Package 'slinky'

December 2, 2015

Type Package

Title Facilitates storage and analysis of LINCS data

Version 1.0

Date 2015-11-30

Author Eric J. Kort

Maintainer Who to complain to <eric.kort@vai.org>

Depends R (>= 3.1.0), rhdf5, httr, rjson, dplyr, tidyjson

Collate 'slinky.R'

Suggests knitr, testthat

VignetteBuilder knitr

LazyData true

Description Loads L1000 data from LINCS level 2 data file into document store and facilitates analysis of that data including calcuation of z-scores and enrichment analysis.

License MIT

NeedsCompilation no

R topics documented:

`Slinky-class` *Slinky Class*

Description

A class to facilitate storage and analysis of the level 2 data from the LINCS project. This class does not provide any of the data (which must be obtained under individual agreement with LINCS). However, once obtained, working with their large binary data files is facilitated by this class.

Details

Make LINCS analysis fun.

Fields

`.ip` (Private) IP address of your LINCS REST server, set with setIp. Default is `127.0.0.1`.

`.port` (Private) Port of your LINCS REST server, set with setPort. Default is `8080`.

`loglevel` How much information should we log? Options are `all`, `error`, `warn`, `none`. Note that `all` or `warn` will result in logfile ~120MB in size. Default is `all`.

`logfile` Where to store the log. Default is `log.txt`.

`silent` Should logging info only be written to file (and not to stdout)? Default is FALSE. Irrelevant if `loglevel` is `none`.

`maxtries` If http request fails, how many times should we retry before moving on? Default is `3`.

Methods

`calc(cluster = NULL)` Calculate zscores and stores them in the document store.

Parameters:

- `cluster` An optional cluster object to use for calculations. Each node must have this package installed on it.

Return Value: None. Called for side effect of populating document store with zscores

`getPlateControls(id)` Fetch normalized expression data for control samples from same plate as id and treated only with the vehicle used for sample id.

Parameters:

- `id` Id of instance for which control data is desired.

Return Value: dataframe containing normalized gene expression for controls.

`loadLevel2(gctxfile = "gex_epsilon_n1429794x978.gctx", col)` Load data for specified column from hdf5 formatted file (.gctx) from LINCS Fetch into your document store via RESTful interface.

Parameters:

- `gctxfile` Path to level 2 gctx file. Default is `./gex_epsilon_n1429794x978.gctx`.
- `gctxfile` Path to instance info file. Default is `./inst.info`.

Return Value: None. Loads data into document store.

`setIp(ip = "127.0.0.1")` Set IP and redefine endpoint

Parameters:

- `ip` The IP address of the your LINCS REST server, default is `127.0.0.1`.

Return Value: none

`setPort(port = "8080")` Set port and redefine endpoint

Parameters:

- `port` The port of the your LINCS REST server, default is `8080`.

Return Value: none

`ZbyPlate(id)` Calculate zscores for specific instance relative to mean of appropriate vehicle controls on same plate.

Parameters:

- `id` Id of instance for which scores are desired.

Return Value: Vector containing robust z-scores for each gene.

Examples

```
lincs <- Slinky$new()
lincs$calc()
```

Index