Package ‘UCSCXenaTools’

January 20, 2025

Title Download and Explore Datasets from UCSC Xena Data Hubs

Version 1.4.8

Maintainer Shixiang Wang <`w_shixiang@163.com`>

Description Download and explore datasets 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.

License GPL-3

URL `https://docs.ropensci.org/UCSCXenaTools/`,
`https://github.com/ropensci/UCSCXenaTools`

BugReports `https://github.com/ropensci/UCSCXenaTools/issues`

Depends R (>= 3.5)

Imports digest, dplyr, httr, jsonlite, magrittr, methods, readr, rlang, utils

Suggests covr, DT, knitr, prettydoc, rmarkdown, shiny, shinydashboard, testthat (>= 2.1.0)

VignetteBuilder knitr

Encoding UTF-8

LazyData true

RoxygenNote 7.2.0

NeedsCompilation no

Author Shixiang Wang [aut, cre] (<`https://orcid.org/0000-0001-9855-7357`>),
Xue-Song Liu [aut] (<`https://orcid.org/0000-0002-7736-0077`>),
Martin Morgan [ctb],
Christine Stawitz [rev] (Christine reviewed the package for ropensci,
see <`https://github.com/ropensci/software-review/issues/315`>),
Carl Ganz [rev] (Carl reviewed the package for ropensci, see
<`https://github.com/ropensci/software-review/issues/315`>)

Repository CRAN

Date/Publication 2022-06-20 08:10:02 UTC

Contents

availTCGA 2
cohorts 3
datasets 3
downloadTCGA 4
fetch 5
getTCGAdata 7
hosts 9
samples 10
showTCGA 11
to_snake 12
XenaBrowse 12
XenaData 13
XenaDataUpdate 13
XenaDownload 14
XenaFilter 15
XenaGenerate 16
XenaHub 17
XenaHub-class 18
XenaPrepare 18
XenaQuery 20
XenaQueryProbeMap 20
XenaScan 21
XenaShiny 22
xena_default_hosts 22

Index **23**

`availTCGA` *Get or Check TCGA Available ProjectID, DataType and FileType*

Description

Get or Check TCGA Available ProjectID, DataType and FileType

Usage

```
availTCGA(which = c("all", "ProjectID", "DataType", "FileType"))
```

Arguments

`which`	a character of `c("All", "ProjectID", "DataType", "FileType")`

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
availTCGA("all")
```

`cohorts` *Get cohorts of XenaHub object*

Description

Get cohorts of XenaHub object

Usage

```
cohorts(x)
```

Arguments

`x`	a XenaHub object

Value

a character vector contains cohorts

Examples

```
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub"); cohorts(xe)
```

`datasets` *Get datasets of XenaHub object*

Description

Get datasets of XenaHub object

Usage

```
datasets(x)
```

Arguments

`x`	a XenaHub object

Value

a character vector contains datasets

Examples

```
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub"); datasets(xe)
```

`downloadTCGA` *Easily Download TCGA Data by Several Options*

Description

TCGA is a very useful database and here we provide this function to download TCGA (include TCGA Pancan) datasets in human-friendly way. Users who are not familiar with R operation will benefit from this.

Usage

```
downloadTCGA(
  project = NULL,
  data_type = NULL,
  file_type = NULL,
  destdir = tempdir(),
  force = FALSE,
  ...
)
```

Arguments

`project`	default is `NULL`. Should be one or more of TCGA project id (character vector) provided by Xena. See all available project id, please use `availTCGA("ProjectID")`.
`data_type`	default is `NULL`. Should be a character vector specify data type. See all available data types by `availTCGA("DataType")`.
`file_type`	default is `NULL`. Should be a character vector specify file type. See all available file types by `availTCGA("FileType")`.
`destdir`	specify a location to store download data. Default is system temp directory.
`force`	logical. if `TRUE`, force to download data no matter whether files exist. Default is `FALSE`.
`...`	other argument to `download.file` function

Details

All availble information about datasets of TCGA can access vis `availTCGA()` and check with `showTCGA()`.

Value

same as `XenaDownload()` function result.

Author(s)

Shixiang Wang w_shixiang@163.com

See Also

`XenaQuery()`, `XenaFilter()`, `XenaDownload()`, `XenaPrepare()`, `availTCGA()`, `showTCGA()`

Examples

```
## Not run:
# download RNASeq data (use UVM as example)
downloadTCGA(project = "UVM",
                 data_type = "Gene Expression RNASeq",
                 file_type = "IlluminaHiSeq RNASeqV2")

## End(Not run)
```

`fetch` *Fetch Data from UCSC Xena Hosts*

Description

When you want to query just data for several genes/samples from UCSC Xena datasets, a better way is to use these `fetch_` functions instead of downloading a whole dataset. Details about functions please see the following sections.

Usage

```
fetch(host, dataset)

fetch_dense_values(
  host,
  dataset,
  identifiers = NULL,
  samples = NULL,
  check = TRUE,
  use_probeMap = FALSE,
  time_limit = 30
)

fetch_sparse_values(host, dataset, genes, samples = NULL, time_limit = 30)

fetch_dataset_samples(host, dataset, limit = NULL)

fetch_dataset_identifiers(host, dataset)

has_probeMap(host, dataset, return_url = FALSE)
```

Arguments

`host`	a UCSC Xena host, like "https://toil.xenahubs.net". All available hosts can be printed by `xena_default_hosts()`.
`dataset`	a UCSC Xena dataset, like "tcga_RSEM_gene_tpm". All available datasets can be printed by running `XenaData$XenaDatasets` or obtained from UCSC Xena datapages.
`identifiers`	Identifiers could be probe (like "ENSG00000000419.12"), gene (like "TP53") etc.. If it is `NULL`, all identifiers in the dataset will be used.
`samples`	ID of samples, like "TCGA-02-0047-01". If it is `NULL`, all samples in the dataset will be used. However, it is better to download the whole datasets if you query many samples and genes.
`check`	if `TRUE`, check whether specified `identifiers` and `samples` exist the dataset (all failed items will be filtered out). However, if `FALSE`, the code is much faster.
`use_probeMap`	if `TRUE`, will check if the dataset has ProbeMap firstly. When the dataset you want to query has a identifier-to-gene mapping, identifiers can be gene symbols even the identifiers of dataset are probes or others.
`time_limit`	time limit for getting response in seconds.
`genes`	gene names.
`limit`	number of samples, if `NULL`, return all samples.
`return_url`	if `TRUE`, returns the info of probeMap instead of a logical value when the result exists.

Details

There are three primary data types: dense matrix (samples by probes (or say identifiers)), sparse (sample, position, variant), and segmented (sample, position, value).

Dense matrices can be genotypic or phenotypic, it is a sample-by-identifiers matrix. Phenotypic matrices have associated field metadata (descriptive names, codes, etc.). Genotypic matricies may have an associated probeMap, which maps probes to genomic locations. If a matrix has hugo probeMap, the probes themselves are gene names. Otherwise, a probeMap is used to map a gene location to a set of probes.

Value

a `matirx` or character vector or a `list`.

Functions

- `fetch_dense_values`: fetches values from a dense matrix.
- `fetch_sparse_values`: fetches values from a sparse `data.frame`.
- `fetch_dataset_samples`: fetches samples from a dataset
- `fetch_dataset_identifiers`: fetches identifies from a dataset.
- `has_probeMap`: checks if a dataset has ProbeMap.

Examples

```
library(UCSCXenaTools)

host <- "https://toil.xenahubs.net"
dataset <- "tcga_RSEM_gene_tpm"
samples <- c("TCGA-02-0047-01", "TCGA-02-0055-01", "TCGA-02-2483-01", "TCGA-02-2485-01")
probes <- c("ENSG00000282740.1", "ENSG00000000005.5", "ENSG00000000419.12")
genes <- c("TP53", "RB1", "PIK3CA")


# Fetch samples
fetch_dataset_samples(host, dataset, 2)
# Fetch identifiers
fetch_dataset_identifiers(host, dataset)
# Fetch expression value by probes
fetch_dense_values(host, dataset, probes, samples, check = FALSE)
# Fetch expression value by gene symbol (if the dataset has probeMap)
has_probeMap(host, dataset)
fetch_dense_values(host, dataset, genes, samples, check = FALSE, use_probeMap = TRUE)
```

`getTCGAdata` *Get TCGA Common Data Sets by Project ID and Property*

Description

This is the most useful function for user to download common TCGA datasets, it is similar to `getFirehoseData` function in `RTCGAToolbox` package.

Usage

```
getTCGAdata(
  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,
  ...
)
```

Arguments

`project`	default is `NULL`. Should be one or more of TCGA project id (character vector) provided by Xena. See all available project id, please use `availTCGA("ProjectID")`.
`clinical`	logical. if `TRUE`, download clinical information. Default is `TRUE`.
`download`	logical. if `TRUE`, download data, otherwise return a result list include data information. Default is `FALSE`. You can set this to `FALSE` if you want to check what you will download or use other function provided by `UCSCXenaTools` to filter result datasets you want to download.
`forceDownload`	logical. if `TRUE`, force to download files no matter if exist. Default is `FALSE`.
`destdir`	specify a location to store download data. Default is system temp directory.
`mRNASeq`	logical. if `TRUE`, download mRNASeq data. Default is `FALSE`.
`mRNAArray`	logical. if `TRUE`, download mRNA microarray data. Default is `FALSE`.
`mRNASeqType`	character vector. Can be one, two or three in `c("normalized", "pancan normalized", "percentile")`.
`miRNASeq`	logical. if `TRUE`, download miRNASeq data. Default is `FALSE`.
`exonRNASeq`	logical. if `TRUE`, download exon RNASeq data. Default is `FALSE`.
`RPPAArray`	logical. if `TRUE`, download RPPA data. Default is `FALSE`.
`ReplicateBaseNormalization`	logical. if `TRUE`, download RPPA data by Replicate Base Normalization (RBN). Default is `FALSE`.
`Methylation`	logical. if `TRUE`, download DNA Methylation data. Default is `FALSE`.
`MethylationType`	character vector. Can be one or two in `c("27K", "450K")`.
`GeneMutation`	logical. if `TRUE`, download gene mutation data. Default is `FALSE`.
`SomaticMutation`	logical. if `TRUE`, download somatic mutation data. Default is `FALSE`.
`GisticCopyNumber`	logical. if `TRUE`, download Gistic2 Copy Number data. Default is `FALSE`.
`Gistic2Threshold`	logical. if `TRUE`, download Threshold Gistic2 data. Default is `TRUE`.
`CopyNumberSegment`	logical. if `TRUE`, download Copy Number Segment data. Default is `FALSE`.
`RemoveGermlineCNV`	logical. if `TRUE`, download Copy Number Segment data which has removed germline copy number variation. Default is `TRUE`.
`...`	other argument to `download.file` function

Details

TCGA Common Data Sets are frequently used for biological analysis. To make easier to achieve these data, this function provide really easy options to choose datasets and behavior. All availble information about datasets of TCGA can access vis `availTCGA()` and check with `showTCGA()`.

Value

if `download=TRUE`, return `data.frame` from `XenaDownload`, otherwise return a list including `XenaHub` object and datasets information

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
###### get data, but not download

# 1 choose project and data types you wanna download
getTCGAdata(project = "LUAD", mRNASeq = TRUE, mRNAArray = TRUE,
mRNASeqType = "normalized", miRNASeq = TRUE, exonRNASeq = TRUE,
RPPAArray = TRUE, Methylation = TRUE, MethylationType = "450K",
GeneMutation = TRUE, SomaticMutation = TRUE)

# 2 only choose 'LUAD' and its clinical data
getTCGAdata(project = "LUAD")
## Not run:
###### download datasets

# 3 download clinical datasets of LUAD and LUSC
getTCGAdata(project = c("LUAD", "LUSC"), clinical = TRUE, download = TRUE)

# 4 download clinical, RPPA and gene mutation datasets of LUAD and LUSC
# getTCGAdata(project = c("LUAD", "LUSC"), clinical = TRUE, RPPAArray = TRUE, GeneMutation = TRUE)

## End(Not run)
```

`hosts` *Get hosts of XenaHub object*

Description

Get hosts of XenaHub object

Usage

```
hosts(x)
```

Arguments

`x`	a XenaHub object

Value

a character vector contains hosts

Examples

```
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub"); hosts(xe)
```

`samples` *Get Samples of a XenaHub object according to 'by' and 'how' action arguments*

Description

One is often interested in identifying samples or features present in each data set, or shared by all data sets, or present in any of several data sets. Identifying these samples, including samples in arbitrarily chosen data sets.

Usage

```
samples(
  x,
  i = character(),
  by = c("hosts", "cohorts", "datasets"),
  how = c("each", "any", "all")
)
```

Arguments

`x`	a XenaHub object
`i`	default is a empty character, it is used to specify the host, cohort or dataset by by option otherwise info will be automatically extracted by code
`by`	a character specify by action
`how`	a character specify how action

Value

a list include samples

Examples

```
## Not run:
xe = XenaHub(cohorts = "Cancer Cell Line Encyclopedia (CCLE)")
# samples in each dataset, first host
x = samples(xe, by="datasets", how="each")[[1]]
lengths(x)        # data sets in ccle cohort on first (only) host

## End(Not run)
```

`showTCGA` *Show TCGA data structure by Project ID or ALL*

Description

This can used to check if data type or file type exist in one or more projects by hand.

Usage

```
showTCGA(project = "all")
```

Arguments

`project`	a character vector. Can be "all" or one or more of TCGA Project IDs.

Value

a `data.frame` including project data structure information.

Author(s)

Shixiang Wang w_shixiang@163.com

See Also

`availTCGA()`

Examples

```
showTCGA("all")
```

`to_snake` *Convert camel case to snake case*

Description

Convert camel case to snake case

Usage

```
to_snake(name)
```

Arguments

`name`	a character vector

Value

same length as `name` but with snake case

Examples

```
to_snake("sparseDataRange")
```

`XenaBrowse` *View Info of Dataset or Cohort at UCSC Xena Website Using Web browser*

Description

This will open dataset/cohort link of UCSC Xena in user's default browser.

Usage

```
XenaBrowse(x, type = c("dataset", "cohort"), multiple = FALSE)
```

Arguments

`x`	a XenaHub object.
`type`	one of "dataset" and "cohort".
`multiple`	if `TRUE`, browse multiple links instead of throwing error.

Examples

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

`XenaData` *Xena Hub Information*

Description

This `data.frame` is very useful for selecting datasets fastly and independent on APIs of UCSC Xena Hubs.

Format

A `tibble`.

Source

Generated from UCSC Xena Data Hubs.

Examples

```
data(XenaData)
str(XenaData)
```

`XenaDataUpdate` *Get or Update Newest Data Information of UCSC Xena Data Hubs*

Description

Get or Update Newest Data Information of UCSC Xena Data Hubs

Usage

```
XenaDataUpdate(saveTolocal = TRUE)
```

Arguments

`saveTolocal`	logical. Whether save to local R package data directory for permanent use or Not.

Value

a `data.frame` contains all datasets information of Xena.

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
## Not run:
XenaDataUpdate()
XenaDataUpdate(saveTolocal = TRUE)

## End(Not run)
```

`XenaDownload` *Download Datasets from UCSC Xena Hubs*

Description

Avaliable datasets list: `https://xenabrowser.net/datapages/`

Usage

```
XenaDownload(
  xquery,
  destdir = tempdir(),
  download_probeMap = FALSE,
  trans_slash = FALSE,
  force = FALSE,
  max_try = 3L,
  ...
)
```

Arguments

`xquery`	a tibble object generated by XenaQuery function.
`destdir`	specify a location to store download data. Default is system temp directory.
`download_probeMap`	if `TRUE`, also download ProbeMap data, which used for id mapping.
`trans_slash`	logical, default is `FALSE`. If `TRUE`, transform slash '/' in dataset id to '__'. This option is for backwards compatibility.
`force`	logical. if `TRUE`, force to download data no matter whether files exist. Default is `FALSE`.
`max_try`	time limit to try downloading the data.
`...`	other argument to `download.file` function

Value

a `tibble`

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
## Not run:
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub")
hosts(xe)
xe_query = XenaQuery(xe)
xe_download = XenaDownload(xe_query)

## End(Not run)
```

`XenaFilter` *Filter a XenaHub Object*

Description

One of main functions in **UCSCXenatools**. It is used to filter `XenaHub` object according to cohorts, datasets. All datasets can be found at `https://xenabrowser.net/datapages/`.

Usage

```
XenaFilter(
  x,
  filterCohorts = NULL,
  filterDatasets = NULL,
  ignore.case = TRUE,
  ...
)
```

Arguments

`x`	a XenaHub object
`filterCohorts`	default is `NULL`. A character used to filter cohorts, regular expression is supported.
`filterDatasets`	default is `NULL`. A character used to filter datasets, regular expression is supported.
`ignore.case`	if `FALSE`, the pattern matching is case sensitive and if `TRUE`, case is ignored during matching.
`...`	other arguments except `value` passed to `base::grep()`.

Value

a `XenaHub` object

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
# operate TCGA datasets
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub")
xe
# get all names of clinical data
xe2 = XenaFilter(xe, filterDatasets = "clinical")
datasets(xe2)
```

`XenaGenerate` *Generate and Subset a XenaHub Object from 'XenaData'*

Description

Generate and Subset a XenaHub Object from 'XenaData'

Usage

```
XenaGenerate(XenaData = UCSCXenaTools::XenaData, subset = TRUE)
```

Arguments

`XenaData`	a `data.frame`. Default is `data(XenaData)`. The input of this option can only be `data(XenaData)` or its subset.
`subset`	logical expression indicating elements or rows to keep.

Value

a XenaHub object.

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
# 1 get all datasets
XenaGenerate()
# 2 get TCGA BRCA
XenaGenerate(subset = XenaCohorts == "TCGA Breast Cancer (BRCA)")
# 3 get all datasets containing BRCA
XenaGenerate(subset = grepl("BRCA", XenaCohorts))
```

`XenaHub` *Generate a XenaHub Object*

Description

It is used to generate original `XenaHub` object according to hosts, cohorts, datasets or hostName. If these arguments not specified, all hosts and corresponding datasets will be returned as a `XenaHub` object. All datasets can be found at `https://xenabrowser.net/datapages/`.

Usage

```
XenaHub(
  hosts = xena_default_hosts(),
  cohorts = character(),
  datasets = character(),
  hostName = c("publicHub", "tcgaHub", "gdcHub", "icgcHub", "toilHub",
   "pancanAtlasHub", "treehouseHub", "pcawgHub", "atacseqHub", "singlecellHub",
    "kidsfirstHub")
)
```

Arguments

`hosts`	a character vector specify UCSC Xena hosts, all available hosts can be found by `xena_default_hosts()` function. `hostName` is a more recommend option.
`cohorts`	default is empty character vector, all cohorts will be returned.
`datasets`	default is empty character vector, all datasets will be returned.
`hostName`	name of host, available options can be accessed by `.xena_hosts` This is an easier option for user than `hosts` option. Note, this option will overlap `hosts`.

Value

a XenaHub object

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
## Not run:
#1 query all hosts, cohorts and datasets
xe = XenaHub()
xe
#2 query only TCGA hosts
xe = XenaHub(hostName = "tcgaHub")
xe
hosts(xe)     # get hosts
```

```
cohorts(xe)   # get cohorts
datasets(xe)  # get datasets
samples(xe)   # get samples

## End(Not run)
```

`XenaHub-class` *Class XenaHub*

Description

a S4 class to represent UCSC Xena Data Hubs

Slots

- `hosts` hosts of data hubs
- `cohorts` cohorts of data hubs
- `datasets` datasets of data hubs

`XenaPrepare` *Prepare (Load) Downloaded Datasets to R*

Description

Prepare (Load) Downloaded Datasets to R

Usage

```
XenaPrepare(
  objects,
  objectsName = NULL,
  use_chunk = FALSE,
  chunk_size = 100,
  subset_rows = TRUE,
  select_cols = TRUE,
  callback = NULL,
  comment = "#",
  na = c("", "NA", "[Discrepancy]"),
  ...
)
```

Arguments

`objects`	a object of character vector or data.frame. If `objects` is data.frame, it should be returned object of XenaDownload function. More easier way is that objects can be character vector specify local files/directory and download urls.
`objectsName`	specify names for elements of return object, i.e. names of list
`use_chunk`	default is `FALSE`. If you want to select subset of original data, please set it to `TRUE` and specify corresponding arguments: `chunk_size`, `select_direction`, `select_names`, `callback`.
`chunk_size`	the number of rows to include in each chunk
`subset_rows`	logical expression indicating elements or rows to keep: missing values are taken as false. `x` can be a representation of data frame you wanna do subset operation. Of note, the first colname of most of datasets in Xena will be set to "sample", you can use it to select rows.
`select_cols`	expression, indicating columns to select from a data frame. 'x' can be a representation of data frame you wanna do subset operation, e.g. `select_cols = colnames(x)[1:3]` will keep only first to third column.
`callback`	a function to call on each chunk, default is `NULL`, this option will overvide operations of subset_rows and select_cols.
`comment`	a character specify comment rows in files
`na`	a character vectory specify `NA` values in files
`...`	other arguments transfer to `read_tsv` function or `read_tsv_chunked` function (when `use_chunk` is `TRUE`) of `readr` package.

Value

a list contains file data, which in way of tibbles

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
## Not run:
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub")
hosts(xe)
xe_query = XenaQuery(xe)

xe_download = XenaDownload(xe_query)
dat = XenaPrepare(xe_download)

## End(Not run)
```

`XenaQuery` *Query URL of Datasets before Downloading*

Description

Query URL of Datasets before Downloading

Usage

```
XenaQuery(x)
```

Arguments

`x`	a XenaHub object

Value

a `data.frame` contains hosts, datasets and url

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub")
hosts(xe)
## Not run:
xe_query = XenaQuery(xe)

## End(Not run)
```

`XenaQueryProbeMap` *Query ProbeMap URL of Datasets*

Description

If dataset has no ProbeMap, it will be ignored.

Usage

```
XenaQueryProbeMap(x)
```

Arguments

`x`	a XenaHub object

Value

a `data.frame` contains hosts, datasets and url

Author(s)

Shixiang Wang w_shixiang@163.com

Examples

```
xe = XenaGenerate(subset = XenaHostNames == "tcgaHub")
hosts(xe)
## Not run:
xe_query = XenaQueryProbeMap(xe)

## End(Not run)
```

`XenaScan` *Scan all rows according to user input by a regular expression*

Description

`XenaScan()` is a function can be used before `XenaGenerate()`.

Usage

```
XenaScan(
  XenaData = UCSCXenaTools::XenaData,
  pattern = NULL,
  ignore.case = TRUE
)
```

Arguments

`XenaData`	a `data.frame`. Default is `data(XenaData)`. The input of this option can only be `data(XenaData)` or its subset.
`pattern`	character string containing a regular expression (or character string for `fixed = TRUE`) to be matched in the given character vector. Coerced by `as.character` to a character string if possible. If a character vector of length 2 or more is supplied, the first element is used with a warning. Missing values are allowed except for `regexpr`, `gregexpr` and `regexec`.
`ignore.case`	if `FALSE`, the pattern matching is *case sensitive* and if `TRUE`, case is ignored during matching.

Value

a `data.frame`

Examples

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

x1 %>%
  XenaGenerate()
x2 %>%
  XenaGenerate()
```

`XenaShiny` *Xena Shiny App*

Description

Xena Shiny App

Usage

```
XenaShiny()
```

`xena_default_hosts` *UCSC Xena Default Hosts*

Description

Return Xena default hosts

Usage

```
xena_default_hosts()
```

Value

A character vector include current defalut hosts

Author(s)

Shixiang Wang w_shixiang@163.com

See Also

`XenaHub()`

Index

.XenaHub *(XenaHub-class)*, 18

as.character, *21*
availTCGA, 2
availTCGA(), *5*, *11*

base::grep(), *15*

cohorts, 3

datasets, 3
downloadTCGA, 4

fetch, 5
fetch_dataset_identifiers *(fetch)*, 5
fetch_dataset_samples *(fetch)*, 5
fetch_dense_values *(fetch)*, 5
fetch_sparse_values *(fetch)*, 5

getTCGAdata, 7

has_probeMap *(fetch)*, 5
hosts, 9

regular expression, *21*

samples, 10
showTCGA, 11
showTCGA(), *5*

to_snake, 12

xena_default_hosts, 22
xena_default_hosts(), *6*
XenaBrowse, 12
XenaData, 13
XenaDataUpdate, 13
XenaDownload, 14, *19*
XenaDownload(), *5*
XenaFilter, 15
XenaFilter(), *5*
XenaGenerate, 16
XenaGenerate(), *21*
XenaHub, *3*, *10*, *12*, *15–17*, 17, *20*
XenaHub(), *22*
XenaHub-class, 18
XenaPrepare, 18
XenaPrepare(), *5*
XenaQuery, *14*, 20
XenaQuery(), *5*
XenaQueryProbeMap, 20
XenaScan, 21
XenaShiny, 22