the order to include metadata. If supplied, must be the same lengthh as "metadata_columns"

annotation_columns
: character vector indicating which column to annotate for each metadata column supplied (default is is "cell_set_preferred_alias")

cluster_column
: column name in "metadata" that corresponds to values in the "cell_set_preferred_alias" column of "cell_set_information"

append
: If TRUE (default), it will append info; if FALSE, it will skip cases where there is already an entry

Value

An updated nomenclature table with new cell sets and updated annotations based on requested metadata

---

apply_CCN   *Apply CCN (**BETA**)*

Description

This **BETA** function is a wrapper for most of the other functions in the CCN library. It takes as input whatever information is available (e.g., dend, nomenclature, cell_assignment, metadata) and uses this to try and output a standard nomenclature table and other CCN outputs. Please see the "Applying CCN to an existing taxonomy: one function" vignette for examples of how to use this function to apply the CCN in various contexts.

Usage

```
apply_CCN(
  dend = NULL,
  nomenclature = NULL,
  cell_assignment = NULL,
  metadata = NULL,
  first_label = setNames("All", 1),
  taxonomy_id = paste0("CCN", format(Sys.time(), "%Y%m%d"), 0),
  taxonomy_author = "Unspecified",
  taxonomy_citation = "",
  structure = "neocortex",
  ontology_tag = NULL,
  metadata_columns = c("subclass_label"),
  metadata_order = NULL,
```

```
  annotation_columns = rep("cell_set_preferred_alias", length(metadata_columns)),
  cluster_column = "cluster_label",
  append_metadata = FALSE,
  ccn_filename = "nomenclature.zip"
)
```

Arguments

dend	dendrogram of cell types to annotate. At least one of dend, nomenclature, cell_assignment, or metadata must be provided.
nomenclature	the nomenclature table output from 'build_nomenclature_table' or related/downstream functions.
cell_assignment	a named vector linking each unique cell id ('names(cell_assignment)') to their cell type assignments ('cell_assignment')
metadata	cell or cell type metadata table that includes the columns to annotate
first_label	a named vector used as prefix for cell_set_label
taxonomy_id	unique accession ID for the taxonomy also used to prefix the cell sets accessions. Defaults to 'CCN[YYYYMMDD]0'
taxonomy_author	the name of a point person for this taxonomy
taxonomy_citation	permanent data identifier corresponding to the taxonomy (or default="" if none). Ideally the DOI for a relevant publication.
structure	the location in the brain (or body) from where the data in the taxonomy was collected
ontology_tag	a standard ontology term (e.g., from UBERON) for the 'structure', or "none" if unavailable. NULL (default) attempts to find one in UBERON using 'find_ontology_terms'.
metadata_columns	a character vector of column names corresponding to the metadata fields to add annotations. Only used if "metadata" is provided
metadata_order	optional character vector of column names indicating the order to include metadata. If supplied, must be the same length as "metadata_columns". Only used if "metadata" is provided
annotation_columns	character vector indicating which column to annotate for each metadata column supplied (default is is "cell_set_preferred_alias"). Only used if "metadata" is provided
cluster_column	column name in "metadata" that corresponds to values in the "cell_set_preferred_alias" column of "cell_set_information". Only used if "metadata" is provided
append_metadata	If TRUE, it will append info; if FALSE (default), it will skip cases where there is already an entry. Only used if "metadata" is provided
ccn_filename	file name for zip file with final CCN files containing the same information that is returned. Will output to current working directory unless full path is specified. Will not output anywhere if set to NULL.

Value

a list containing the three CCN standard outputs:

Describe is optional and can go after and param or return

cell_set_information Final nomenclature table where rows correspond to cell sets and columns correspond to standard CCN columns.

final_dendrogram A dendrogram updated with node labels and CCN annotations, if dend was provided. This is what is output in dend.json

mapping A data frame where the first columns corresponds to each cell's unique ID (if cell_assignment or metadata is provided) and the remaining columns correspond to cell sets. Entries are either 0 = cell unassigned to cell set or 1 = cell assigned to cell set.

---

build_nomenclature_table

Build initial nomenclature table from dendrogram

Description

Take a standard R dendrogram variable and some taxonomy metadata and generate an initial nomenclature table with one cell set corresponding to each dendrogram node and leaf, retaining leaf labels as preferred aliases.

Usage

```
build_nomenclature_table(
  dend,
  first_label = setNames("All", 1),
  taxonomy_id = paste0("CCN", format(Sys.time(), "%Y%m%d"), 0),
  taxonomy_author = "Unspecified",
  taxonomy_citation = "",
  structure = "neocortex",
  ontology_tag = "UBERON:0001950"
)
```

Arguments

dend	dendrogram of cell types to annotate
first_label	a named vector used as prefix for cell_set_label
taxonomy_id	unique accession ID for the taxonomy also used to prefix the cell sets accessions. Defaults to 'CCN[YYYYMMDD]0'
taxonomy_author	
	the name of a point person for this taxonomy
taxonomy_citation	
	permanent data identifier corresponding to the taxonomy (or default="" if none). Ideally the DOI for a relevant publication.
structure	the location in the brain (or body) from where the data in the taxonomy was collected
ontology_tag	a standard ontology term (e.g., from UBERON) for the 'structure', or "none" if unavailable. The function 'find_ontology_terms' can be used to find one.

Value

a list containing two dendrogram versions and the initial nomenclature table:

   # Describe is optional and can go after and param or return

cell_set_information Initial nomenclature table where rows correspond to cell sets (e.g., dendrogram leaves and nodes) and columns correspond to standard CCN columns.

initial_dendrogram A slightly clean version of the input 'dend'.

updated_dendrogram A dendrogram updated with node labels. This is useful to visualize when manually annotating dendrograms.

---

cell_assignment_from_dendrogram

Assign cells to cell sets from dendrogram

Description

Automatically link each cell to each cell set that is available in the dendrogram to produce a table of the probabilities of each cell mapping to each cell type. In this case we define hard probabilities (0 = unassigned to cell set; 1 = assigned to cell set) but this could be adapted to reflect real probabilities calculated elsewhere.

Usage

```
cell_assignment_from_dendrogram(
  dend,
  samples,
  cell_id,
  mapping = data.frame(sample_name = samples),
  continue = TRUE
)
```

Arguments

dend	dendrogram of annotated cell types
samples	a character vector of unique sample (e.g., cell or nucleus) names
cell_id	a character vector of cell_set_accession_ids that corresponds to each sample
mapping	do not set this parameter
continue	do not set this parameter

Value

A data frame where the first columns corresponds to the cell sample_name and the remaining columns correspond to cell sets from the dendrogram. Entries are either 0 = cell unassigned to cell set or 1 = cell assigned to cell set.

```
cell_assignment_from_groups_of_cell_types
```
Assign cells to cell sets from nomenclature table

Description

Automatically link each cell to each cell set that is included in a nomenclature table to produce a table of the probabilities of each cell mapping to each cell type. In this case we define hard probabilities (0 = unassigned to cell set; 1 = assigned to cell set) but this could be adapted to reflect real probabilities calculated elsewhere. Only cell sets not already included in input mapping table are added. NOTE: This function has some issues if more than one "first_label" is in use.

Usage

```
cell_assignment_from_groups_of_cell_types(
  updated_nomenclature,
  cell_id,
  mapping,
  verbose = TRUE
)
```

Arguments

updated_nomenclature	
	dendrogram of annotated cell types
cell_id	a character vector of cell_set_accession_ids that corresponds to each sample
mapping	A data frame where the first columns corresponds to the cell sample_name and the remaining columns correspond to cell sets (e.g., the output from 'cell_assignment_from_dendrogr
verbose	A logical indicating whether to output cell set accession IDs of new annotations to screen (default=TRUE)

Value

An updated "mapping" variable with additional cells sets added from nomenclature table. Entries are either 0 = cell unassigned to cell set or 1 = cell assigned to cell set.

```
define_child_accessions
```
Label cell set children

Description

Create an additional tag called child_cell_set_accessions, which is a "|"-separated character vector indicating all of the child set sets. This is calculated by parsing the cell_set_label tags and is helpful for integration into downstream ontology use cases.

Usage

```
define_child_accessions(nomenclature)
```

Arguments

nomenclature	the nomenclature table output from 'build_nomenclature_table' or related/downstream functions

Value

An updated nomenclature table with a "child_cell_set_accessions" column appended

---

dend_to_list   *Convert R dendrogram to a list*

Description

Converts an R dendrogram to a list format. This step is necessary for outputting a dendrogram in json format. NOTE: If this function crashes, the "omit_names" variable may need to be updated to exclude additional variables in dend.

Usage

```
dend_to_list(dend, omit_names = c("markers", "markers.byCl", "class"))
```

Arguments

dend	any R dendrogram of annotated cell types
omit_names	character vector of attributes to exclude from the conversion to list format. This is necessary because attributes in complex formats sometimes cannot be properly converted without the function crashing

Value

The R dendgram information in list format, with all but the omitted attributes

---

find_ontology_terms   *Find ontology terms by querying the Ontology Lookup Service directly*
                              from R. This is useful for quickly finding UBERON (or other) ontology
                              IDs for inclusion in the CCN. Note: this is a wrapper for functions
                              from library 'rols' (http://lgatto.github.com/rols/) Please cite as spec-
                              ified in 'citation("rols")' if you use this function.

Description

Find ontology terms by querying the Ontology Lookup Service directly from R. This is useful for quickly finding UBERON (or other) ontology IDs for inclusion in the CCN. Note: this is a wrapper for functions from library 'rols' (http://lgatto.github.com/rols/) Please cite as specified in 'citation("rols")' if you use this function.

Usage

```
find_ontology_terms(query, exact = TRUE, ontology = "UBERON", ...)
```

Arguments

`query`	Term you want to search for in the ontology (e.g., "neocortex")
`exact`	Return only exact matches to input term (TRUE; default) or that's that are close
`ontology`	Which ontology to search. Default is "UBERON". "" will search all ontologies
`...`	Additional parameters to 'OlsSearch'

Value

what is returned
Describe is optional and can go after and param or return

name description

`merge_cell_set_labels` *Merge cell set labels*

Description

Takes as input a vector of cell set labels and outputs a single character that merges the cell set labels together in a specific format.

Usage

```
merge_cell_set_labels(cell_set_label_vector, sep = " ")
```

Arguments

`cell_set_label_vector`
character vector of cell set labels for the form "prefix value"

Value

character corresponding to merged value

`overwrite_dend_node_labels`

Overwrite dendrogram node labels

Description

Overwrites node (but not leaf) labels of a dendrogram with unique labels of the format "n##". This is useful for visualization of the dendrogram for manual annotation of cell sets. Optionally also change leaf labels.

Usage

```
overwrite_dend_node_labels(dend, n = 1, lab = labels(dend))
```

Arguments

dend	any R dendrogram of annotated cell types
n	starting value for numbering (default of 1 does not need to be changed)
lab	leaf labels to write, defaults to existing leaf labels

Value

R dendrogram with new node (and potentially leaf) labels

---

plot_dend   *Plot dendrogram*

Description

Function for plotting a dendrogram in a way that displays node labels. This function is copied from 'scrattch.hicat': https://github.com/AllenInstitute/scrattch.hicat

Usage

```
plot_dend(dend, dendro_data = NULL, node_size = 1, r = c(-0.1, 1))
```

Arguments

dend	any R dendrogram of annotated cell types
dendro_data	(I don't know)
node_size	size of nodes (default = 1)
r	(I don't know)

Value

A plot of the dendrogram in 'ggplot2' format

---

update_dendrogram_with_nomenclature
                        Add nomenclature annotations to dendrogram

---

Description

This code will take the information from the table above and add it to the initial dendrogram object. When plotted the only visible difference will be that the new cell set alias names (if any) will show up to replace the n## labels from the initial plot. However, ALL of the meta-data read in from the table will be added to the relevant nodes or leafs.

Usage

```
update_dendrogram_with_nomenclature(
  dend,
  cell_set_information,
  current_label = "original_label",
  new_label = "cell_set_preferred_alias"
)
```

Arguments

dend   dendrogram of cell types to annotate

cell_set_information

any table or data frame with information to annotate a dendrogram, typically the nomenclature table output from 'build_nomenclature_table'

current_label   column name in 'cell_set_information' that contains dendrogram labels (default of "original_label" should be retained if running this function after 'build_nomenclature_table')

new_label   column name holding desired dendrogram labels after running script (default "cell_set_preferred_alias")

Value

An annotated dendrogram with updated labels

Index