Package ‘genieBPC’

October 27, 2022

Type Package
Title Project GENIE BioPharma Collaborative Data Processing Pipeline
Version 1.0.1
Description The American Association Research (AACR) Project Genomics Evidence Neoplasia Information Exchange (GENIE) BioPharma Collaborative represents a multi-year, multi-institution effort to build a pan-cancer repository of linked clinico-genomic data. The genomic and clinical data are provided in multiple releases (separate releases for each cancer cohort with updates following data corrections), which are stored on the data sharing platform ‘Synapse’ <https://www.synapse.org/>.
The ‘genieBPC’ package provides a seamless way to obtain the data corresponding to each release from ‘Synapse’ and to prepare datasets for analysis.
License MIT + file LICENSE

BugReports https://github.com/GENIE-BPC/genieBPC/issues
Depends R (>= 3.4)
Imports cli (>= 2.5.0), dplyr (>= 1.0.6), dtplyr (>= 1.1.0), htr, jsonlite, purrr (>= 0.3.4), rlang (>= 1.0.0), stringr (>= 1.4.0), sunburstR, tibble (>= 3.1.2), tidyR
Suggests covr (>= 3.5.1), ggplot2 (>= 3.3.5), gt (>= 0.3.0), gts{summary (>= 1.5.2), knitr (>= 1.33), magrittr (>= 2.0.1), plotly (>= 4.10.0), rmarkdown (>= 2.8), testthat (>= 3.0.0), markdown, spelling
VignetteBuilder knitr
Config/testthat/edition 3
Encoding UTF-8
LazyData TRUE
RoxygenNote 7.2.1
URL https://genie-bpc.github.io/genieBPC/
Language en-US
check_genie_access

Description

Check Access to GENIE Data

Usage

check_genie_access(username = NULL, password = NULL)

Arguments

username

'Synapse' username. If NULL, package will search package environment for
"username". If not found, package will look in environmental variables for
'SYNAPSE_USERNAME'.

password

'Synapse' password. If NULL, package will search package environment for
"password". If not found package will search environmental variables for 'SYNAPSE_PASSWORD'.

**create_analytic_cohort**

**Value**

A success message if you are able to access GENIE BPC data; otherwise an error

**Author(s)**

Karissa Whiting

**Examples**

```r
## Not run:
# if credentials are saved:
check_genie_access()

## End(Not run)
```

```r
create_analytic_cohort

*Select cohort of patients for analysis*
```

**Description**

This function allows the user to create a cohort from the GENIE BPC data based on cancer diagnosis information such as cancer cohort, treating institution, histology, and stage at diagnosis, as well as cancer-directed regimen information including regimen name and regimen order. This function returns each of the clinical and genomic data files subset on the patients that met criteria for the analytic cohort. Documentation regarding the structure and contents of each file can be found in the Analytic Data Guide corresponding to each data release, as well as in the Clinical Data Structure vignette.

**Usage**

```r
create_analytic_cohort(
  data_synapse,
  index_ca_seq = 1,
  institution,
  stage_dx,
  histology,
  regimen_drugs,
  regimen_type = "Exact",
  regimen_order,
  regimen_order_type,
  return_summary = FALSE
)
```
Arguments

`data_synapse` The item from the nested list returned from `pull_data_synapse()` that corresponds to the cancer cohort of interest.

`index_ca_seq` Index cancer sequence. Default is 1, indicating the patient’s first index cancer. The index cancer is also referred to as the BPC Project cancer in the GENIE BPC Analytic Data Guide; this is the cancer that met the eligibility criteria for the project and was selected at random for PRISSMM phenomic data curation. Specifying multiple index cancer sequences, e.g. `index_ca_seq = c(1, 2)` will return index cancers to patients with 1 index cancer and will return the first AND second index cancers to patients with multiple.

`institution` GENIE BPC participating institution. Must be one of "DFCI", "MSK", "UHN", or "VICC" for NSCLC cohorts; must be one of "DFCI", "MSK", "VICC" for CRC and BrCa. Default selection is all institutions. This parameter corresponds to the variable ‘institution’ in the Analytic Data Guide.

`stage_dx` Stage at diagnosis. Must be one of "Stage I", "Stage II", "Stage III", "Stage I-III NOS", "Stage IV". The default selection is all stages. Note that if this parameter is specified, any cases that are missing stage information are automatically excluded from the resulting cohort. This parameter corresponds to the variable ‘stage_dx’ in the Analytic Data Guide.

`histology` Cancer histology. For all cancer cohorts except for BrCa (breast cancer), this parameter corresponds to the variable ‘ca_hist_adeno_squamous’ and must be one of "Adenocarcinoma", "Squamous cell", "Sarcoma", "Small cell carcinoma", "Carcinoma", "Other histologies/mixed tumor". For BrCa, this parameter corresponds to the variable ‘ca_hist_brca’ and must be one of "Invasive lobular carcinoma", "Invasive ductal carcinoma", "Other histology". The default selection is all histologies. Note that if this parameter is specified, any cases that are missing histology information are automatically excluded from the resulting cohort.

`regimen_drugs` Vector with names of drugs in cancer-directed regimen, separated by a comma. For example, to specify a regimen consisting of Carboplatin and Pemetrexed, specify `regimen_drugs = "Carboplatin, Pemetrexed"`. Acceptable values are found in the ‘drug_regimen_list’ dataset provided with this package. This parameter corresponds to the variable ‘regimen_drugs’ in the Analytic Data Guide.

`regimen_type` Indicates whether the regimen(s) specified in ‘regimen_drugs’ indicates the exact regimen to return, or if regimens containing the drugs listed in ‘regimen_drugs’ should be returned. Must be one of "Exact" or "Containing". The default is "Exact".

`regimen_order` Order of cancer-directed regimen. If multiple drugs are specified, ‘regimen_order’ indicates the regimen order for all drugs; different values of ‘regimen_order’ cannot be specified for different drug regimens. If multiple values are specified, e.g. c(1, 2), then drug regimens that met either order criteria are returned.

`regimen_order_type` Specifies whether the ‘regimen_order’ parameter refers to the order of receipt of the drug regimen within the cancer diagnosis (across all other drug regimens; "within cancer") or the order of receipt of the drug regimen within the times that that drug regimen was administered (e.g. the first time carboplatin pemetrexed..."
create_analytic_cohort

was received, out of all times that the patient received carboplatin pemetrexed; "within regimen"). Acceptable values are "within cancer" and "within regimen".

return_summary  Specifies whether a summary table for the cohort is returned. Default is FALSE. The `gtsummary` package is required to return a summary table.

Details

See the create_analytic_cohort vignette for further documentation and examples.

Value

A list of data frames containing clinical and next generation sequencing information for patients that met the specified criteria. Optionally, if return_summary = TRUE, the list also includes summary tables for the number of records per dataset ("tbl_overall_summary") as well as tables of key cancer diagnosis ("tbl_cohort"), cancer-directed regimen ("tbl_drugs") and next generation sequencing ("tbl_ngs") variables.

Author(s)

Jessica Lavery

Examples

# Examples using package test data
# Example 1 ----------------------------------
# Create a cohort of all patients with stage IV NSCLC adenocarcinoma and # obtain all of their corresponding clinical and genomic data

ex1 <- create_analytic_cohort(
  data_synapse = genieBPC::nsclc_test_data,
  stage_dx = "Stage IV",
  histology = "Adenocarcinoma"
)

names(ex1)

# Example 2 ----------------------------------
# Create a cohort of all NSCLC patients who received Cisplatin, # Pemetrexed Disodium or Cisplatin, Etoposide as their first drug regimen # for their first index NSCLC

ex2 <- create_analytic_cohort(
  data_synapse = genieBPC::nsclc_test_data,
  regimen_drugs = c("Cisplatin, Pemetrexed Disodium", "Cisplatin, Etoposide"),
  regimen_order = 1,
  regimen_order_type = "within cancer"
)
# Example 3 ----------------------------------
# Create a cohort of all NSCLC patients who received Cisplatin, Pemetrexed Disodium at any time throughout the course of treatment for their cancer diagnosis, but in the event that the patient received the drug multiple times, only select the first time.

ex3 <- create_analytic_cohort(
  data_synapse = genieBPC::nsclc_test_data,
  regimen_drugs = c("Cisplatin, Pemetrexed Disodium"),
  regimen_order = 1,
  regimen_order_type = "within regimen"
)

# Example 4 ----------------------------------
# Using create_analytic_cohort with pull_data_synapse
nsclc_2_0 <- pull_data_synapse("NSCLC", version = "v2.0-public")
ex4 <- create_analytic_cohort(
  data_synapse = nsclc_2_0$NSCLC_v2.0,
  regimen_drugs = c("Cisplatin, Pemetrexed Disodium"),
  regimen_order = 1,
  regimen_order_type = "within regimen"
)

---

**drug_regimen_list**  
*List of Drug Regimen Names by Cohort*

**Description**
A dataset containing the cancer-directed drug names and their synonyms.

**Usage**

```
drug_regimen_list
```

**Format**
A table for cancer-directed drugs associated with each cancer cohort:

- **cohort** GENIE BPC Project cancer. Must be one of "NSCLC" (non-small cell lung cancer), "CRC" (colorectal cancer), or "BrCa" (breast cancer). Future cohorts will include "PANC" (pancreatic cancer), "Prostate" (prostate cancer), and "BLADDER" (bladder cancer).
- **drug_name** Name of generic/ingredient cancer-directed drug
- **drug_name_full** Name of generic/ingredient cancer-directed drug with associated synonyms in parentheses ...
**drug_regimen_sunburst**  Visualize drug regimen sequences in a sunburst plot

**Description**

This function allows the user to visualize the complete treatment course for selected cancer diagnoses.

**Usage**

```
drug_regimen_sunburst(data_synapse, data_cohort, max_n_regimens = NULL, ...)
```

**Arguments**

- `data_synapse`: The item from the nested list returned from `pull_data_synapse()`
- `data_cohort`: The list returned from the `create_analytic_cohort()` function call
- `max_n_regimens`: The maximum number of regimens displayed in the sunburst plot
- `...`: Additional parameters passed to `sunburstR::sunburst()`

**Details**

See the [drug_regimen_sunburst vignette](#) for additional details and examples.

**Value**

Returns data frame ‘treatment_history’ and interactive plot ‘sunburst_plot’

**Examples**

```r
# Example 1 ----------------------------------
# Example using package test data
geneBPC::nsclc_test_data

# get clinico-genomic files for a specific cohort
nsclc_sub <- create_analytic_cohort(data_synapse = genieBPC::nsclc_test_data,
    stage_dx = c("Stage III", "Stage IV")
)

# create sunburst plot
ex1 <- drug_regimen_sunburst(data_synapse = nsclc_test_data,
    data_cohort = nsclc_sub,
    max_n_regimens = 3
)

# Example 2 ----------------------------------
# using pull_data_synapse
sclc_2_0 <- pull_data_synapse("NSCLC", version = "v2.0-public")
```
nsclc_stg_iv <- create_analytic_cohort(
  data_synapse = nsclc_2_0$NSCLC_v2.0,
  stage = "Stage IV"
)

ex2 <- drug_regimen_sunburst(
  data_synapse = nsclc_2_0$NSCLC_v2.0,
  data_cohort = nsclc_stg_iv,
  max_n_regimens = 3
)

---

### fetch_samples

#### Description

This function links patients in a cohort (created by `create_analytic_cohort()`) with their corresponding next generation sequencing (NGS) reports, which can be used to link to the corresponding GENIE genomic data.

#### Usage

```r
fetch_samples(cohort, data_synapse, df_record_ids)
```

#### Arguments

- **cohort**: GENIE BPC Project cancer. Must be one of "NSCLC", "CRC", or "BrCa"
- **data_synapse**: The item from the nested list returned from `pull_data_synapse()`
- **df_record_ids**: NGS data frame from the `create_analytic_cohort()` function. Must contain variables: cohort, record_id and ca_seq.

#### Details

Subset cancer panel test data to patients in the cohort of interest

#### Value

The NGS reports for each patient, stored as the 'cohort_ngs' data frame of the create_analytic_cohort object

#### Author(s)

Axel Martin
### Genie Panels Included in GENIE BPC Data

**Description**
A dataset containing the name, assay identifier, and number of genes in each next-generation sequencing targeted panel included in GENIE BPC.

**Usage**
genie_panels

**Format**
A data frame with 12 rows and 3 variables:

- **Sequence.Assay.ID** Next-generation sequencing targeted panel assay identifier
- **Panel** Panel name
- **Genes** Number of genes included ...

### nsclc_test_data

**Description**
A named list of simulated NSCLC clinical data

**Usage**
nsclc_test_data

**Format**
A list of clinical data frames

- **pt_char** Patient characteristic data.frame
- **ca_dx_index** Index cancer diagnosis data.frame
- **ca_dx_non_index** Non-index cancer diagnosis data.frame
- **ca_drugs** Cancer directed-regimen data.frame
- **prismm_imaging** PRISSMM Imaging report data.frame
- **prismm_pathology** PRISSMM Pathology report data.frame
- **prismm_md** PRISSMM medical oncologist report data.frame
- **cpt** CPT/NGS data.frame
**pull_data_synapse**

Obtain clinical & genomic data files for GENIE BPC Project

**Description**

Function to access specified versions of clinical and genomic GENIE BPC data from Synapse and read them into the R environment. See the `pull_data_synapse` vignette for further documentation and examples.

**Usage**

```r
pull_data_synapse(
  cohort = NULL,
  version = NULL,
  download_location = NULL,
  username = NULL,
  password = NULL
)
```

**Arguments**

- **cohort**
  Vector or list specifying the cohort(s) of interest. Must be one of "NSCLC" (Non-Small Cell Lung Cancer), "CRC" (Colorectal Cancer), or "BrCa" (Breast Cancer).

- **version**
  Vector specifying the version of the data. Must be one of the following: "v1.1-consortium", "v1.2-consortium", "v2.1-consortium", "v2.0-public". When entering multiple cohorts, the order of the version numbers corresponds to the order that the cohorts are specified; the cohort and version number must be in the same order in order to pull the correct data. See examples below.

- **download_location**
  if 'NULL' (default), data will be returned as a list of dataframes with requested data as list items. Otherwise, specify a folder path to have data automatically downloaded there. When a path is specified, data are not read into the R environment.

- **username**
  'Synapse' username

- **password**
  'Synapse' password

**Value**

Returns a nested list of clinical and genomic data corresponding to the specified cohort(s).

**Authentication**

To access data, users must have a valid 'Synapse' account with permission to access the data set and they must have accepted any necessary 'Terms of Use'. Users must always authenticate themselves in their current R session. (see README: Data Access and Authentication)
for details). To set your 'Synapse' credentials during each session, call:

`set_synapse_credentials(username = "your_username", password = "your_password")`

If your credentials are stored as environmental variables, you do not need to call `set_synapse_credentials()` explicitly each session. To store authentication information in your environmental variables, add the following to your .Renviron file, then restart your R session (tip: you can use `usethis::edit_r_environ()` to easily open/edit this file):

- ‘SYNAPSE_USERNAME = <your-username>’
- ‘SYNAPSE_PASSWORD = <your-password>’

Alternatively, you can pass your username and password to each individual data pull function if preferred, although it is recommended that you manage your passwords outside of your scripts for security purposes.

Analytic Data Guides

Documentation corresponding to the clinical data files can be found on 'Synapse' in the Analytic Data Guides:

- NSCLC v1.1-Consortium Analytic Data Guide
- NSCLC v2.1-Consortium Analytic Data Guide
- NSCLC v2.0-Public Analytic Data Guide
- CRC v1.1-Consortium Analytic Data Guide
- CRC v1.2-Consortium Analytic Data Guide
- BrCa v1.1-Consortium Analytic Data Guide

Author(s)

Karissa Whiting, Michael Curry

Examples

```r
# Example 1 ----------------------------------
# Set up 'Synapse' credentials
set_synapse_credentials()

# Print available versions of the data
synapse_version(most_recent = TRUE)

# Pull version 2.0-public for non-small cell lung cancer
# and version 1.1-consortium for colorectal cancer data
ex1 <- pull_data_synapse(
  cohort = c("NSCLC", "BrCa"),
  version = c("v2.0-public", "v1.1-consortium")
)

names(ex1)
```
Regimen Abbreviations

**Description**
A dataset containing the cancer-directed drug regimens and their common abbreviations

**Usage**
regimen_abbreviations

**Format**
A table for cancer-directed drug regimens and their common abbreviations

- **regimen_drugs** List of all drugs in the regimen
- **abbreviation** Common name of drug regimen, e.g. FOLFOX ...

Select Unique NGS

**Description**
For patients with multiple associated next generation (NGS) sequencing reports, select one unique NGS report per patient for the purpose of creating an analytic dataset based on user-defined criterion, including OncoTree code, primary vs. metastatic tumor sample, and earliest vs. most recent sample. If multiple reports for a patient remain available after the user-defined specifications, or if no specifications are provided, the panel with the largest number of genes is selected by default. Sample optimization is performed in the order that the arguments are specified in the function, regardless of the arguments’ order provided by the user. Namely, the OncoTree code is prioritized first, sample type is prioritized second and finally the time is prioritized last. For patients with exactly one genomic sample, that unique genomic sample will be returned regardless of whether it meets the user-specified parameters. Running the select_unique_ngs() function will ensure that the resulting dataset returned by merging the next generation sequencing report data onto the cohort_ca_dx dataset returned by create_analytic_cohort() will maintain the structure of cohort_ca_dx (either one record per patient or one record per diagnosis). Currently, if multiple diagnoses per patient are returned from create_analytic_cohort(), using select_unique_ngs() will select a single NGS report per patient. In future iterations, this will be updated so that one NGS report per diagnosis can be selected.

**Usage**
select_unique_ngs(
    data_cohort,
    oncotree_code = NULL,
    sample_type = NULL,
    min_max_time = NULL
)
select_unique_ngs

Arguments

data_cohort output object of the create_analytic_cohort function.
oncotree_code character vector specifying which sample OncoTree codes to keep. See "cpt_oncotree_code" column of data_cohort argument above to get options.
sample_type character specifying which type of genomic sample to prioritize, options are "Primary", "Local" and "Metastasis". Default is to not select a NGS sample based on the sample type.
min_max_time character specifying if the first or last genomic sample recorded should be kept. Options are "min" (first) and "max" (last).

Details

Note that the NGS dataset serves as the link between the clinical and genomic data, where the NGS dataset includes one record per NGS report per patient, including the NGS sample ID that is used to link to the genomic data files. Merging data from the NGS report onto the analytic cohort returned from create_analytic_cohort() therefore allows users to utilize all clinical and genomic data available.

See the select_unique_ngs vignette for further documentation and examples.

Value

returns the 'cohort_ngs' object of the create_analytic_cohort with unique genomic samples taken from each patients.

Author(s)

Karissa Whiting

Examples

# Example 1 ----------------------------------
# Create a cohort of all patients with stage IV NSCLC of
# histology adenocarcinoma
nsclc_2_0 <- pull_data_synapse("NSCLC", version = "v2.0-public")
ex1 <- create_analytic_cohort(
  data_synapse = nsclc_2_0$NSCLC_v2.0,
  stage_dx = c("Stage IV"),
  histology = "Adenocarcinoma"
)

# select unique next generation sequencing reports for those patients
samples_data1 <- select_unique_ngs(
  data_cohort = ex1$cohort_ngs,
  sample_type = "Primary"
)

# Example 2 ----------------------------------
# Create a cohort of all NSCLC patients who received Cisplatin, Pemetrexed Disodium or Cisplatin, Etoposide as their first drug regimen
ex2 <- create_analytic_cohort(
  data_synapse = nsclc_2_0$NSCLC_v2.0,
  regimen_drugs = c("Cisplatin, Pemetrexed Disodium", "Cisplatin, Etoposide"),
  regimen_order = 1,
  regimen_order_type = "within regimen"
)
samples_data2 <- select_unique_ngs(
  data_cohort = ex2$cohort_ngs,
  oncotree_code = "NSCLCPD",
  sample_type = "Metastasis",
  min_max_time = "max"
)

---

**set_synapse_credentials**

*Connect to 'Synapse' API*

**Description**

This function sets 'Synapse' credentials for the user's current session.

**Usage**

`set_synapse_credentials(username = NULL, password = NULL)`

**Arguments**

- **username**
  - 'Synapse' username. If NULL, package will search environmental variables for 'SYNAPSE_USERNAME'.

- **password**
  - 'Synapse' password. If NULL, package will search environmental variables for 'SYNAPSE_PASSWORD'.

**Details**

To access data, users must have a valid 'Synapse' account with permission to access the data set and they must have accepted any necessary 'Terms of Use'. Users must authenticate themselves in their current R session. (see https://genie-bpc.github.io/genieBPC/README 'Data Access and Authentication' for details). To set your 'Synapse' credentials during each session, call: `set_synapse_credentials(username = "your_username", password = "your_password")`.

If your credentials are stored as environmental variables, you do not need to call `set_synapse_credentials()` explicitly each session. To store authentication information in your environmental variables, add the
following to your .Renviron file, then restart your R session (tip: you can use `usethis::edit_r_environ()` to easily open/edit this file):

- `SYNAPSE_USERNAME = <your-username>`
- `SYNAPSE_PASSWORD = <your-password>`

Alternatively, you can pass your username and password to each individual data pull function if preferred, although it is recommended that you manage your passwords outside of your scripts for security purposes.

**Value**

A success message if your credentials are valid for 'Synapse' platform; otherwise an error

**Author(s)**

Karissa Whiting

**Examples**

```r
## Not run:
set_synapse_credentials(
  username = "your-username",
  password = "your-password"
)
## End(Not run)
```

---

**synapse_tables**

'Synapse' table IDs

**Description**

A dataset containing the 'Synapse' table IDs for each clinical dataset in GENIE BPC.

**Usage**

`synapse_tables`

**Format**

A lookup table for 'Synapse' clinical data table IDs:

- `cohort` GENIE BPC Project Cohort
- `df` Clinical dataset
- `version` Release version
- `synapse_id` 'Synapse' table ID for each dataset
- `release_date` Month and year of data release ...
Description

GENIE BPC data are updated periodically to add variables and reflect additional data cleaning. Each time the data are updated the data release version number is incremented. The ‘synapse_version()’ function will get available version numbers for each cohort to help the user determine what is the most recent version for each cohort.

Usage

synapse_version(most_recent = FALSE)

Arguments

most_recent Indicates whether the function will return only the most recent version number for each cohort (‘most_recent’ = TRUE) or all available version numbers for each cohort (‘most_recent’= FALSE)

Details

Specifies the version numbers available for each cancer cohort. Version numbers are specified as part of the call to ‘pull_data_synapse()’.

Value

Returns a table containing the available versions for each cohort. Consortium releases are restricted to GENIE BPC consortium members.

Examples

synapse_version()
synapse_version(most_recent = TRUE)
Index

* datasets
  - drug_regimen_list, 6
  - genie_panels, 9
  - nsclc_test_data, 9
  - regimen_abbreviations, 12
  - synapse_tables, 15

check_genie_access, 2
create_analytic_cohort, 3

drug_regimen_list, 6
drug_regimen_sunburst, 7

fetch_samples, 8

genie_panels, 9

nsclc_test_data, 9

pull_data_synapse, 10

regimen_abbreviations, 12

select_unique_ngs, 12
set_synapse_credentials, 14
synapse_tables, 15
synapse_version, 16