UCSCXenaTools: Download Public Cancer Genomic Data from UCSC Xena Hubs

Shixiang Wang

2019-12-14

UCSCXenaTools is an R package for downloading and exploring data from **UCSC Xena data hubs**, which are

> a collection of UCSC-hosted public databases such as TCGA, ICGC, TARGET, GTEx, CCLE, and others. Databases are normalized so they can be combined, linked, filtered, explored and downloaded.
>
> – UCSC Xena

If you use this package in academic field, please cite:

```
Wang, Shixiang, et al. "The predictive power of tumor mutational burden
    in lung cancer immunotherapy response is influenced by patients' sex."
    International journal of cancer (2019).
```

Installation

Install stable release from CRAN with:

```
install.packages("UCSCXenaTools")
```

You can also install devel version of **UCSCXenaTools** from github with:

```
# install.packages('remotes')
remotes::install_github("ShixiangWang/UCSCXenaTools")
```

Data Hub List

All datasets are available at https://xenabrowser.net/datapages/.
Currently, **UCSCXenaTools** supports 10 data hubs of UCSC Xena.

- UCSC Public Hub: https://ucscpublic.xenahubs.net
- TCGA Hub: https://tcga.xenahubs.net
- GDC Xena Hub: https://gdc.xenahubs.net
- ICGC Xena Hub: https://icgc.xenahubs.net
- Pan-Cancer Atlas Hub: https://pancanatlas.xenahubs.net
- GA4GH (TOIL) Hub: https://toil.xenahubs.net
- Treehouse Hub: https://xena.treehouse.gi.ucsc.edu
- PCAWG Hub: https://pcawg.xenahubs.net
- ATAC-seq Hub: https://atacseq.xenahubs.net

- Singel Cell Xena hub: https://singlecellnew.xenahubs.net

Users can update dataset list from the newest version of UCSC Xena by hand with `XenaDataUpdate()` function, followed by restarting R and `library(UCSCXenaTools)`.

If any url of data hubs are changed or a new data hub is online, please remind me by emailing to w_shixiang@163.com or opening an issue on GitHub.

Usage

Download UCSC Xena Datasets and load them into R by **UCSCXenaTools** is a workflow with `generate`, `filter`, `query`, `download` and `prepare` 5 steps, which are implemented as `XenaGenerate`, `XenaFilter`, `XenaQuery`, `XenaDownload` and `XenaPrepare` functions, respectively. They are very clear and easy to use and combine with other packages like `dplyr`.

To show the basic usage of **UCSCXenaTools**, we will download clinical data of LUNG, LUAD, LUSC from TCGA (hg19 version) data hub.

XenaData data.frame

Begin from version `0.2.0`, **UCSCXenaTools** uses a `data.frame` object (built in package, someone may call it `tibble`) `XenaData` to generate an instance of `XenaHub` class to record general information of all datasets of UCSC Xena Data Hubs.

You can load `XenaData` after loading `UCSCXenaTools` into R.

```
library(UCSCXenaTools)
#> ========================================================================================
#> UCSCXenaTools version 1.2.9
#> Project URL: https://github.com/ropensci/UCSCXenaTools
#> Usages: https://cran.r-project.org/web/packages/UCSCXenaTools/vignettes/USCSXenaTools.html
#>
#> If you use it in published research, please cite:
#> Wang et al., (2019). The UCSCXenaTools R package: a toolkit for accessing genomics data
#>   from UCSC Xena platform, from cancer multi-omics to single-cell RNA-seq.
#>   Journal of Open Source Software, 4(40), 1627, https://doi.org/10.21105/joss.01627
#> ========================================================================================
#>                                --Enjoy it--
data(XenaData)

head(XenaData)
#> # A tibble: 6 x 17
#>   XenaHosts XenaHostNames XenaCohorts XenaDatasets SampleCount DataSubtype Label
```

```
#>   <chr>     <chr>         <chr>       <chr>              <int> <chr>       <chr>
#> 1 https://... publicHub     Breast Can... ucsfNeve_pu...          51 gene expre... Neve...
#> 2 https://... publicHub     Breast Can... ucsfNeve_pu...          57 phenotype   Phen...
#> 3 https://... publicHub     Glioma (Ko... kotliarov20...         194 copy number Kotl...
#> 4 https://... publicHub     Glioma (Ko... kotliarov20...         194 phenotype   Phen...
#> 5 https://... publicHub     Lung Cance... weir2007_pu...         383 copy number CGH
#> 6 https://... publicHub     Lung Cance... weir2007_pu...         383 phenotype   Phen...
#> # ... with 10 more variables: Type <chr>, AnatomicalOrigin <chr>,
#> #   SampleType <chr>, Tags <chr>, ProbeMap <chr>, LongTitle <chr>,
#> #   Citation <chr>, Version <chr>, Unit <chr>, Platform <chr>
```

Names of all hub names/urls can be accessed by object `.xena_hosts`:

```
UCSCXenaTools:::.xena_hosts
#>    https://ucscpublic.xenahubs.net          https://tcga.xenahubs.net
#>                         "publicHub"                           "tcgaHub"
#>           https://gdc.xenahubs.net          https://icgc.xenahubs.net
#>                           "gdcHub"                           "icgcHub"
#>          https://toil.xenahubs.net   https://pancanatlas.xenahubs.net
#>                          "toilHub"                  "pancanAtlasHub"
#> https://xena.treehouse.gi.ucsc.edu         https://pcawg.xenahubs.net
#>                      "treehouseHub"                          "pcawgHub"
#>       https://atacseq.xenahubs.net https://singlecellnew.xenahubs.net
#>                        "atacseqHub"                    "singlecellHub"
```

Generate a XenaHub object

This can be implemented by `XenaGenerate` function, which generates `XenaHub` object from `XenaData` data frame.

```
XenaGenerate()
#> class: XenaHub
#> hosts():
#>   https://ucscpublic.xenahubs.net
#>   https://tcga.xenahubs.net
#>   https://gdc.xenahubs.net
#>   https://icgc.xenahubs.net
#>   https://toil.xenahubs.net
#>   https://pancanatlas.xenahubs.net
#>   https://xena.treehouse.gi.ucsc.edu
#>   https://pcawg.xenahubs.net
#>   https://atacseq.xenahubs.net
#>   https://singlecellnew.xenahubs.net
#> cohorts() (148 total):
```

```
#>   Breast Cancer Cell Lines (Neve 2006)
#>   Glioma (Kotliarov 2006)
#>   Lung Cancer CGH (Weir 2007)
#>   ...
#>   UCSC Cell Browser Multiple Sclerosis
#>   HCA Human Hematopoietic Profiling
#> datasets() (1738 total):
#>   ucsfNeve_public/ucsfNeveExp_genomicMatrix
#>   ucsfNeve_public/ucsfNeve_public_clinicalMatrix
#>   kotliarov2006_public/kotliarov2006_genomicMatrix
#>   ...
#>   HCA/Human_Hematopoietic_Profiling/cells.tsv
#>   HCA/Human_Hematopoietic_Profiling/expression.tsv
```

You can set `subset` argument to narrow datasets.

```
XenaGenerate(subset = XenaHostNames == "tcgaHub")
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
#> cohorts() (38 total):
#>   TCGA Ovarian Cancer (OV)
#>   TCGA Kidney Clear Cell Carcinoma (KIRC)
#>   TCGA Lower Grade Glioma (LGG)
#>   ...
#>   TCGA Colon Cancer (COAD)
#>   TCGA Formalin Fixed Paraffin-Embedded Pilot Phase II (FPPP)
#> datasets() (879 total):
#>   TCGA.OV.sampleMap/HumanMethylation27
#>   TCGA.OV.sampleMap/HumanMethylation450
#>   TCGA.OV.sampleMap/Gistic2_CopyNumber_Gistic2_all_data_by_genes
#>   ...
#>   TCGA.FPPP.sampleMap/miRNA_HiSeq_gene
#>   TCGA.FPPP.sampleMap/FPPP_clinicalMatrix
```

You can also use `XenaHub()` to generate a `XenaHub` object for API communication, but it is not recommended.

It's possible to extract info from `XenaHub` object by `hosts()`, `cohorts()` and `datasets()`.

```
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub")
# get hosts
hosts(xe)
#> [1] "https://tcga.xenahubs.net"
```

```
# get cohorts
head(cohorts(xe))
#> [1] "TCGA Ovarian Cancer (OV)"
#> [2] "TCGA Kidney Clear Cell Carcinoma (KIRC)"
#> [3] "TCGA Lower Grade Glioma (LGG)"
#> [4] "TCGA Kidney Papillary Cell Carcinoma (KIRP)"
#> [5] "TCGA Pan-Cancer (PANCAN)"
#> [6] "TCGA Bile Duct Cancer (CHOL)"
# get datasets
head(datasets(xe))
#> [1] "TCGA.OV.sampleMap/HumanMethylation27"
#> [2] "TCGA.OV.sampleMap/HumanMethylation450"
#> [3] "TCGA.OV.sampleMap/Gistic2_CopyNumber_Gistic2_all_data_by_genes"
#> [4] "TCGA.OV.sampleMap/mutation_broad"
#> [5] "TCGA.OV.sampleMap/OV_clinicalMatrix"
#> [6] "TCGA.OV.sampleMap/mutation_wustl_hiseq"
```

Pipe operator %>% can also be used here.

```
library(dplyr)
XenaData %>% filter(XenaHostNames == "tcgaHub", grepl("BRCA", XenaCohorts), grepl("Path",
    XenaDatasets)) %>% XenaGenerate()
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
#> cohorts() (1 total):
#>   TCGA Breast Cancer (BRCA)
#> datasets() (4 total):
#>   TCGA.BRCA.sampleMap/Pathway_Paradigm_mRNA_And_Copy_Number
#>   TCGA.BRCA.sampleMap/Pathway_Paradigm_RNASeq
#>   TCGA.BRCA.sampleMap/Pathway_Paradigm_RNASeq_And_Copy_Number
#>   TCGA.BRCA.sampleMap/Pathway_Paradigm_mRNA
```

Sometimes we only know some keywords, `XenaScan()` can be used to scan all rows to detect if the keywords exist in `XenaData`.

```
x1 = XenaScan(pattern = "Blood")
x2 = XenaScan(pattern = "LUNG", ignore.case = FALSE)

x1 %>% XenaGenerate()
#> class: XenaHub
#> hosts():
#>   https://ucscpublic.xenahubs.net
#>   https://tcga.xenahubs.net
```

```
#> cohorts() (6 total):
#>   Connectivity Map
#>   TARGET Acute Lymphoblastic Leukemia
#>   Pediatric tumor (Khan)
#>   Acute lymphoblastic leukemia (Mullighan 2008)
#>   TCGA Pan-Cancer (PANCAN)
#>   TCGA Acute Myeloid Leukemia (LAML)
#> datasets() (34 total):
#>   cmap/rankMatrix_reverse
#>   TARGET_ALL/TARGETcnv_genomicMatrix
#>   TARGET_ALL/TARGETexp_genomicMatrix
#>   ...
#>   TCGA.LAML.sampleMap/mutation_wustl
#>   TCGA.LAML.sampleMap/Pathway_Paradigm_RNASeq_And_Copy_Number
x2 %>% XenaGenerate()
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
#> cohorts() (1 total):
#>   TCGA Lung Cancer (LUNG)
#> datasets() (13 total):
#>   TCGA.LUNG.sampleMap/HumanMethylation27
#>   TCGA.LUNG.sampleMap/HumanMethylation450
#>   TCGA.LUNG.sampleMap/Gistic2_CopyNumber_Gistic2_all_data_by_genes
#>   ...
#>   TCGA.LUNG.sampleMap/HiSeqV2_exon
#>   TCGA.LUNG.sampleMap/AgilentG4502A_07_3
```

Filter

There are too many datasets in `xe`, you can filter them by `XenaFilter` function. Regular expression can be used here.

```
(xe2 <- XenaFilter(xe, filterDatasets = "clinical"))
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
#> cohorts() (37 total):
#>   TCGA Ovarian Cancer (OV)
#>   TCGA Kidney Clear Cell Carcinoma (KIRC)
#>   TCGA Lower Grade Glioma (LGG)
#>   ...
#>   TCGA Colon Cancer (COAD)
```

```
#>   TCGA Formalin Fixed Paraffin-Embedded Pilot Phase II (FPPP)
#> datasets() (37 total):
#>   TCGA.OV.sampleMap/OV_clinicalMatrix
#>   TCGA.KIRC.sampleMap/KIRC_clinicalMatrix
#>   TCGA.LGG.sampleMap/LGG_clinicalMatrix
#>   ...
#>   TCGA.COAD.sampleMap/COAD_clinicalMatrix
#>   TCGA.FPPP.sampleMap/FPPP_clinicalMatrix
```

Then select `LUAD`, `LUSC` and `LUNG` 3 datasets.

```
xe2 <- XenaFilter(xe2, filterDatasets = "LUAD|LUSC|LUNG")
```

Pipe can be used here.

```
xe %>% XenaFilter(filterDatasets = "clinical") %>% XenaFilter(filterDatasets = "luad|lusc|lung")
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
#> cohorts() (3 total):
#>   TCGA Lung Cancer (LUNG)
#>   TCGA Lung Adenocarcinoma (LUAD)
#>   TCGA Lung Squamous Cell Carcinoma (LUSC)
#> datasets() (3 total):
#>   TCGA.LUNG.sampleMap/LUNG_clinicalMatrix
#>   TCGA.LUAD.sampleMap/LUAD_clinicalMatrix
#>   TCGA.LUSC.sampleMap/LUSC_clinicalMatrix
```

Browse datasets

Sometimes, you may want to check data before you query and download data.
A new feature `XenaBrowse` is implemented in **UCSCXenaTools**.

Create two XenaHub objects:

- `to_browse` - a XenaHub object contains a cohort and a dataset.
- `to_browse2` - a XenaHub object contains 2 cohorts and 2 datasets.

```
to_browse <- XenaGenerate(subset = XenaHostNames == "tcgaHub") %>% XenaFilter(filterDatasets = "clinica
    XenaFilter(filterDatasets = "LUAD")

to_browse
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
```

```
#> cohorts() (1 total):
#>   TCGA Lung Adenocarcinoma (LUAD)
#> datasets() (1 total):
#>   TCGA.LUAD.sampleMap/LUAD_clinicalMatrix

to_browse2 <- XenaGenerate(subset = XenaHostNames == "tcgaHub") %>% XenaFilter(filterDatasets = "clinica
    XenaFilter(filterDatasets = "LUAD|LUSC")

to_browse2
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
#> cohorts() (2 total):
#>   TCGA Lung Adenocarcinoma (LUAD)
#>   TCGA Lung Squamous Cell Carcinoma (LUSC)
#> datasets() (2 total):
#>   TCGA.LUAD.sampleMap/LUAD_clinicalMatrix
#>   TCGA.LUSC.sampleMap/LUSC_clinicalMatrix
```

`XenaBrowse()` function can be used to browse dataset/cohort links using your default web browser. At default, this function limit one dataset/cohort for preventing user to open too many links at once.

```
# This will open you web browser
XenaBrowse(to_browse)

XenaBrowse(to_browse, type = "cohort")
```

```
# This will throw error
XenaBrowse(to_browse2)
#> Error in XenaBrowse(to_browse2): This function limite 1 dataset to browse.
#>  Set multiple to TRUE if you want to browse multiple links.

XenaBrowse(to_browse2, type = "cohort")
#> Error in XenaBrowse(to_browse2, type = "cohort"): This function limite 1 cohort to browse.
#>  Set multiple to TRUE if you want to browse multiple links.
```

When you make sure you want to open multiple links, you can set `multiple` option to `TRUE`.

```
XenaBrowse(to_browse2, multiple = TRUE)
XenaBrowse(to_browse2, type = "cohort", multiple = TRUE)
```

Query

Create a query before downloading data.

```
xe2_query = XenaQuery(xe2)
#> This will check url status, please be patient.
xe2_query
#>                        hosts                                    datasets
#> 1 https://tcga.xenahubs.net TCGA.LUNG.sampleMap/LUNG_clinicalMatrix
#> 2 https://tcga.xenahubs.net TCGA.LUAD.sampleMap/LUAD_clinicalMatrix
#> 3 https://tcga.xenahubs.net TCGA.LUSC.sampleMap/LUSC_clinicalMatrix
#>                                                                                  url
#> 1 https://tcga.xenahubs.net/download/TCGA.LUNG.sampleMap/LUNG_clinicalMatrix
#> 2 https://tcga.xenahubs.net/download/TCGA.LUAD.sampleMap/LUAD_clinicalMatrix
#> 3 https://tcga.xenahubs.net/download/TCGA.LUSC.sampleMap/LUSC_clinicalMatrix
```

Download

Default, data will be downloaded to system temp directory. You can specify the path.

If the data exists, command will not run to download them, but you can force it by `force` option.

```
destdir = file.path(tempdir(), "test")
xe2_download = XenaDownload(xe2_query, destdir = destdir, trans_slash = TRUE)
#> All downloaded files will under directory D:/Tool/Rtmp\RtmpEB6Kgv/test.
#> Downloading TCGA.LUNG.sampleMap__LUNG_clinicalMatrix
#> Downloading TCGA.LUAD.sampleMap__LUAD_clinicalMatrix
#> Downloading TCGA.LUSC.sampleMap__LUSC_clinicalMatrix
#> Note file names inherit from names in datasets column
#>   and '/' all changed to '__'.
```

Of note, at default, the downloaded files will keep same directory structure as Xena. You can set `trans_slash` to TRUE, it will transform `/` in dataset id to `__`, this will make all downloaded files are under same directory.

Prepare

There are 4 ways to prepare data to R.

```
# way1: directory
cli1 = XenaPrepare(destdir)
names(cli1)
#> [1] "TCGA.LUAD.sampleMap__LUAD_clinicalMatrix"
```

```
#> [2] "TCGA.LUNG.sampleMap__LUNG_clinicalMatrix"
#> [3] "TCGA.LUSC.sampleMap__LUSC_clinicalMatrix"
```

```
# way2: local files
cli2 = XenaPrepare(file.path(destdir, "/TCGA.LUAD.sampleMap__LUAD_clinicalMatrix"))
class(cli2)
#> [1] "spec_tbl_df" "tbl_df"      "tbl"         "data.frame"
```

```
# way3: urls
cli3 = XenaPrepare(xe2_download$url[1:2])
names(cli3)
## [1] "LUSC_clinicalMatrix" "LUNG_clinicalMatrix"
```

```
# way4: xenadownload object
cli4 = XenaPrepare(xe2_download)
names(cli4)
#> [1] "TCGA.LUNG.sampleMap__LUNG_clinicalMatrix"
#> [2] "TCGA.LUAD.sampleMap__LUAD_clinicalMatrix"
#> [3] "TCGA.LUSC.sampleMap__LUSC_clinicalMatrix"
```

From v0.2.6, `XenaPrepare()` can enable chunk feature when file is too big and user only need subset of file.

Following code show how to subset some rows or columns of files, `sample` is the name of the first column, user can directly use it in logical expression, `x` can be a representation of data frame user wanna do subset operation. More custom operation can be set as a function and pass to `callback` option.

```
# select rows which sample (gene symbol here) in "HIF3A" or "RNF17"
testRNA = UCSCXenaTools::XenaPrepare("~/Download/HiSeqV2.gz", use_chunk = TRUE, subset_rows = sample %in
# only keep 1 to 3 columns
testRNA = UCSCXenaTools::XenaPrepare("~/Download/HiSeqV2.gz", use_chunk = TRUE, select_cols = colnames(
```

Download TCGA data with readable options

getTCGAdata

`getTCGAdata` provides a more readable way for downloading TCGA (hg19 version, different from `gdcHub`) datasets, user can specify multiple options to select data and corresponding file type to download. Default this function will return a list include `XenaHub` object and selected datasets information. Once you are sure the datasets are exactly what you want, `download` can be set to `TRUE` to download the data.

Check arguments of `getTCGAdata`:

```
args(getTCGAdata)
#> function (project = NULL, clinical = TRUE, download = FALSE,
#>     forceDownload = FALSE, destdir = tempdir(), mRNASeq = FALSE,
#>     mRNAArray = FALSE, mRNASeqType = "normalized", miRNASeq = FALSE,
#>     exonRNASeq = FALSE, RPPAArray = FALSE, ReplicateBaseNormalization = FALSE,
#>     Methylation = FALSE, MethylationType = c("27K", "450K"),
#>     GeneMutation = FALSE, SomaticMutation = FALSE, GisticCopyNumber = FALSE,
#>     Gistic2Threshold = TRUE, CopyNumberSegment = FALSE, RemoveGermlineCNV = TRUE,
#>     ...)
#> NULL

# or run ??getTCGAdata to read documentation
```

Select one or more projects, default will select only clinical datasets:

```
getTCGAdata(c("UVM", "LUAD"))
#> $Xena
#> class: XenaHub
#> hosts():
#>   https://tcga.xenahubs.net
#> cohorts() (2 total):
#>   TCGA Lung Adenocarcinoma (LUAD)
#>   TCGA Ocular melanomas (UVM)
#> datasets() (2 total):
#>   TCGA.LUAD.sampleMap/LUAD_clinicalMatrix
#>   TCGA.UVM.sampleMap/UVM_clinicalMatrix
#>
#> $DataInfo
#> # A tibble: 2 x 20
#>   XenaHosts XenaHostNames XenaCohorts XenaDatasets SampleCount DataSubtype Label
#>   <chr>     <chr>         <chr>       <chr>              <int> <chr>       <chr>
#> 1 https://... tcgaHub       TCGA Lung ... TCGA.LUAD.s...         706 phenotype   Phen...
#> 2 https://... tcgaHub       TCGA Ocula... TCGA.UVM.sa...          80 phenotype   Phen...
#> # ... with 13 more variables: Type <chr>, AnatomicalOrigin <chr>,
#> #   SampleType <chr>, Tags <chr>, ProbeMap <chr>, LongTitle <chr>,
#> #   Citation <chr>, Version <chr>, Unit <chr>, Platform <chr>, ProjectID <chr>,
#> #   DataType <chr>, FileType <chr>

tcga_data = getTCGAdata(c("UVM", "LUAD"))

# only return XenaHub object
tcga_data$Xena
#> class: XenaHub
#> hosts():
```

```
#>   https://tcga.xenahubs.net
#> cohorts() (2 total):
#>   TCGA Lung Adenocarcinoma (LUAD)
#>   TCGA Ocular melanomas (UVM)
#> datasets() (2 total):
#>   TCGA.LUAD.sampleMap/LUAD_clinicalMatrix
#>   TCGA.UVM.sampleMap/UVM_clinicalMatrix

# only return datasets information
tcga_data$DataInfo
#> # A tibble: 2 x 20
#>   XenaHosts XenaHostNames XenaCohorts XenaDatasets SampleCount DataSubtype Label
#>   <chr>     <chr>         <chr>       <chr>              <int> <chr>       <chr>
#> 1 https://... tcgaHub       TCGA Lung ... TCGA.LUAD.s...         706 phenotype   Phen...
#> 2 https://... tcgaHub       TCGA Ocula... TCGA.UVM.sa...          80 phenotype   Phen...
#> # ... with 13 more variables: Type <chr>, AnatomicalOrigin <chr>,
#> #   SampleType <chr>, Tags <chr>, ProbeMap <chr>, LongTitle <chr>,
#> #   Citation <chr>, Version <chr>, Unit <chr>, Platform <chr>, ProjectID <chr>,
#> #   DataType <chr>, FileType <chr>
```

Set `download=TRUE` to download data, default data will be downloaded to system temp directory (you can specify the path with `destdir` option):

```
# only download clinical data
getTCGAdata(c("UVM", "LUAD"), download = TRUE)
```

Support Data Type and Options:

- clinical information: `clinical`
- mRNA Sequencing: `mRNASeq`
- mRNA microarray: `mRNAArray`
- miRNA Sequencing: `miRNASeq`
- exon Sequencing: `exonRNASeq`
- RPPA array: `RPPAArray`
- DNA Methylation: `Methylation`
- Gene mutation: `GeneMutation`
- Somatic mutation: `SomaticMutation`
- Gistic2 Copy Number: `GisticCopyNumber`
- Copy Number Segment: `CopyNumberSegment`

other data type supported by Xena cannot download use this function. Please refer to `downloadTCGA` function or `XenaGenerate` function.

NOTE: Sequencing data are all based on Illumina Hiseq platform, other platform (Illumina GA) data supported by Xena cannot download using this function. This is for building consistent data download flow. Mutation use

`broad automated` version (except `PANCAN` use `MC3 Public Version`). If you wan to download other datasets, please refer to `downloadTCGA` function or `XenaGenerate` function.

Download any TCGA data by datatypes and filetypes

`downloadTCGA` function can be used to download any TCGA data supported by Xena, but in a way different from `getTCGAdata` function.

```
# download RNASeq data (use UVM as an example)
downloadTCGA(project = "UVM", data_type = "Gene Expression RNASeq", file_type = "IlluminaHiSeq RNASeqV2
```

See the arguments:

```
args(downloadTCGA)
#> function (project = NULL, data_type = NULL, file_type = NULL,
#>     destdir = tempdir(), force = FALSE, ...)
#> NULL
```

Except `destdir` option, you only need to select three arguments for downloading data. Even throught the number is far less than `getTCGAdata`, it is more complex than the latter.

Before you download data, you need spare some time to figure out what data type and file type available and what your datasets have.

`availTCGA` can return all information you need:

```
availTCGA()
#> Note not all projects have listed data types and file types, you can use showTCGA function to check
#> $ProjectID
#>  [1] "OV"       "KIRC"     "LGG"      "KIRP"     "PANCAN"   "CHOL"
#>  [7] "COADREAD" "ACC"      "CESC"     "READ"     "SARC"     "DLBC"
#> [13] "PRAD"     "LUNG"     "LIHC"     "KICH"     "HNSC"     "PCPG"
#> [19] "ESCA"     "THCA"     "LUAD"     "LAML"     "BLCA"     "SKCM"
#> [25] "LUSC"     "TGCT"     "PAAD"     "GBM"      "STAD"     "MESO"
#> [31] "UVM"      "GBMLGG"   "THYM"     "UCEC"     "BRCA"     "UCS"
#> [37] "COAD"     "FPPP"
#>
#> $DataType
#>  [1] "DNA Methylation"
#>  [2] "Gene Level Copy Number"
#>  [3] "Somatic Mutation"
#>  [4] "Phenotype"
#>  [5] "Protein Expression RPPA"
#>  [6] "Gene Expression Array"
#>  [7] "Gene Expression RNASeq"
```

```
#>  [8] "Gene Somatic Non-silent Mutation"
#>  [9] "Copy Number Segments"
#> [10] "miRNA Mature Strand Expression RNASeq"
#> [11] "PARADIGM Pathway Activity"
#> [12] "Exon Expression RNASeq"
#> [13] "Transcription Factor Regulatory Impact"
#> [14] "Signatures"
#> [15] "iCluster"
#>
#> $FileType
#>  [1] "Methylation27K"
#>  [2] "Methylation450K"
#>  [3] "Gistic2"
#>  [4] "broad automated"
#>  [5] "Clinical Information"
#>  [6] "wustl hiseq automated"
#>  [7] "RPPA normalized by RBN"
#>  [8] "Affymetrix U133A Microarray"
#>  [9] "bcm SOLiD"
#> [10] "IlluminaHiSeq RNASeqV2 in percentile rank"
#> [11] "IlluminaHiSeq RNASeqV2 pancan normalized"
#> [12] "IlluminaHiSeq RNASeqV2"
#> [13] "IlluminaHiSeq RNASeq"
#> [14] "After remove germline cnv"
#> [15] "Agilent 244K Microarray"
#> [16] "PANCAN AWG analyzed"
#> [17] "bcm SOLiD curated"
#> [18] "wustl automated"
#> [19] "Use Microarray plus Copy Number"
#> [20] "Gistic2 thresholded"
#> [21] "RPPA"
#> [22] "Before remove germline cnv"
#> [23] "Gene Expression Subtype"
#> [24] "Use only RNASeq"
#> [25] "Use RNASeq plus Copy Number"
#> [26] "Use only Microarray"
#> [27] "RABIT Use Agilent 244K Microarray"
#> [28] "RABIT Use Affymetrix U133A Microarray"
#> [29] "bcm automated"
#> [30] "IlluminaGA RNASeq"
#> [31] "RABIT Use IlluminaHiSeq RNASeqV2"
#> [32] "RABIT Use IlluminaHiSeq RNASeq"
#> [33] "MethylMix"
```

```
#> [34] "ucsc automated"
#> [35] "broad curated"
#> [36] "bcm curated"
#> [37] "Platform-corrected PANCAN12 dataset"
#> [38] "bsgsc automated"
#> [39] "bcgsc automated"
#> [40] "wustl curated"
#> [41] "IlluminaGA RNASeqV2"
#> [42] "RABIT Use IlluminaGA RNASeqV2"
#> [43] "RABIT Use IlluminaGA RNASeq"
#> [44] "Batch effects normalized"
#> [45] "MC3 Public Version"
#> [46] "TCGA Sample Type and Primary Disease"
#> [47] "RPPA pancan normalized"
#> [48] "Tumor copy number"
#> [49] "Genome-wide DNA Damage Footprint HRD Score"
#> [50] "TCGA Molecular Subtype"
#> [51] "iCluster cluster assignments"
#> [52] "iCluster latent variables"
#> [53] "RNA based StemnessScore"
#> [54] "DNA methylation based StemnessScore"
#> [55] "Pancan Gene Programs"
#> [56] "Immune Model Based Subtype"
#> [57] "Immune Signature Scores"
```

Note not all datasets have these property, `showTCGA` can help you to check it. It will return all data in TCGA, you can use following code in RStudio and search your data.

```
View(showTCGA())
```

OR you can use shiny app provided by `UCSCXenaTools` to search.
Run shiny by:

```
UCSCXenaTools::XenaShiny()
```

SessionInfo

```
sessionInfo()
#> R version 3.6.1 (2019-07-05)
#> Platform: x86_64-w64-mingw32/x64 (64-bit)
#> Running under: Windows 10 x64 (build 18362)
#>
```

```
#> Matrix products: default
#>
#> locale:
#> [1] LC_COLLATE=Chinese (Simplified)_China.936
#> [2] LC_CTYPE=Chinese (Simplified)_China.936
#> [3] LC_MONETARY=Chinese (Simplified)_China.936
#> [4] LC_NUMERIC=C
#> [5] LC_TIME=Chinese (Simplified)_China.936
#>
#> attached base packages:
#> [1] stats     graphics  grDevices utils     datasets  methods   base
#>
#> other attached packages:
#> [1] dplyr_0.8.3         UCSCXenaTools_1.2.9 pacman_0.5.1
#>
#> loaded via a namespace (and not attached):
#>  [1] Rcpp_1.0.3       knitr_1.26       magrittr_1.5     hms_0.5.2
#>  [5] tidyselect_0.2.5 R6_2.4.1         rlang_0.4.1      fansi_0.4.0
#>  [9] stringr_1.4.0    httr_1.4.1       tools_3.6.1      tint_0.1.2
#> [13] xfun_0.11        utf8_1.1.4       cli_2.0.0        htmltools_0.4.0
#> [17] yaml_2.2.0       digest_0.6.22    assertthat_0.2.1 tibble_2.1.3
#> [21] crayon_1.3.4     purrr_0.3.3      readr_1.3.1      formatR_1.7
#> [25] vctrs_0.2.0      curl_4.2         zeallot_0.1.0    glue_1.3.1
#> [29] evaluate_0.14    rmarkdown_1.18   stringi_1.4.3    compiler_3.6.1
#> [33] pillar_1.4.2     backports_1.1.5  pkgconfig_2.0.3
```

Bug Report

I have no time to test if all conditions are right and all datasets can normally be downloaded. So if you have any question or suggestion, please open an issue on Github at https://github.com/ShixiangWang/UCSCXenaTools/issues.

Acknowledgement

This package is based on XenaR, thanks Martin Morgan for his work.

LICENSE