Package 'RTCGA'

April 23, 2016

Title The Cancer Genome Atlas Data Integration

Version 1.0.2

Date 2015-01-18

Author Marcin Kosinski <m.p.kosinski@gmail.com>, Przemyslaw Biecek <przemyslaw.biecek@gmail.com>

Maintainer Marcin Kosinski <m.p.kosinski@gmail.com>

Description The Cancer Genome Atlas (TCGA) Data Portal provides a platform for researchers to search, download, and analyze data sets generated by TCGA. It contains clinical information, genomic characterization data, and high level sequence analysis of the tumor genomes. The key is to understand genomics to improve cancer care. RTCGA package offers download and integration of the variety and volume of TCGA data using patient barcode key, what enables easier data possession. This may have an benefcial infuence on impact on development of science and improvement of patients' treatment. Furthermore, RTCGA package transforms TCGA data to tidy form which is convenient to use.

BugReports https://github.com/RTCGA/RTCGA/issues

License GPL-2

LazyLoad yes

LazyData yes

Depends R (>= 3.2.0), knitr

Imports XML, assertthat, stringi, rvest, data.table, magrittr, xml2

Suggests testthat, pander, RTCGA.rnaseq, RTCGA.clinical, RTCGA.mutations

Repository Bioconductor

biocViews Software, DataImport, DataRepresentation, Preprocessing

VignetteBuilder knitr

NeedsCompilation no

RoxygenNote 5.0.1

R topics documented:

`RTCGA-package` *The Caner Genome Atlas data integration*

Description

The Cancer Genome Atlas (TCGA) Data Portal provides a platform for researchers to search, download, and analyze data sets generated by TCGA. It contains clinical information, genomic characterization data, and high level sequence analysis of the tumor genomes. The key is to understand genomics to improve cancer care. RTCGA package offers download and integration of the variety and volume of TCGA data using patient barcode key, what enables easier data possession. This may have an benefcial infuence on impact on development of science and improvement of patients' treatment. Furthermore, RTCGA package transforms TCGA data to form which is convenient to use in R statistical package. Those data transformations can be a part of statistical analysis pipeline which can be more reproducible with RTCGA

Details

For more detailed information visit **RTCGA** `wiki` on Github.

Author(s)

Marcin Kosinski [aut, cre] < `m.p.kosinski@gmail.com` >
Przemyslaw Biecek [aut] < `przemyslaw.biecek@gmail.com` >

See Also

Other RTCGA: `checkTCGA`, `datasetsTCGA`, `downloadTCGA`, `infoTCGA`, `readTCGA`

Examples

```
## Not run:
browseVignettes('RTCGA')

## End(Not run)
```

checkTCGA *Information about datasets from TCGA project*

Description

The checkTCGA function let's to check

- DataSets: TCGA datasets' names for current release date and cohort.
- Dates: TCGA datasets' dates of release.

Usage

```
checkTCGA(what, cancerType, date = NULL)
```

Arguments

what	One of DataSets or Dates.
cancerType	A character of length 1 containing abbreviation (Cohort code - http://gdac.broadinstitute.org/) of types of cancers to check for.
date	A NULL or character specifying from which date informations should be checked. By default (date = NULL) the newest available date is used. All available dates can be checked on http://gdac.broadinstitute.org/runs/ or by using checkTCGA('Dates') function. Required format 'YYYY-MM-DD'.

Details

- If what='DataSets' enables to check TCGA datasets' names for current release date and cohort.
- If what='Dates' enables to check dates of TCGA datasets' releases.

Value

- If what='DataSets' a data.frame of available datasets' names (to pass to the downloadTCGA function) and sizes.
- If what='Dates' a vector of available dates to pass to the downloadTCGA function.

See Also

Other RTCGA: RTCGA-package, datasetsTCGA, downloadTCGA, infoTCGA, readTCGA

Examples

```
############################

# names for current release date and cohort
checkTCGA('DataSets', 'BRCA' )
## Not run:
checkTCGA('DataSets', 'OV', tail(checkTCGA('Dates'))[3] )
#checkTCGA('DataSets', 'OV', checkTCGA('Dates')[5] ) # error

## End(Not run)
# dates of TCGA datasets' releases.
checkTCGA('Dates')

############################
## Not run:
# TCGA datasets' names availability for
# current release date and cancer type.

releaseDate <- '2015-08-21'
cancerTypes <- c('OV', 'BRCA')

cancerTypes %>% sapply(function(element){
  grep(x = checkTCGA('DataSets', element, releaseDate)[, 1],
      pattern = 'humanmethylation450', value = TRUE) %>%
       as.vector()
       })


## End(Not run)
```

`datasetsTCGA` *RTCGA.data - The Family of R Packages with Data from The Cancer Genome Atlas Study*

Description

Snapshots of the clinical, mutations, CNVs, rnaseq, RPPA, mRNA, miRNASeq and methylation datasets from the `2015-11-01` release date (check all dates of release with `checkTCGA('Dates')`) are included in the `RTCGA.data` family (factory) that contains 9 packages:

- **RTCGA.rnaseq** rnaseq
- **RTCGA.clinical** clinical
- **RTCGA.mutations** mutations
- **RTCGA.CNV**
- **RTCGA.RPPA**
- **RTCGA.mRNA**

- **RTCGA.miRNASeq**
- **RTCGA.methylation**
- **RTCGA.PANCAN12** (not from TCGA)

Details

For more detailed information visit **RTCGA.data** website https://rtcga.github.io/RTCGA. One can install all data packages with `installTCGA` (devel).

Author(s)

Marcin Kosinski [aut, cre] < m.p.kosinski@gmail.com >
Przemyslaw Biecek [aut] < przemyslaw.biecek@gmail.com >
Witold Chodor [aut] < witoldchodor@gmail.com >

See Also

Other RTCGA: `RTCGA-package`, `checkTCGA`, `downloadTCGA`, `infoTCGA`, `readTCGA`

Examples

```
# installation of packages containing snapshots
# of TCGA project's datasets

## Not run:

## RTCGA GitHub development newest versions
library(RTCGA)
?installTCGA

## Bioconductor releases
source('http://bioconductor.org/biocLite.R')
biocLite(RTCGA.clinical)
biocLite(RTCGA.mutations)
biocLite(RTCGA.rnaseq)
biocLite(RTCGA.CNV)
biocLite(RTCGA.RPPA)
biocLite(RTCGA.mRNA)
biocLite(RTCGA.miRNASeq)
biocLite(RTCGA.methylation)

# use cases and examples + more data info
browseVignettes('RTCGA')

## End(Not run)
```

`downloadTCGA` *Download TCGA data*

Description

Enables to download TCGA data from specified dates of releases of concrete Cohorts of cancer types. Pass a name of required dataset to the `dataSet` parameter. By default the Merged Clinical dataSet is downloaded (value `dataSet = 'Merge_Clinical.Level_1'`) from the newest available date of the release.

Usage

```
downloadTCGA(cancerTypes, dataSet = "Merge_Clinical.Level_1", destDir,
  date = NULL, untarFile = TRUE, removeTar = TRUE, allDataSets = FALSE)
```

Arguments

`cancerTypes`	A character vector containing abbreviations (Cohort code) of types of cancers to download from http://gdac.broadinstitute.org/. For easy access from R check details below.
`dataSet`	A part of the name of dataSet to be downloaded from http://gdac.broadinstitute.org/runs/. By default the Merged Clinical dataSet is downloaded (value `dataSet = 'Merge_Clinical.Level_1'`). Available datasets' names can be checked using checkTCGA function.
`destDir`	A character specifying a directory into which `dataSets` will be downloaded.
`date`	A `NULL` or character specifying from which date `dataSets` should be downloaded. By default (`date = NULL`) the newest available date is used. All available dates can be checked on http://gdac.broadinstitute.org/runs/ or by using checkTCGA function. Required format `'YYYY-MM-DD'`.
`untarFile`	Logical - should the downloaded file be untarred. Default is `TRUE`.
`removeTar`	Logical - should the downloaded `.tar` file be removed after untarring. Default is `TRUE`.
`allDataSets`	Logical - should download all datasets matching `dataSet` parameter or only the first one (without `FFPE` phrase if possible).

Details

All cohort names can be checked using: `sub( x = names( infoTCGA() ), '-counts', '' )`.

Value

No values. It only downloads files.

See Also

Other RTCGA: `RTCGA-package`, `checkTCGA`, `datasetsTCGA`, `infoTCGA`, `readTCGA`

Examples

```
dir.create( 'hre')

downloadTCGA( cancerTypes = 'ACC', dataSet = 'miR_gene_expression',
destDir = 'hre', date =  tail( checkTCGA('Dates'), 2 )[1] )

## Not run:
downloadTCGA( cancerTypes = c('BRCA', 'OV'), destDir = 'hre',
 date = tail( checkTCGA('Dates'), 2 )[1] )

## End(Not run)
```

---

infoTCGA *Information about cohorts from TCGA project*

Description

Function restores codes and counts for each cohort from TCGA project.

Usage

```
infoTCGA()
```

Value

A list with a tabular information from http://gdac.broadinstitute.org/.

See Also

Other RTCGA: RTCGA-package, checkTCGA, datasetsTCGA, downloadTCGA, readTCGA

Examples

```
infoTCGA()

(cohorts <- infoTCGA() %>%
rownames() %>%
   sub('-counts', '', x=.))
```

readTCGA *Read TCGA data to the tidy format*

Description

readTCGA function allows to read unzipped files:

- clinical data - Merge_Clinical.Level_1
- rnaseq data (genes' expressions) - rnaseqv2__illuminahiseq_rnaseqv2
- genes' mutations data - Mutation_Packager_Calls.Level
- Reverse phase protein array data (RPPA) - protein_normalization__data.Level_3
- Merge transcriptome agilent data (mRNA) - Merge_transcriptome__agilentg4502a_07_3__unc_edu__Level_3__u
- miRNASeq data - Merge_mirnaseq__illuminaga_mirnaseq__bcgsc_ca__Level_3__miR_gene_expression__data or "Merge_mirnaseq__illuminahiseq_mirnaseq__bcgsc_ca__Level_3__miR_gene_expression__data.Level_3
- methylation data - Merge_methylation__humanmethylation27
- isoforms data - Merge_rnaseqv2__illuminahiseq_rnaseqv2__unc_edu__Level_3__RSEM_isoforms_normalized_

from TCGA project. Those files can be easily downloded with downloadTCGA function. See examples.

Usage

```
readTCGA(path, dataType, ...)
```

Arguments

path	See details and examples.
dataType	One of 'clinical', 'rnaseq', 'mutations', 'RPPA', 'mRNA', 'miRNASeq', 'methylation', 'i depending on which type of data user is trying to read in the tidy format.
...	Further arguments passed to the as.data.frame.

Details

All cohort names can be checked using: sub(x = names(infoTCGA()), '-counts', '').

Parameter path specification:

- If dataType = 'clinical' a path to a cancerType.clin.merged.txt file.
- If dataType = 'mutations' a path to the unzziped folder Mutation_Packager_Calls.Level containing .maf files.
- If dataType = 'rnaseq' a path to the uzziped file rnaseqv2__illuminahiseq_rnaseqv2__unc_edu__Level_3__RSE
- If dataType = 'RPPA' a path to the unzipped file in folder protein_normalization__data.Level_3.
- If dataType = 'mRNA' a path to the unzipped file cancerType.transcriptome__agilentg4502a_07_3__unc_edu__L
- If dataType = 'miRNASeq' a path to unzipped files cancerType.mirnaseq__illuminahiseq_mirnaseq__bcgsc_ca_ or cancerType.mirnaseq__illuminaga_mirnaseq__bcgsc_ca__Level_3__miR_gene_expression__data.data.t
- If dataType = 'methylation' a path to unzipped files cancerType.methylation__humanmethylation27__jhu_usc
- If dataType = 'isoforms' a path to unzipped files cancerType.rnaseqv2__illuminahiseq_rnaseqv2__unc_edu__

Value

An output:

- If `dataType = 'clinical'` a `data.frame` with clinical data.
- If `dataType = 'rnaseq'` a `data.frame` with rnaseq data.
- If `dataType = 'mutations'` a `data.frame` with mutations data.
- If `dataType = 'RPPA'` a `data.frame` with RPPA data.
- If `dataType = 'mRNA'` a `data.frame` with mRNA data.
- If `dataType = 'miRNASeq'` a `data.frame` with miRNASeq data.
- If `dataType = 'methylation'` a `data.frame` with methylation data.
- If `dataType = 'isoforms'` a `data.frame` with isoforms data.

Author(s)

Marcin Kosinski, <m.p.kosinski@gmail.com>

Witold Chodor, <witoldchodor@gmail.com>

See Also

Other RTCGA: RTCGA-package, checkTCGA, datasetsTCGA, downloadTCGA, infoTCGA

Examples

```
## Not run:

###############
##### clinical
###############

dir.create('data')

# downloading clinical data
# dataset = "clinical" is default parameter so we may omit it
downloadTCGA( cancerTypes = c('BRCA', 'OV'),
              destDir = 'data' )


# reading datasets
sapply( c('BRCA', 'OV'), function( element ){
    folder <- grep( paste0( '(_',element,'\\.', '|','_',element,'-FFPE)', '.*Clinical'),
                    list.files('data/'),value = TRUE)
    path <- paste0( 'data/', folder, '/', element, '.clin.merged.txt')
    assign( value = readTCGA( path, 'clinical' ),
            x = paste0(element, '.clin.data'), envir = .GlobalEnv)
    })

#############
##### rnaseq
```

```
############

dir.create('data2')

# downloading rnaseq data
downloadTCGA( cancerTypes = 'BRCA',
           dataSet = 'rnaseqv2__illuminahiseq_rnaseqv2__unc_edu__Level_3__RSEM_genes_normalized__data.Level',
              destDir = 'data2' )

# shortening paths and directories
list.files( 'data2/') %>%
    file.path( 'data2', .) %>%
    file.rename( to = substr(.,start=1,stop=50))

# reading data
list.files( 'data2/') %>%
    file.path( 'data2', .) -> folder

folder %>%
    list.files %>%
    file.path( folder, .) %>%
    grep( pattern = 'illuminahiseq', x = ., value = TRUE) -> pathRNA
readTCGA( path = pathRNA, dataType = 'rnaseq' ) -> my_data


###############
##### mutations
###############

# Example directory in which untarred data will be stored
dir.create('data3')


downloadTCGA( cancerTypes = 'OV',
              dataSet = 'Mutation_Packager_Calls.Level',
              destDir = 'data3' )

# reading data
list.files( 'data3/') %>%
    file.path( 'data3', .) -> folder

readTCGA(folder, 'mutations') -> mut_file

#################
##### methylation
#################

# Example directory in which untarred data will be stored
dir.create('data4')

# Download KIRP methylation data and store it in data4 folder
cancerType = "KIRP"
downloadTCGA(cancerTypes = cancerType,
```

```
             dataSet = "Merge_methylation__humanmethylation27",
             destDir = "data4")

# Shorten path of subdirectory with KIRP methylation data
list.files(path = "data4", full.names = TRUE) %>%
    file.rename(from = ., to = file.path("data4", paste0(cancerType, ".methylation")))

# Remove manifest.txt file
list.files(path = "data4", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE) %>%
    grep("MANIFEST.txt", x = ., value = TRUE) %>%
    file.remove()

# Read KIRP methylation data
path <- list.files(path = "data4", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE)

KIRP.methylation <- readTCGA(path, dataType = "methylation")


##########
##### RPPA
##########

# Directory in which untarred data will be stored
dir.create('data5')

# Download BRCA RPPA data and store it in data5 folder
cancerType = "BRCA"
downloadTCGA(cancerTypes = cancerType,
             dataSet = "protein_normalization__data.Level_3",
             destDir = "data5")

# Shorten path of subdirectory with BRCA RPPA data
list.files(path = "data5", full.names = TRUE) %>%
    file.rename(from = ., to = file.path("data5", paste0(cancerType, ".RPPA")))

# Remove manifest.txt file
list.files(path = "data5", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE) %>%
    grep("MANIFEST.txt", x = ., value = TRUE) %>%
    file.remove()

# Read BRCA RPPA data
path <- list.files(path = "data5", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE)

BRCA.RPPA <- readTCGA(path, dataType = "RPPA")


##########
##### mRNA
##########
```

```
# Directory in which untarred data will be stored
dir.create('data6')

# Download UCEC mRNA data and store it in data6 folder
cancerType = "UCEC"
downloadTCGA(cancerTypes = cancerType,
          dataSet = "Merge_transcriptome__agilentg4502a_07_3__unc_edu__Level_3__unc_lowess_normalization_gene_
             destDir = "data6")

# Shorten path of subdirectory with UCEC mRNA data
list.files(path = "data6", full.names = TRUE) %>%
    file.rename(from = ., to = file.path("data6",paste0(cancerType, ".mRNA")))

# Remove manifest.txt file
list.files(path = "data6", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE) %>%
    grep("MANIFEST.txt", x = ., value = TRUE) %>%
    file.remove()

# Read UCEC mRNA data
path <- list.files(path = "data6", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE)

UCEC.mRNA <- readTCGA(path, dataType = "mRNA")

###############
##### miRNASeq
###############

# Directory in which untarred data will be stored
dir.create('data7')

# Download BRCA miRNASeq data and store it in data7 folder
# Remember that miRNASeq data are produced by two machines:
# Illumina Genome Analyzer and Illumina HiSeq 2000 machines
cancerType <- "BRCA"
downloadTCGA(cancerTypes = cancerType,
          dataSet = "Merge_mirnaseq__illuminaga_mirnaseq__bcgsc_ca__Level_3__miR_gene_expression__data.Level_3
             destDir = "data7")

downloadTCGA(cancerTypes = cancerType,
          dataSet = "Merge_mirnaseq__illuminahiseq_mirnaseq__bcgsc_ca__Level_3__miR_gene_expression__data.Leve
             destDir = "data7")

# Shorten path of subdirectory with BRCA miRNASeq data
list.files(path = "data7", full.names = TRUE) %>%
    sapply(function(path){
        if (grepl(pattern = "illuminaga", path)){
            file.rename(from = grep(pattern = "illuminaga", path, value = TRUE),
                     to = file.path("data7",paste0(cancerType, ".miRNASeq.illuminaga")))
        } else if (grepl(pattern = "illuminahiseq", path)){
            file.rename(from = grep(pattern = "illuminahiseq", path, value = TRUE),
```

```
                    to = file.path("data7",paste0(cancerType, ".miRNASeq.illuminahiseq")))
        }
    })

# Remove manifest.txt file
list.files(path = "data7", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE) %>%
    grep("MANIFEST.txt", x = ., value = TRUE) %>%
    file.remove()

# Read BRCA miRNASeq data
path <- list.files(path = "data7", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE)
path_illuminaga <- grep("illuminaga", path, fixed = TRUE, value = TRUE)
path_illuminahiseq <- grep("illuminahiseq", path, fixed = TRUE, value = TRUE)

BRCA.miRNASeq.illuminaga <- readTCGA(path_illuminaga, dataType = "miRNASeq")
BRCA.miRNASeq.illuminahiseq <- readTCGA(path_illuminahiseq, dataType = "miRNASeq")

BRCA.miRNASeq.illuminaga <- cbind(machine = "Illumina Genome Analyzer", BRCA.miRNASeq.illuminaga)
BRCA.miRNASeq.illuminahiseq <- cbind(machine = "Illumina HiSeq 2000", BRCA.miRNASeq.illuminahiseq)

BRCA.miRNASeq <- rbind(BRCA.miRNASeq.illuminaga, BRCA.miRNASeq.illuminahiseq)

###############
##### isoforms
###############

# Directory in which untarred data will be stored
dir.create('data8')

# Download ACC isoforms data and store it in data8 folder
cancerType = "ACC"
downloadTCGA(cancerTypes = cancerType,
          dataSet = "Merge_rnaseqv2__illuminahiseq_rnaseqv2__unc_edu__Level_3__RSEM_isoforms_normalized__data.
            destDir = "data8")

# Shorten path of subdirectory with ACC isoforms data
list.files(path = "data8", full.names = TRUE) %>%
    file.rename(from = ., to = file.path("data8",paste0(cancerType, ".isoforms")))

# Remove manifest.txt file
list.files(path = "data8", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE) %>%
    grep("MANIFEST.txt", x = ., value = TRUE) %>%
    file.remove()

# Read ACC isoforms data
path <- list.files(path = "data8", full.names = TRUE) %>%
    list.files(path = ., full.names = TRUE)

ACC.isoforms <- readTCGA(path, dataType = "isoforms")
```

```
## End(Not run)
```

Index