Package 'TCGA2STAT'

October 22, 2015

Type Package

Title Simple TCGA Data Access for Integrated Statistical Analysis in R

Version 1.2

Date 2015-10-21

Author Ying-Wooi Wan, Genevera I. Allen, Matthew L. Anderson, Zhandong Liu

Maintainer Ying-Wooi Wan <yingwoow@bcm.edu>

Depends R (>= 3.0.0)

Imports XML, parallel, CNTools

Description Automatically downloads and processes TCGA genomics and clinical data into a format convenient for statistical analyses in the R environment.

License GPL-2

URL http://www.liuzlab.org/TCGA2STAT/

LazyLoad true

VignetteBuilder knitr

Suggests knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2015-10-22 22:14:30

R topics documented:

TCGA2STAT-package *TCGA2STAT: TCGA to Statistical Analysis*

Description

This widget automatically downloads and processes TCGA genomics and clinical data into a format convenient for statistical analyses in the R environment.

Details

Package:	TCGA2STAT
Type:	Package
Version:	1.0
Date:	2015-06-12
License:	GPL-2

This package contains a set of tools to obtain TCGA data as an object that can be seamlessly integrated into statistical analysis pipelines in the R environment. The data imported by this package is the version-stamped standardized data sets hosted and maintained by the Broad GDAC Firehose (http://gdac.broadinstitute.org/).

Both genomics and clinical data of cancer patients can be conveniently imported and merged via one simple interface of the main function in the package: `getTCGA`. Users just have to specify the type of cancer, data profiling platform, and/or clinical covariates, and the specified data will be imported and processed into a gene-by-sample matrix or a list of matrices for downstream statistical analyses in R. The types of diseases and molecular platforms supported by `TCGA2STAT` are summarized in Appendix A of the package vignette.

Author(s)

Ying-Wooi Wan, Genevera I. Allen, Matthew L. Anderson, Zhandong Liu

Maintainer: Ying-Wooi Wan<yingwoow@bcm.edu>

See Also

`getTCGA`

Examples

```
library(TCGA2STAT)

#-- Default: Get only the omics-profiles, for example RNA-SeqV2
ACC.ov <- getTCGA(disease="ACC", data.type="RNASeq2")
```

```
str(ACC.ov)

#-- Get the RNA-SeqV2, all clinical data, and expression merged with
#   overall-survival for tumor samples
ACC.ov.os <- getTCGA(disease = "ACC", data.type = "RNASeq2", clinical=TRUE)

#-- Get expression and merged with specific clinical covariate
ACC.ov.eth <- getTCGA(disease = "ACC", data.type = "RNASeq2", clinical=TRUE,
                        cvars="ethnicity")
head(ACC.ov.eth$merged.dat[,1:5])
```

---

geneinfo *Gene Map for hg19*

Description

A matrix of information on genes from the human reference genome version 19; used when merging CNA/CNV data at gene level with `CNTools`.

Usage

```
data("geneinfo")
```

Format

A data frame with 22705 observations on the following 5 variables.

`chrom` - chromosome a gene is on.

`start` - starting coordinate of a gene.

`end` - ending coordinate of a gene.

`geneid` - Entrez gene ID.

`genename` - official gene symbol.

---

getTCGA *Get TCGA Data.*

Description

Obtain TCGA data from the Broad GDAC Firehose and process the data into a format ready for statistical analysis.

Usage

```
getTCGA(disease = "GBM", data.type = "RNASeq2", type = "", filter = "Y",
p = getOption("mc.cores", 2L), clinical = FALSE, cvars = "OS")
```

Arguments

disease	acronym for cancer type; default to "GBM" for glioblastoma multiforme.
data.type	genomic data profiling platform; default to "RNASeq2" for gene level RNA-Seq data from the second pipeline (RNASeqV2).
type	specific type of measurement produced by certain platforms.
filter	chromosome to be filtered out during data import; only applicable CNA or CNV data.
p	maximum number of processing cores used in parallel processing; default to the value set in "mc.cores" global option or 2.
clinical	logical value to indicate if clinical data is to be imported; default to FALSE.
cvars	clinical covariates to be merged with genomic data; default to "OS" for overall survival.

Details

Values for disease include "ACC", "BLCA", "BRCA", "CESC", "CHOL", "COAD", "COADREAD", "DLBC", "ESCA", "FPPP", "GBM", "GBMLGG", "HNSC", "KICH", "KIPAN", "KIRC", "KIRP", "LAML", "LGG", "LIHC", "LUAD", "LUSC", "MESO", "OV", "PAAD", "PCPG", "PRAD", "READ", "SARC", "SKCM", "STAD", "TGCT", "THCA", "THYM", "UCEC", "UCS", and "UVM". Values for data.type include "RNASeq2", "RNASeq", "miRNASeq", "CNA_SNP", "CNV_SNP", "CNA_CGH", "Methylation", "Mutation", "mRNA_Array", and "miRNA_Array". Note that not all combinations are permitted; Appendix A of the package vignette outlines all values of disease and data.type accommodated by TCGA2STAT.

The type parameter should only be used along with these data.type parameters:

- RNASeq - "count" for raw read counts (default); "RPKM" for normalized read counts (reads per kilobase per million mapped reads).
- miRNASeq - "count" for raw read counts (default); "rpmmm" for normalized read counts.
- Mutation - "somatic" for non-silent somatic mutations (default); "all" for all mutations.
- Methylation - "27K" platform (default); "450K" platform.
- CNA_CGH - "415K" for CGH Custom Microarray 2x415K (default); "244A" for CGH Microarray.
- mRNA_Array - "G450" for Agilent 244K Custom Gene Expression G4502A (default); "U133" for Affymetrix Human Genome U133A 2.0 Array; "Huex" for Affymetrix Human Exon 1.0 ST Array.

The Level III RNA-Seq, miRNA-Seq, mRNA-array, and miRNA-array data imported are at gene level, but not the mutation, copy number alterations/variation (CNA/CNV), and methylation data. Our package processes and aggregates the mutation and CNA/CNV data at the gene level. The mutation data imported are in MAF files, where each file contains mutations found for the particular patient, and the number of mutations differs across patients. We filter the mutation data based on status and variant classification and then aggregate the filtered data at the gene level. The Level III CNA/CNV data imported are in segments; therefore we employ the CNTools package to merge the segmented data into gene-level data. The methylation data imported is at probe level where each probe represents a CpG site. As methylation profiles at different CpG sites within the same gene could vary a lot, it would not be biological meaningful to aggregate the probe-level methylation data into gene-level data. We return the methylation data at probe level.

Value

A list containing:

dat	a matrix of dimension *gene* x *sample*.
clinical	a matrix of dimension *sample* x *clinical covariates*; NULL if clinical=FALSE
merged.dat	a matrix, which is the merged dat and clinical data as specified by cvars. Thus, each matrix of size *sample* x (*cvars* + *gene*); NULL if clinical=FALSE or cvars is not a valid name for clinical covariate.

and for methylation data, an additional element:

cpgs	a matrix of dimension *cpg sites* x *3*. The three columns are gene symbol, chromosome, and genomic coordinate for each CpG site. The order of CpG sites in this matrix is the same as the order in dat.

Examples

```
library(TCGA2STAT)
rsem.ov <- getTCGA(disease="OV", data.type="RNASeq2")
rnaseq.ov <- getTCGA(disease="OV", data.type="RNASeq", type="RPKM")
rnaseq_os.ov <- getTCGA(disease="OV", data.type="RNASeq", type="RPKM", clinical=TRUE)
```

OMICSBind *Merge Data of Two Types of Molecular Profiles.*

Description

Combine the data matrix of two types of molecular profiles for the same tumor samples.

Usage

```
OMICSBind(dat1, dat2)
```

Arguments

dat1	data matrix in *gene* x *sample* format.
dat2	data matrix in *gene* x *sample* format.

Details

This function combines data matrices of two types of molecular profiles for the same set of tumor samples in the correct patient order. The number of genes returned is the sum of genes from two input matrices; the number of samples returned will be the tumor samples common to both input matrices. Gene names in the returned data will be suffixed by "d1" or "d2" to indicate if the gene is from the first or the second input matrix.

Value

A list of three elements:

`merged.data`	a matrix in *sample* x *gene* format
`X`	a matrix in *sample* x "*gene from dat1*" format with samples in the same order as `Y`
`Y`	a matrix of size *sample* x "*gene from dat2*" format with samples in the same order as `X`

Examples

```
library(TCGA2STAT)

seq <- getTCGA(disease="OV", data.type="RNASeq2")
mut <- getTCGA(disease="OV", data.type="Mutation", type="all")

seq.mut <- OMICSBind(dat1 = seq$dat, dat2 = mut$dat)
str(seq.mut)
```

SampleSplit *Split the Data by Sample Types.*

Description

Split the TCGA data into data matrices of different sample types (normal, tumor, or recurrent tumor).

Usage

```
SampleSplit(dat)
```

Arguments

`dat`	data matrix in *gene* x *sample* format.

Value

A list of three elements:

`primary.tumor`	a matrix of tumor samples of dimension *gene* x *sample*.
`recurrent.tumor`	a matrix of recurrent tumor samples of dimension *gene* x *sample*.
`normal`	a matrix from normal samples of dimension *gene* x *sample*.

Examples

```
library(TCGA2STAT)
lusc.rnaseq2 <- getTCGA(disease="LUSC", data.type="RNASeq2")
lusc.rnaseq2.bytype <- SampleSplit(lusc.rnaseq2$dat)
```

`TumorNormalMatch` *Get Matched Tumor and Normal Samples.*

Description

Get data matrix of molecular profiles for matching tumor and normal samples.

Usage

```
TumorNormalMatch(dat)
```

Arguments

`dat`	data matrix in *gene* x *sample* format.

Details

This function returns a list of two gene-by-sample matrices. The samples in both matrices are of matching patients and are sorted at the same order.

Value

A list of two elements:

`primary.tumor`	a matrix containing tumor samples; of dimension of *gene* x *tumor samples*.
`normal`	a matrix of normal samples; of dimension *gene* x *normal samples*.

Examples

```
library(TCGA2STAT)
lusc.rnaseq2 <- getTCGA(disease="LUSC", data.type="RNASeq2")
lusc.rnaseq2.tum.norm <- TumorNormalMatch(lusc.rnaseq2$dat)
```

Index