Package 'flowCL'

April 12, 2022

Type Package

Title Semantic labelling of flow cytometric cell populations

Version 1.32.0

Date 2018-12-01

Author Justin Meskas, Radina Droumeva

Maintainer Justin Meskas <justinmeskas@gmail.com>

Depends R (>= 3.4), Rgraphviz, SPARQL

Imports methods, grDevices, utils, graph

Description Semantic labelling of flow cytometric cell populations.

biocViews FlowCytometry, ImmunoOncology

Suggests RUnit, BiocGenerics

License Artistic-2.0

LazyLoad yes

git_url https://git.bioconductor.org/packages/flowCL

git_branch RELEASE_3_14

git_last_commit 29ae7ec

git_last_commit_date 2021-10-26

Date/Publication 2022-04-12

R topics documented:

---

flowCL-package   *Semantic labelling of flow cytometric cell populations.*

Description

flowCL uses SPARQL to match a phenotype to a cell type from the cell ontology. If the match is not unique, then the best alternative is returned.

Details

Package:	flowCL
Type:	Package
License:	Artistic-2.0
LazyLoad:	yes
Depends:	Rgraphviz, SPARQL

flowCL executes queries against the Cell Ontology (CL), available at http://cellontology.org. The CL file is hosted on a triplestore, i.e., a database for storage and retrieval of Resource Description Framework (RDF) triples. The SPARQL endpoint at http://cell.inference.me:7200/repositories/CL (old endpoint: http://cell.ctde.net:8080/openrdf-sesame/repositories/CL) is used to execute the SPARQL queries retrieving the correct matches from the CL. While other SPARQL endpoints can be used, users should be aware that in our case the CL file has been reasoned upon, and resulting extra inferred axioms have been added to the triplestore, providing a more complete result set.

Author(s)

Maintainer: Justin Meskas <justinmeskas@gmail.com>

Authors: Justin Meskas, Radina Droumeva

Examples

```
# # Load a pre-loaded archive. Skipping this chuck will cause flowCL to
# ## slowly build a new one.
# flowCL("archive")
#
# # Simple two marker example
# Res <- flowCL("CCR7+CD45RA+")
# tmp <- Res$'CCR7+CD45RA+'
# plot(tmp[[1]], nodeAttrs=tmp[[2]], edgeAttrs=tmp[[3]], attrs=tmp[[4]])
# Res$Table
#
# # Exact match example
# Res <- flowCL("CD3-CD19-CD20-CD14+", Verbose = TRUE, OntolNamesTD = TRUE)
# tmp <- Res$'CD3-CD19-CD20-CD14+'
# plot(tmp[[1]], nodeAttrs=tmp[[2]], edgeAttrs=tmp[[3]], attrs=tmp[[4]])
# Res$Table
```

```
#
# # Cell Label Example
# x <-"CD3-CD19-CD20-CD14+"
# Res <- flowCL(x)
# Res$Cell_Label[[x]][[1]]
#
# # As a secondary way to view the results,
# ## see "[current directory]/flowCL_results/".
# # Figures created called tree_(phenotype).pdf give the cell hierarchy
# ## dependent on the markers in the phenotype.
# # A list of results from Res$Table are stored in listPhenotypes.csv.
```

---

flowCL   *Semantic labelling of flow cytometric cell populations.*

Description

flowCL uses queries to match a phenotype to a cell type from the cell ontology. If the match is not unique, then the best alternative is returned.

Usage

```
flowCL ( MarkerList = "HIPC", ExpMrkrLst = NULL, Indices = NULL,
    Verbose = FALSE, KeepArch = TRUE, MaxHitsPht = 5,
    OntolNamesTD = FALSE, ResetArch = FALSE, VisualSkip = FALSE )
```

Arguments

MarkerList
: A list of phenotypes to query the cell ontology with (Ex. "CD3+CD4-CD8+" as a single element of the list). There is an option to use a preloaded list. This preloaded list can be used by inputting "HIPC" into MarkerList. This will first query all the individual markers, then will query all the common HIPC phenotypes.

ExpMrkrLst
: A list of all of the phenotypes that were used in the experiment. This will be used to inform the user that a certain marker could have been used to further identify the correct population. If ExpMrkLst is left blank, flowCL will define ExpMrkLst as each phenotype in MarkerList. If ExpMrkrLst has only one element, flowCL will define every ExpMrkrLst phenotype as being the same as the one input.

Indices
: A vector of indices that dictate which of the MarkerList phenotypes will be queried. If left blank, all phenotypes in the list will be queried.

Verbose
: A logical value that dictates if computational information is printed while the code is running. The default is FALSE.

ResetArch
: A logical value that dictates if the archive folder, "flowCL_results", is deleted before the queries take place. This will increase the simulation time, but should be done every once in a while to account for updates from the ontology. The default is FALSE.

KeepArch	A logical value that dictates if the archive folder, "flowCL_results", is deleted after the queries take place. Set to FALSE to insure there is no unwanted files being stored on the hard drive. The default is TRUE.
MaxHitsPht	An integer for the maximum number of cell types that are returned per phenotype queried. The default is 5.
OntolNamesTD	A logical value that controls if the phenotypes in the tree diagrams(TD) are short names (ex. CD4) or the long ontology names (ex. CD4 molecule). Short names are used if OntolNamesTD is FALSE, while long ontology names are used if TRUE. The default is FALSE.
VisualSkip	A logical value that controls if the visualization step is skipped or not. TRUE is for skip, while FALSE is for no skip. The default is FALSE.

Details

flowCL executes queries against the Cell Ontology (CL), available at http://cellontology.org. The CL file is hosted on a triplestore, i.e., a database for storage and retrieval of Resource Description Framework (RDF) triples. The SPARQL endpoint at http://cell.inference.me:7200/repositories/CL (old endpoint: http://cell.ctde.net:8080/openrdf-sesame/repositories/CL) is used to execute the SPARQL queries retrieving the correct matches from the CL. While other SPARQL endpoints can be used, users should be aware that in our case the CL file has been reasoned upon, and resulting extra inferred axioms have been added to the triplestore, providing a more complete result set.

Value

A list containing N + 5 elements. Where N is the number of phenotypes queried. Each of these N elements contains information for plotting the results. The other five elements show the cell labels (Cell_Labels), the matching markers in a list form (Marker_Groups) and a bracket form (Markers), ranking scores (Ranking) and a table (Table). The cell labels element lists the cell labels in order of highest score based on their ranking, which is in a form easily extracted and used by other R packages and functions. Marker_Groups and Markers list markers that were queried and that are part of a certain cell type. In Markers these markers are displayed in the form of A B (C) [D]. A and B together make up the markers input for the query. B, C and D together are the markers that make up the definition of the particular cell type. C lists the markers that were pat of the experiment that were not part of the query, while D lists all other markers that make up the cell type that were not part of the experimental markers. A lists all the markers in the input for the query that were not required for that particular cell type. The table is a list of all the related information of each phenotype queried. This table is mainly for users to see the results in R.

Author(s)

Maintainer: Justin Meskas <justinmeskas@gmail.com>

Authors: Justin Meskas, Radina Droumeva

Examples

```
# # Load a pre-loaded archive. Skipping this chuck will cause flowCL to
# ## slowly build a new one.
# flowCL("archive")
```

```
#
# # Simple two marker example
# Res <- flowCL("CCR7+CD45RA+")
# tmp <- Res$'CCR7+CD45RA+'
# plot(tmp[[1]], nodeAttrs=tmp[[2]], edgeAttrs=tmp[[3]], attrs=tmp[[4]])
# Res$Table
#
# # Exact match example
# Res <- flowCL("CCR7+CD45RA+CD8+", Verbose = TRUE, OntolNamesTD = TRUE)
# tmp <- Res$'CCR7+CD45RA+CD8+'
# plot(tmp[[1]], nodeAttrs=tmp[[2]], edgeAttrs=tmp[[3]], attrs=tmp[[4]])
# Res$Table
#
# # Cell Label Example
# x <-"CCR7+CD45RA+CD8+"
# Res <- flowCL(x)
# Res$Cell_Label[[x]][[1]]
#
# # As a secondary way to view the results,
# ## see "[current directory]/flowCL_results/".
# # Figures created called tree_(phenotype).pdf give the cell hierarchy
# ## dependent on the markers in the phenotype.
# # A list of results from Res$Table are stored in listPhenotypes.csv.
```

---

flowCL_archive   *A list of data used to build a local archive.*

Description

A list of data used to build a local archive.

Usage

```
data(flowCL_archive)
```

Value

Loads the a pre-built archive.

Author(s)

Justin Meskas

test.flowCL.connection

A function to check that a connection exists between the user and the ontology server.

Description

A function to check that a connection exists between the user and the ontology server.

Usage

```
test.flowCL.connection()
```

Value

The date of the lastest update of the cell ontology is returned.

Author(s)

Justin Meskas

Examples

```
test.flowCL.connection()
```

Index