TCGA-Assembler User Manual

Yitan Zhu[1], Yuan Ji[1,2]

1. Center for Biomedical Research Informatics, NorthShore University HealthSystem, Evanston, IL 60201
2. Department of Health Studies, The University of Chicago, Chicago, IL 60637

Email: zhuyitan@gmail.com

Feb. 4th, 2014

CAUTION: We identified a defect in Mac application "Preview.app". Do not open this pdf file in Preview and copy/paste the R code for testing. Instead use Adobe Reader (http://get.adobe.com/reader/) or Acrobat to open this pdf file and copy/paste code to R for testing.

Introduction

TCGA-Assembler can acquire and process most of TCGA public data including level-3 data of RNA-seq, DNA methylation, DNA copy number, protein expression, and miRNA-seq, and also de-identified patient clinical information. To download TCGA-Assembler, go to http://health.bsd.uchicago.edu/yji/TCGA-Assembler.htm. Click on Download Software and unzip the downloaded files to your desired location. The folder of the package is TCGA-Assembler.

The TCGA-Assembler package includes two modules, Module A and Module B. The following gives a brief introduction of the contents in the package.

- Module_A.r includes all Module A functions. Module A downloads data from the open-access HTTP directory of TCGA Data Coordinating Center (DCC), the Uniform Resource Locator (URL) of which is https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/. Each sample in TCGA has one or multiple data files produced by each assay platform. Module A assembles the data files of individual samples into data matrices, where each row is a genomic feature and each column is a sample.
- Module_B.r includes all Module B functions. Module B processes data matrix files to fulfill various data manipulation needs. Module B can process both data matrix files assembled by TCGA-Assembler Module A and data matrix files downloaded from Firehose website at the Broad Institute (https://confluence.broadinstitute.org/display/GDAC/Dashboard-Stddata).
- QuickStartGuide_Examples.r includes the code of all examples in TCGA-Assembler Quick Start Guide.
- DirectoryTraverseResult_Jan-30-2014.rda is an R data file including the URLs of all TCGA open-access data files. It was generated by running the TraverseAllDirectories function in Module A on Jan. 30th, 2014.
- SupportingFiles folder includes supporting files and annotation files used by Module B functions.
- UserManualExampleData folder holds some data files either acquired/processed by TCGA-Assembler or downloaded from Firehose website, which will be used by the examples in this user manual. There are four subfolders in the UserManualExampleData folder. The RawData.TCGA-Assembler subfolder is used to hold data downloaded from TCGA DCC by TCGA-Assembler. The ProcessedData.TCGA-Assembler subfolder is used to contain results obtained by processing the raw data files in the RawData.TCGA-Assembler subfolder. The RawData.Firehose subfolder contains data files downloaded from the Firehose website. We keep only several samples in the Firehose data files to make their sizes small, so that they can be included in the package, and the data formats are not changed. The ProcessedData.Firehose subfolder is used to contain results obtained by processing the Firehose data files in the RawData.Firehose subfolder.

Because downloading and processing TCGA data may require significant memory space depending on the size of data, we recommend using TCGA-Assembler on computers with 16GB or larger RAM.

To use TCGA-Assembler, R and R packages including HGNChelper, RCurl, httr, stringr, digest, bitops, and their dependent packages need to be installed. They are freely available from http://www.r-project.org/.

General Procedure of Using TCGA-Assembler Pipeline

Step 1: Start R, and set the TCGA-Assembler folder (i.e. the package folder) as the Present Working Directory (PWD) of R.

Step 2: Execute Module A function TraverseAllDirectories to traverse all open-access directories on TCGA DCC data server to gather the URLs of all public data files. Due to the vast amount of sub-directories (~18,000) and files (~1,300,000), this traverse process can take about an hour to complete depending upon the Internet connection speed. *A good thing is that it needs to be done only once to download TCGA public data of all cancer types and assay platforms.*

Step 3: Use the data file URLs obtained in Step 2 and other Module A functions, whose names start with "Download" to (1) acquire TCGA public data of a specific cancer type and assay platform and (2) assemble the data files of individual samples into data matrices in tab-delimited .txt files.

Step 4: Data matrix .txt files generated by TCGA-Assembler Module A or downloaded from Firehose website should be processed by corresponding Module B functions, whose names start with "Process". These functions extract useful measurements from the .txt files and import the data into R. Data quality check, removal of redundant information, and some data calculation, such as calculating gene-level copy number values, are fulfilled by these functions.

Step 5: Based on the processed data generated in Step 4, use other data processing functions in Module B to do various data manipulations, such as merging multi-platform data into a single mega data table.

We included a directory traverse result file named DirectoryTraverseResult_Jan-30-2014.rda in the package. It was generated by running the TraverseAllDirectories function on Jan. 30th, 2014. The URLs of TCGA public data files are usually stable and valid for a quite long time. So with this file, you can skip Step 2 and directly run the examples of Module A functions in this user manual.

Contents

Module A includes the following functions:

Module B includes the following functions:

DownloadClinicalData <- function(traverseResultFile, saveFolderName, cancerType, clinicalDataType, outputFileName = "")

This function downloads clinical data files of specified cancer type and save them in the specified folder.

Input arguments

traverseResultFile: a character string. The path of the directory traverse result file that includes the URLs of all open-access data files on TCGA server.

saveFolderName: a character string. The path of the directory to save downloaded clinical data files.

cancerType: a character string indicating the cancer type for which to download data. Options include ACC, BLCA, BRCA, CESC, COAD, DLBC, ESCA, GBM, HNSC, KICH, KIRC, KIRP, LAML, LGG, LIHC, LUAD, LUSC, OV, PAAD, PRAD, READ, SARC, SKCM, STAD, THCA, UCEC, UCS. Look at https://tcga-data.nci.nih.gov/tcga/ for information about cancer types.

clinicalDataType: a character vector indicating the types of clinical information to be obtained. Options include patient, drug, follow_up, and radiation. patient indicates patient information including survival, cancer grades, and others. drug indicates records of patient drug treatment. follow_up indicates patient follow up information. radiation indicates history of radiation therapy on patients.

outputFileName: a character string used to form the names of downloaded data files. It is the empty string by default.

Output clinical data files

Please refer to TCGA (http://cancergenome.nih.gov/) for information about the clinical data format. The downloaded clinical data files have names composed of outputFileName and their original file names with "__" separating the two. If outputFileName is an empty string, the names of the downloaded data files are the same as their original file names on TCGA DCC.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

source("Module_A.r");  # Load Module A functions

# Acquire clinical information of bladder urothelial carcinoma (BLCA) and
# rectum adenocarcinoma (READ) patients.

DownloadClinicalData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName =
"./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "BLCA", clinicalDataType =
c("patient", "drug", "follow_up", "radiation"));

DownloadClinicalData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName =
"./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", clinicalDataType =
c("patient", "drug"));
```

DownloadCNAData <- function(traverseResultFile, saveFolderName, cancerType, assayPlatform = "genome_wide_snp_6", inputPatientIDs = NULL, outputFileName = "")

This function downloads copy number data of samples belonging to the specified cancer type and measured by the specified assay platform. The samples include tumors, their matched normal tissues, and control samples if exist. The function downloads the data files of individual samples and combines them into tab-delimited .txt files.

Input arguments

traverseResultFile: a character string. The path of the directory traverse result file that includes the URLs of all open-access data files on TCGA server.

saveFolderName: a character string. The path of the directory to save the acquired data files.

cancerType: a character string indicating the cancer type for which to download data. Options include ACC, BLCA, BRCA, CESC, COAD, DLBC, ESCA, GBM, HNSC, KICH, KIRC, KIRP, LAML, LGG, LIHC, LUAD, LUSC, OV, PAAD, PRAD, READ, SARC, SKCM, STAD, THCA, UCEC, UCS. Look at https://tcga-data.nci.nih.gov/tcga/ for information about cancer types.

assayPlatform: a character string indicating the assay platform used to generate the data. Currently, the only option allowed (also be default) is genome_wide_snp_6, which indicates Affymetrix® Genome-Wide Human SNP Array 6.0. In TCGA, Genome-Wide Human SNP Array 6.0 provides the most abundant copy number data.

inputPatientIDs: a character vector that includes TCGA barcodes of the patients/samples, the data of which should be obtained. If it is NULL (by default), data of all samples in the specified cancer category will be acquired. Barcodes in inputPatientIDs must start with TCGA-, but do not need to be full-length and complete, because string matching for identifying the samples' data starts from the beginning of the barcodes. For details of TCGA barcodes, refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode.

outputFileName: a character string used to form the names of obtained data files. It is the empty string by default.

Output arguments

A list object of four character matrices containing the data and their descriptions retrieved from TCGA DCC. Refer to the output data file section for introduction of the contents in the character matrices.

Output data files

The DownloadCNAData function generates four tab-delimited .txt data files. The file names consist of six components, (1) outputFileName, (2) cancer type, (3) institution that generated the data, which is broad.mit.edu, (4) assay platform used to generate the data, which is genome_wide_snp_6, (5) data type, (6) the date on which the input directory traverse result file was generated. Double-underscore "__" is used to separate the six components in the file name. If outputFileName is an empty string, the file names consist of only the other five components. The file names of the four data files differ in component (5), data type, which can be hg18, hg19, nocnv_hg18, nocnv_hg19. hg18 and hg19 indicate that in preparation for segmentation, the probes are sorted based on the order of reference genome hg18 and hg19, respectively. nocnv indicates that a fixed set of probes that frequently contain germline CNVs are removed prior to segmentation. All four data files have the same format. Each row corresponds to a segment. Column one is TCGA sample barcode. Column two is chromosome id. Column three is the start position of the segment. Column four is the end position of the segment. Column five is the number of probes in the segment.

Column six is the copy number value transferred by base2 log(copy number/2), centered on 0. For more details of the data format, data type, and data generation pipeline, please refer to TCGA description at https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/cgcc/broad.mit.edu/genome_wide_snp_6/snp/broad.mit.edu_READ.Genome_Wide_SNP_6.mage-tab.1.2003.0/DESCRIPTION.txt.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

source("Module_A.r");  # Load Module A functions

# Acquire copy number data of six rectum adenocarcinoma (READ) patient samples.

READ_CNARawData = DownloadCNAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda",
saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ",
assayPlatform = "genome_wide_snp_6", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01",
"TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AF-2692-10", "TCGA-AG-4021-10"));

# Acquire copy number data of all bladder urothelial carcinoma (BLCA) patient samples.

BLCA_CNARawData = DownloadCNAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda",
saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "BLCA",
assayPlatform = "genome_wide_snp_6");
```

DownloadMethylationData <- function(traverseResultFile, saveFolderName, cancerType, assayPlatform, inputPatientIDs = NULL, outputFileName = "")

This function downloads methylation data of specified cancer type measured by Illumina HumanMethylation27 BeadChip or Illumina HumanMethylation450 BeadChip. It downloads the data files of individual samples and combines them into data matrix format. The samples to be downloaded include tumors, their matched normal tissues, and control samples if exist.

Input arguments

traverseResultFile: a character string. The path of the directory traverse result file that includes the URLs of all open-access data files on TCGA server.

saveFolderName: a character string. The path of the directory to save the acquired data matrix files.

cancerType: a character string indicating the cancer type for which to download data. Options include ACC, BLCA, BRCA, CESC, COAD, DLBC, ESCA, GBM, HNSC, KICH, KIRC, KIRP, LAML, LGG, LIHC, LUAD, LUSC, OV, PAAD, PRAD, READ, SARC, SKCM, STAD, THCA, UCEC, UCS. Look at https://tcga-data.nci.nih.gov/tcga/ for information about cancer types.

assayPlatform: a character string indicating the assay platform used to generate the data. Options include humanmethylation27 and humanmethylation450.

inputPatientIDs: a character vector that includes TCGA barcodes of the patients/samples, the data of which should be obtained. If it is NULL (by default), data of all samples in the specified cancer category will be acquired. Barcodes in inputPatientIDs must start with TCGA-, but do not need to be full-length and complete, because string matching for identifying the samples' data starts from the beginning of the barcodes. For details of TCGA barcodes, refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode.

outputFileName: a character string used to form the names of obtained data files. It is the empty string by default.

Output arguments

A character matrix containing the data and their descriptions retrieved from TCGA DCC. Refer to the description of output data file for introduction of the content in the character matrix.

Output data files

The name of the output data file consists of five components, (1) outputFileName, (2) cancer type, (3) institution that generated the data, i.e. jhu-usc.edu, (4) assay platform used to generate the data, i.e. humanmethylation27 or humanmethylation450, (5) the date on which the input directory traverse result file was generated. Double-underscore "__" is used to separate the five components in the file name. If outputFileName is an empty string, the file name consists of only the other four components. The output data file is a tab-delimited .txt file. In the file, the first column includes the indices that Illumina gives to CpG sites. The second column is gene symbol. The third column is chromosome id. The fourth column includes genomic coordinates of CpG sites. Starting from column five, each column is the data of a sample, and the first row gives the TCGA barcodes of samples.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace
```

```
source("Module_A.r");  # Load Module A functions

# Acquire humanmethylation450 data of six rectum adenocarcinoma (READ) patient samples.

READ_Methylation450RawData = DownloadMethylationData(traverseResultFile =
"./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName =
"./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform =
"humanmethylation450", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01",
"TCGA-F5-6812-01", "TCGA-AG-A01W-11", "TCGA-AG-3731-11"));

# Acquire humanmethylation27 data of all rectum adenocarcinoma (READ) patient samples.

READ_Methylation27RawData = DownloadMethylationData(traverseResultFile =
"./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName =
"./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform =
"humanmethylation27");
```

DownloadmiRNASeqData <- function(traverseResultFile, saveFolderName, cancerType, assayPlatform = "miRNASeq", inputPatientIDs = NULL, outputFileName = "")

This function downloads miRNASeq data of specified cancer type. It downloads the data files of individual samples and combines them into data matrices. The samples to be acquired include tumors, their matched normal tissues, and control samples if exist.

Input arguments

traverseResultFile: a character string. The path of the directory traverse result file that includes the URLs of all open-access data files on TCGA server.

saveFolderName: a character string. The path of the directory to save the acquired data matrix files.

cancerType: a character string indicating the cancer type for which to download data. Options include ACC, BLCA, BRCA, CESC, COAD, DLBC, ESCA, GBM, HNSC, KICH, KIRC, KIRP, LAML, LGG, LIHC, LUAD, LUSC, OV, PAAD, PRAD, READ, SARC, SKCM, STAD, THCA, UCEC, UCS. Look at https://tcga-data.nci.nih.gov/tcga/ for information about cancer types.

assayPlatform: a character string indicating the assay platform used to generate the data. Currently, the only option allowed is miRNASeq. It acquires data generated by both illuminaga_mirnaseq assay and illuminahiseq_mirnaseq assay, if the data exist.

inputPatientIDs: a character vector that includes TCGA barcodes of the patients/samples, the data of which should be obtained. If it is NULL (by default), data of all samples in the specified cancer category will be acquired. Barcodes in inputPatientIDs must start with TCGA-, but do not need to be full-length and complete, because string matching for identifying the samples' data starts from the beginning of the barcodes. For details of TCGA barcodes, refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode.

outputFileName: a character string used to form the names of obtained data files. It is the empty string by default.

Output arguments

A list object of character matrices containing the data and their descriptions retrieved from TCGA DCC. Refer to the description of output data files for introduction of the contents in the character matrices.

Output data files

For both illuminaga_mirnaseq assay and illuminahiseq_mirnaseq assay, there will be two output data files. The name of an output data file includes six components, (1) outputFileName, (2) cancer type, (3) institution that generated the data, i.e. bcgsc.ca, (4) assay used to generate the data, i.e. illuminaga_mirnaseq or illuminahiseq_mirnaseq, (5) reference genome used, i.e. NCBI36 or GRCh37, (6) the date on which the input directory traverse result file was generated. Double-underscore "__" is used to separate the six components in the file name. If outputFileName is an empty string, the file names consist of only the other five components. The two output data files of the same platform differ in the reference genome used to map the reads. NCBI36 and GRCh37 indicate Hg18 and Hg19 reference genomes, respectively. Both files are tab-delimited .txt files containing the miRNA expression data in matrix format. In the files, the first row gives TCGA barcodes of samples and the second row indicates whether a column is read count or reads per million miRNA reads mapped. The first column gives miRNA names. And starting from the second column, two columns correspond to one sample.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

source("Module_A.r");  # Load Module A functions

# Acquire miRNASeq data of six rectum adenocarcinoma (READ) patient samples.

READ_miRNASeqRawData = DownloadmiRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "miRNASeq", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AF-2689-11", "TCGA-AF-2691-11"));

# Acquire miRNASeq data of all bladder urothelial carcinoma (BLCA) patient samples.

BLCA_miRNASeqRawData = DownloadmiRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "BLCA", assayPlatform = "miRNASeq");
```

DownloadRNASeqData <-function (traverseResultFile, saveFolderName, cancerType, assayPlatform, dataType, inputPatientIDs = NULL, outputFileName = "")

This function downloads RNASeq data of specified cancer type generated by specified assay pipeline. It downloads the data files of individual samples and combines them into data matrix format. The samples to be acquired include tumors, their matched normal tissues, and control samples if exist.

Input arguments

traverseResultFile: a character string indicating the path of the directory traverse result file that includes the URLs of all open-access data files on TCGA server.

saveFolderName: a character string. The path of the directory to save the acquired data matrix files.

cancerType: a character string indicating the cancer type for which to download data. Options include ACC, BLCA, BRCA, CESC, COAD, DLBC, ESCA, GBM, HNSC, KICH, KIRC, KIRP, LAML, LGG, LIHC, LUAD, LUSC, OV, PAAD, PRAD, READ, SARC, SKCM, STAD, THCA, UCEC, UCS. Look at https://tcga-data.nci.nih.gov/tcga/ for information about cancer types.

assayPlatform: a character string indicating the assay platform and data pipeline used to generate the data. Options include RNASeqV1 and RNASeqV2. The main difference between these two is the data post-processing pipeline used. RNASeqV1 generates three files for each sample, including gene expression, exon expression, and junction expression. RNASeqV2 generates six files for each sample, including expression estimates of genes and isoforms, normalized expression values of genes and isoforms, expressions of exons and junctions. Also, this function acquires data generated by both illuminaga_mirnaseq assay and illuminahiseq_mirnaseq assay, if the data exist.

dataType: a character vector indicating which types of data should be acquired. The following table shows the available options, when assayPlatform is RNASeqV1 or RNASeqV2. Multiple types of data can be acquired simultaneously.

assayPlatform = RNASeqV1		assayPlatform = RNASeqV2	
gene.quantification	Gene expression values	rsem.genes.results	Gene expression values estimated by RSEM algorithm
exon.quantification	Exon expression values	rsem.genes.normalized_results	Normalized gene expression values
spljxn.quantification	Splice junction expression values	rsem.isoforms.results	Isoform expression values estimated by RSEM algorithm
		rsem.isoforms.normalized_results	Normalized isoform expression values
		exon_quantification	Exon expression values
		junction_quantification	Splice junction expression values

inputPatientIDs: a character vector that includes TCGA barcodes of the patients/samples whose data should be obtained. If it is NULL (by default), data of all samples in the specified cancer category will be acquired. Barcodes in inputPatientIDs must start with TCGA-, but do not need to be full-length and complete, because string matching for identifying the samples' data starts from the beginning of the barcodes. For details of TCGA barcodes, refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode.

outputFileName: a character string used to form the names of obtained data files. It is the empty string by default.

Output arguments

A list object of character matrices containing the data and their descriptions retrieved from TCGA DCC. Refer to the description of output data files for introduction of the contents in the character matrices.

Output data files

This function can generate nine different types of data files, each for one of the data types shown in the table above. The name of an output data file includes six components, (1) outputFileName, (2) cancer type, (3) institution that generated the data, which can be unc.edu or bcgsc.ca, (4) assay and post-processing pipeline used to generate the data, i.e. illuminaga_rnaseqv2, illuminaga_rnaseq, illuminahiseq_rnaseqv2, or illuminahiseq_rnaseq, (5) data type, as shown in the table above, (6) the date on which the input directory traverse result file was generated. Double-underscore "__" is used to separate the six components in the file name. If outputFileName is an empty string, the file names consist of only the other five components. For details of the formats of RNASeqV1 data types, please refer to TCGA description at https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/cgcc/unc.edu/illuminaga_rnaseq/rnaseq/unc.edu_READ.IlluminaGA_RNASeq.mage-tab.1.3.0/DESCRIPTION.txt. For details of the formats of RNASeqV2 data types, please refer to TCGA description at https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/cgcc/unc.edu/illuminahiseq_rnaseqv2/rnaseqv2/unc.edu_READ.IlluminaHiSeq_RNASeqV2.mage-tab.1.6.0/DESCRIPTION.txt

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

# Acquire normalized gene expression data and exon expression data of six rectum adenocarcinoma
# (READ) patient samples, produced by RNASeqV2 pipeline.

READ_RNASeqV2RawData = DownloadRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "RNASeqV2", dataType = c("rsem.genes.normalized_results", "exon_quantification"), inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-3732-11", "TCGA-AG-3742-11"));

# Acquire gene expression data and splice junction expression data of all READ patient samples,
# produced by RNASeqV1 pipeline.

READ_RNASeqV1RawData = DownloadRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "RNASeqV1", dataType = c("gene.quantification", "spljxn.quantification"));
```

DownloadRPPAData <- function(traverseResultFile, saveFolderName, cancerType, assayPlatform = "mda_rppa_core", inputPatientIDs = NULL, outputFileName = "")

This function downloads RPPA protein expression data of specified cancer type. It downloads the data files of individual samples and combines them into data matrix format. The samples to be acquired include tumors, their matched normal tissues, and control samples if exist.

Input arguments

traverseResultFile: a character string. The path of the directory traverse result file that includes the URLs of all open-access data files on TCGA server.

saveFolderName: a character string. The path of the directory to save the acquired data matrix files.

cancerType: a character string. The cancer type for which to download data. Options include ACC, BLCA, BRCA, CESC, COAD, DLBC, ESCA, GBM, HNSC, KICH, KIRC, KIRP, LAML, LGG, LIHC, LUAD, LUSC, OV, PAAD, PRAD, READ, SARC, SKCM, STAD, THCA, UCEC, UCS. Look at https://tcga-data.nci.nih.gov/tcga/ for information about cancer types.

assayPlatform: a character string indicating the assay platform used to generate the data. It must be mda_rppa_core, which is also the default value.

inputPatientIDs: a character vector that includes TCGA barcodes of the patients/samples whose data should be obtained. If it is NULL (by default), data of all samples in the specified cancer category will be acquired. Barcodes in inputPatientIDs must start with TCGA-, but do not need to be full-length and complete, because string matching for identifying the samples' data starts from the beginning of the barcodes. For details of TCGA barcodes, refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode.

outputFileName: a character string used to form the names of obtained data files. It is the empty string by default.

Output arguments

A character matrix containing the data and their descriptions retrieved from TCGA DCC. Refer to the description of output data file for introduction of the content in the character matrix.

Output data file

The name of the output data file includes five components, (1) outputFileName, (2) cancer type, (3) institution that generated the data, which is mdanderson.org, (4) assay platform used to generate the data, i.e. mda_rppa_core, (5) the date on which the input directory traverse result file was generated. Double-underscore "__" is used to separate the five components in the file name. If outputFileName is an empty string, the file names consist of only the other four components. The output data file is a tab-delimited .txt file containing RPPA protein expression data in matrix format. The top row gives the TCGA barcodes of samples. The first column shows protein antibody name (after "|") and corresponding gene symbols (before "|") that encode the protein. Starting from the second column, each column corresponds to a sample. For details about how the data were generated and normalized, please refer to (https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/cgcc/mdanderson.org/mda_rppa_core/protein_exp/mdanderson.org_READ.MDA_RPPA_Core.Level_3.1.0.0/DESCRIPTION.txt).

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.
```

```
source("Module_A.r");  # Load Module A functions

# Acquire RPPA protein expression data of six rectum adenocarcinoma (READ) patient samples.

READ_RPPARawData = DownloadRPPAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "mda_rppa_core", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-3582-01", "TCGA-AG-4001-01"));

# Acquire RPPA protein expression data of all bladder urothelial carcinoma (BLCA) patient samples.

BLCA_RPPARawData = DownloadRPPAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "BLCA", assayPlatform = "mda_rppa_core");
```

TraverseAllDirectories <- function(entryPoint, fileLabel)

This function traverses sub-directories in the open-access HTTP directory on TCGA DCC data server and obtains the URLs of public TCGA data files. The URLs will be saved in a .rda file in the present working directory. The file name is composed of the input argument fileLabel and the date on which this function is executed. Thus through the file name we can tell how old the URL information included in the file is.

Input arguments

entryPoint: URL of the directory on TCGA DCC server that needs to be traversed. To obtain the URLs of all open-access TCGA data files, use https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/ as the entry point.

fileLabel: a string to be included in the name of the file to store the traverse results.

Output directory traverse result file

The name of the directory traverse result file is composed of the input argument fileLabel and the date on which this function is executed. There are three variables included in the file.

file_url: a character vector including the URLs of all data files under the entryPoint.

upper_file_url: the same as file_url, except that all characters in upper_file_url are in uppercase to facilitate case insensitive string matching.

dir_url: a character vector including the URLs of all sub-directories under the entryPoint.

Note

Currently, there are about 1,300,000 data files and 18,000 sub-directories (based on numbers obtained in Jan., 2013) in the open-access directory on TCGA DCC data server. To traverse all the sub-directories takes about an hour to complete depending upon Internet connection speed. A good thing is that it needs to be done only once for downloading the data of all various cancer types and assay platforms. In the package, we include a directory traverse result file obtained on Jan. 30th, 2014. The file name is DirectoryTraverseResult_Jan-30-2014.rda, which includes the URLs of all public data files existing on TCGA data server on Jan. 30th, 2014. You can use this directory traverse result file to quickly start downloading TCGA data and skip the step of running the TraverseAllDirectories function. The file URLs are usually stable and valid for quite a long time.

Example

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

source("Module_A.r");  # Load Module A functions

# Get the URLs of public data files of all cancer types and all assay platforms.

TraverseAllDirectories(entryPoint = "https://tcga-
data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/", fileLabel =
"DirectoryTraverseResult");

# Get the URLs of all rectum adenocarcinoma (READ) public data files. The output directory traverse
# result file can be used to download READ data generated by all different assay platforms.

TraverseAllDirectories(entryPoint = "https://tcga-
data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/read/", fileLabel =
"READ_DirectoryTraverseResult");
```

CalculateSingleValueMethylationData<-function(input, regionOption, DHSOption, outputFileName, outputFileFolder, chipAnnotationFile = "./SupportingFiles/MethylationChipAnnotation.rda")

This function does the following

(1) Calculate an average methylation value for each gene based on certain CpG sites according to the specified option (see details of regionOption and DHSOption).

(2) Draw and save a box plot of the obtained single-value methylation data. The file name of the box plot picture is composed of outputFileName, regionOption, DHSOption, and "boxplot.png", with double-underscore "__" separating them.

(3) Save the single-value methylation data as a tab-delimited .txt file. The first two columns are gene symbol and the single-value type that shows the regionOption and DHSOption used to calculate the data with "|" in between to separate them. The other columns are data of individual samples. The first row gives TCGA barcodes of samples (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode). The name of the file is composed of outputFileName, regionOption, and DHSOption with double-underscore "__" separating them.

(4) Save the single-value methylation data as an R data file (.rda) that includes two variables. The first variable is Des, which is a two-column character matrix including gene symbol and the single-value type that shows the regionOption and DHSOption used to calculate the data, with "|" in between to separate them. Des serves as the description of the second variable, Data, which is a numeric matrix. Each column in Data is a sample and each row corresponds to a gene. The column names of Data are TCGA barcodes of samples. The name of the file is composed of outputFileName, regionOption, and DHSOption with double-underscore "__" separating them.

The output .txt file and .rda file contain the same data. Having the same data in two different file formats is for the convenience of using data under different software environments.

Input arguments

input: a list object containing the methylation data based on which a single methylation value needs to be calculated for each gene. This list object can be generated by the ProcessMethylation27Data, ProcessMethylation450Data, and MergeMethylationData functions. It is a list object formed by two variables Des and Data. Des is a four-column character matrix including Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. Data is a numeric matrix containing methylation data of samples, where each column corresponds to a sample and the column names are the TCGA barcodes of samples. Des serves as the description of Data. Another way to generate this list object is to load the .rda file produced by the ProcessMethylation27Data, ProcessMethylation450Data, and MergeMethylationData functions, and form a list object using the loaded Des and Data variables.

regionOption: a character string indicating for which genomic region of a gene the average methylation value should be calculated, based on HumanMethylation450 BeadChip annotations provided by Illumina (http://support.illumina.com/downloads/humanmethylation450_15017482_v1-2_product_files.ilmn). Available options are TSS1500, TSS200, 5'UTR, 1stExon, Body, 3'UTR, and All. TSS1500: within 1500 bps of a Transcription Start Site (TSS). TSS200: within 200 bps of a TSS. 5'UTR: 5' untranslated region. 1stExon: first exon. Body: gene body. 3'UTR: 3' untranslated region. All: all CpG sites associated with a gene no matter which genomic region the CpG sites are in. For details about the definitions of the region options, please refer to the Illumina annotations. TSS1500 region includes TSS200 region as a subset.

DHSOption: a character string that can be DHS, notDHS, or Both, indicating whether only CpG sites that are DNAse hypersensitive will be included in the calculation. DHS selects CpG sites that are DNAse hypersensitive. notDHS selects CpG sites that are not labeled as DNAse hypersensitive. Both selects all CpG sites no matter whether they are DNAse hypersensitive or not. The DNAse hypersensitivity of CpG sites are experimentally determined and provided by Illumina chip annotations.

outputFileName: a character string used to form the names of output data files and box plot picture file.

outputFileFolder: a character string indicating the directory in which the output files will be saved.

chipAnnotationFile: a character string indicating the path of the chip annotation file to be used by the function, which is the MethylationChipAnnotation.rda file in the SupportingFiles folder in the package.

Output argument

A list object of two variables. The first variable is Des, a two-column character matrix including gene symbol and the single-value type that shows the regionOption and DHSOption used to calculate the data with "|" in between to separate them. Des serves as the description of the second variable, Data, which is a numeric data matrix. Each column in the data matrix is a sample and each row corresponds to a gene. The column names of Data are the TCGA barcodes of samples.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download humanmethylation450 data of six rectum adenocarcinoma (READ) patient samples

Methylation450RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "humanmethylation450", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-A01W-11", "TCGA-AG-3731-11"));

# Process the downloaded humanmethylation450 data and import the data into R.

Methylation450Data = ProcessMethylation450Data(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation450__Jan-30-2014.txt", outputFileName = "READ__humanmethylation450", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Calculate average methylation values of all CpG sites associated with the genes.

Methylation450_All_Both = CalculateSingleValueMethylationData(input = Methylation450Data, regionOption = "All", DHSOption = "Both", outputFileName = "READ__humanmethylation450__SingleValue", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler", chipAnnotationFile = "./SupportingFiles/MethylationChipAnnotation.rda");

# Calculate average methylation values of CpG sites within TSS200 region of the genes.

Methylation450_TSS200_Both = CalculateSingleValueMethylationData(input = Methylation450Data, regionOption = "TSS200", DHSOption = "Both", outputFileName =
```

```
"READ__humanmethylation450__SingleValue", outputFileFolder =
"./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Calculate average methylation values of CpG sites that are within TSS1500 region of the genes
# and are DNAse hypersensitive.

Methylation450_TSS1500_DHS = CalculateSingleValueMethylationData(input = Methylation450Data,
regionOption = "TSS1500", DHSOption = "DHS", outputFileName =
"READ__humanmethylation450__SingleValue", outputFileFolder =
"./UserManualExampleData/ProcessedData.TCGA-Assembler", chipAnnotationFile =
"./SupportingFiles/MethylationChipAnnotation.rda");
```

CheckGeneSymbol<-function(Des)

This function identifies obsolete gene symbols and gene symbol errors caused by auto-conversion in spreadsheet programs, like Excel, and corrects them to official HGNC gene symbols. Symbols that are not known errors, obsolete gene symbols, or official gene symbols will be left unchanged.

Input argument

Des: a character matrix containing the description of genomic features. One column of the matrix MUST be gene symbol, and its column name MUST be "GeneSymbol".

Output argument

The output is a character matrix with the same structure of Des, but with the gene symbols checked and corrected.

Note

If a symbol can not be uniquely mapped to one official gene symbol, for example a symbol is the alias for more than one genes, all official gene symbols that are associated with this alias will be included in the corrected gene symbol, with triple-underscore "___" separating each involved official gene symbol.

Example

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download humanmethylation27 data of six rectum adenocarcinoma (READ) patient samples

Methylation27RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "humanmethylation27", inputPatientIDs = c("TCGA-AG-3583-01", "TCGA-AG-A032-01", "TCGA-AF-2692-11", "TCGA-AG-4001-01", "TCGA-AG-3608-01", "TCGA-AG-3574-01"));

# Process the downloaded humanmethylation27 data and import the data into R

Methylation27Data = ProcessMethylation27Data(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation27__Jan-30-2014.txt", outputFileName = "READ__humanmethylation27", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

Des = Methylation27Data$Des;

# We purposely make an symbol error to simulate mistakes that Excel may
# make by transferring gene symbol to date format.

ID = which(Des[, "GeneSymbol"] == "SEPT7");  # SEPT7 is an official gene symbol

# Change SEPT7 to a wrong gene symbol in date format

Des[ID, "GeneSymbol"] = "7-Sep";

# Check and Correct gene symbol using the CheckGeneSymbol function
```

```
Des = CheckGeneSymbol(Des);
print(Des[ID, "GeneSymbol"]);
```

CombineMultiPlatformData<-function(inputDataList, combineStyle = "Intersect")

This function combines multi-platform datasets into a single mega data table, through matching of patient samples and genomic features. There are two possible ways to match patient samples across platforms. One is to identify samples measured by all assay platforms and merge the multi-platform data of these common samples, which is called Intersect and also the default setting of this function. The other is to include a sample as long as it is measured by at least one assay platform, which is called Union. Matching genomic features refers to putting together the multi-platform data of the same gene (i.e. making them adjacent rows in the data table). Currently, this function works on combining five different types of data, including gene expression, protein expression, DNA Methylation, DNA copy number, and miRNA expression. To match genomic features, the data must contain the information of genes associated with the genomic features (such as through the gene symbols in Des variable produced by many data processing functions in this package). DNA copy number data must be preprocessed by the ProcessCNAData function to get gene-level copy number values for combining with data from other platforms.

Input argument

inputDataList: a vector of list objects. Each element in the vector is a list object of three variables that represent one dataset to be combined. The names of the three variables are Des, Data, and dataType. Des is a character matrix including descriptions of genomic features. One column in Des must be gene symbols and with a column name of "GeneSymbol", because matching of genomic features is based on this column. Data is a numeric matrix containing the data. Each row is a genomic feature and each column is a sample. The column names must be the TCGA sample barcodes (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode). dataType is a character string indicating the type of data. Options of dataType include GeneExp, ProteinExp, Methylation, CNA, and miRNAExp, standing for gene expression, protein expression, DNA methylation, DNA copy number, and miRNA expression, respectively.

combineStyle: a character string indicating how the samples should be combined. Intersect, which is the default setting, selects samples that are measured by all assay platforms. Union includes a sample as long as it is measured by at least one assay platform.

Output argument

A list object of two variables Des and Data.

Des is a character matrix including descriptions of genomic features. Des has three columns. The first column is gene symbol. The second column is platform, which can be GE, PE, ME, CN, and miRExp, representing gene expression, protein expression, methylation, copy number, and miRNA expression, respectively. The third column is a description of genomic features. If the platform is GE, the description is Entrez ID of the gene. If the platform is PE, the description is the name of the protein antibody used in the RPPA assay. For CN platform, the description shows the chromosome ID and strand of gene. For miRExp platform, the description column is empty. For ME platform, if the data are single-value methylation data calculated by the CalculateSingleValueMethylationData function, the description column gives the single-value type indicating how the data were calculated (refer to CalculateSingleValueMethylationData function for the definition of single-value type); if the data are methylation data of CpG sites, the description column gives the Illumina ID, chromosome ID, and genomic coordinate of the CpG sites with "|" separating them.

Data is a numeric matrix including the merged data. Each row is a genomic feature and each column is a sample. The column names of Data are the TCGA barcodes of samples with their first 15 characters, which indicate the tissue source site, patient index, and sample type. Rows of Data are ordered so that genomic features of the same gene are adjacent in the matrix. Des serves as the description of Data.

Note
If there are more than one sample of the same patient and the same tissue type measured by an assay platform (which is actually a rare case), only one of the samples will be kept for data combining.

Example

```
# In this example, we will first download and process various kinds of data of several rectum
# adenocarcinoma (READ) patient samples, and then merge them into single mega table using
# both the Intersect approach and Union approach.  The present working directory of R must
# be TCGA-Assembler, i.e. the package folder, to run the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download and process copy number data of six READ patient samples.

CNARawData = DownloadCNAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda",
saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ",
assayPlatform = "genome_wide_snp_6", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01",
"TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AF-2692-10", "TCGA-AG-4021-10"));

GeneLevelCNA = ProcessCNAData(inputFilePath = "./UserManualExampleData/RawData.TCGA-
Assembler/READ__broad.mit.edu__genome_wide_snp_6__hg19__Jan-30-2014.txt", outputFileName =
"READ__genome_wide_snp_6__GeneLevelCNA", outputFileFolder =
"./UserManualExampleData/ProcessedData.TCGA-Assembler", refGenomeFile =
"./SupportingFiles/Hg19GenePosition.txt");

# Download and process humanmethylation450 data of six READ patient samples

Methylation450RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-
30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType =
"READ", assayPlatform = "humanmethylation450", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-
01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-A01W-11", "TCGA-AG-3731-11"));

Methylation450Data = ProcessMethylation450Data(inputFilePath =
"./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation450__Jan-
30-2014.txt", outputFileName = "READ__humanmethylation450", outputFileFolder =
"./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Calculate single-value methylation data.

Methylation450_TSS1500_DHS = CalculateSingleValueMethylationData(input = Methylation450Data,
regionOption = "TSS1500", DHSOption = "DHS", outputFileName =
"READ__humanmethylation450__SingleValue", outputFileFolder =
"./UserManualExampleData/ProcessedData.TCGA-Assembler", chipAnnotationFile =
"./SupportingFiles/MethylationChipAnnotation.rda");

# Download and process miRNA-seq data of six READ patient samples

miRNASeqRawData = DownloadmiRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-
2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType =
"READ", assayPlatform = "miRNASeq", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-
G5-6572-01", "TCGA-F5-6812-01", "TCGA-AF-2689-11", "TCGA-AF-2691-11"));
```

```
miRNASeqData = ProcessmiRNASeqData(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__bcgsc.ca__illuminahiseq_mirnaseq__NCBI36__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_mirnaseq", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Download and process normalized gene expression data of six READ patient samples

RNASeqRawData = DownloadRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "RNASeqV2", dataType = "rsem.genes.normalized_results", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-3732-11", "TCGA-AG-3742-11"));

GeneExpData = ProcessRNASeqData(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__unc.edu__illuminahiseq_rnaseqv2__rsem.genes.normalized_results__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_rnaseqv2__GeneExp", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler", dataType = "GeneExp", verType = "RNASeqV2");

# Download and process RPPA protein expression data of six READ patient samples

RPPARawData = DownloadRPPAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "mda_rppa_core", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-3582-01", "TCGA-AG-4001-01"));

RPPAData = ProcessRPPADataWithGeneAnnotation(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__mdanderson.org__mda_rppa_core__Jan-30-2014.txt", outputFileName = "READ__mda_rppa_core", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Put multi-modal data in a vector of list objects to be inputted into CombineMultiPlatformData function.

inputDataList = vector("list", 5);

inputDataList[[1]] = list(Des = GeneExpData$Des, Data = GeneExpData$Data, dataType = "GeneExp");

inputDataList[[2]] = list(Des = Methylation450_TSS1500_DHS$Des, Data = Methylation450_TSS1500_DHS$Data, dataType = "Methylation");

inputDataList[[3]] = list(Des = GeneLevelCNA$Des, Data = GeneLevelCNA$Data, dataType = "CNA");

inputDataList[[4]] = list(Des = RPPAData$Des, Data = RPPAData$Data, dataType = "ProteinExp");

inputDataList[[5]] = list(Des = miRNASeqData$Des, Data = miRNASeqData$Data, dataType = "miRNAExp");

# Merge multi-platform data using Intersect approach.

MergedData = CombineMultiPlatformData(inputDataList = inputDataList);

# Merge multi-platform data using Union approach.

MergedData = CombineMultiPlatformData(inputDataList = inputDataList, combineStyle = "Union");
```

ExtractTissueSpecificSamples<-function(inputData, tissueType, singleSampleFlag, sampleTypeFile = "./SupportingFiles/TCGASampleType.txt")

This function extracts from the input data matrix a subset of the data belonging to the specified tissue types.

Input arguments

inputData: a numeric matrix containing data. Each row is a genomic feature and each column is a sample. The column names of the data matrix must be TCGA sample barcodes, which must start with TCGA- and have a length no shorter than 15 characters. Refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode for details of TCGA barcodes.

tissueType: a character vector indicating for which tissue types to extract data. The first column in the following table gives the options of tissueType. The second column gives the interpretation of the options. And the third column shows the sample categories, i.e. tumor or normal.

Option	Tissue Type	Category
TP	Primary Solid Tumor	Tumor
TR	Recurrent Solid Tumor	Tumor
TB	Primary Blood Derived Cancer - Peripheral Blood	Tumor
TRBM	Recurrent Blood Derived Cancer - Bone Marrow	Tumor
TAP	Additional - New Primary	Tumor
TM	Metastatic	Tumor
TAM	Additional Metastatic	Tumor
THOC	Human Tumor Original Cells	Tumor
TBM	Primary Blood Derived Cancer - Bone Marrow	Tumor
NB	Blood Derived Normal	Normal
NT	Solid Tissue Normal	Normal
NBC	Buccal Cell Normal	Normal
NEBV	EBV Immortalized Normal	Normal
NBM	Bone Marrow Normal	Normal

singleSampleFlag: a logical variable. If it is TRUE, when there are multiple samples of the same patient and the same tissue type, only one sample is kept; if it is FALSE, all samples are kept.

sampleTypeFile: a character string indicating the path of the TCGA sample type file to be used by the function, which is TCGASampleType.txt in the SupportingFiles folder in the package.

Output argument

A numeric matrix containing the extracted data. Each column in the matrix is a sample and each row is a genomic feature. The column names of the matrix are the TCGA barcodes of samples.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions
```

```
source("Module_B.r");   # Load Module B functions

# Download humanmethylation27 data of six rectum adenocarcinoma (READ) patient samples

Methylation27RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "humanmethylation27", inputPatientIDs = c("TCGA-AG-3583-01", "TCGA-AG-A032-01", "TCGA-AF-2692-11", "TCGA-AG-4001-01", "TCGA-AG-3608-01", "TCGA-AG-3574-01"));

# Process the downloaded humanmethylation27 data and import the data into R

Methylation27Data = ProcessMethylation27Data(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation27__Jan-30-2014.txt", outputFileName = "READ__humanmethylation27", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Extract methylation data of primary solid tumor samples.

ExtractedData__TP = ExtractTissueSpecificSamples(inputData = Methylation27Data$Data, tissueType = "TP", singleSampleFlag = FALSE, sampleTypeFile = "./SupportingFiles/TCGASampleType.txt");

# Extract methylation data of primary solid tumor samples and solid tissue normal samples.

ExtractedData__TP_NT = ExtractTissueSpecificSamples(inputData = Methylation27Data$Data, tissueType = c("TP", "NT"), singleSampleFlag = TRUE);
```

MergeMethylationData<-function(input1, input2, outputFileName, outputFileFolder)

This function combines two methylation datasets together by doing the following:

(1) Identify the CpG sites that appear in both datasets, and combine the data of these CpG sites.

(2) Perform quantile normalization on the combined data.

(3) Draw and save a box plot of the combined data before and after normalization for quality control purpose. The picture file names are composed of outputFileName and "__BeforeNormalizationBoxplot.png" or "__AfterNormalizationBoxplot.png".

(4) Save the normalized combined data as a tab-delimited .txt file. The first four columns are Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. The other columns are data of individual samples. The first row gives TCGA barcodes of samples (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode).

(5) Save the normalized combined data as an R data file (.rda) that includes two variables. The first variable is Des, which is a character matrix including the four-column description of CpG sites. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in the matrix corresponds to a sample and the column names are TCGA barcodes of samples.

The output .txt file and .rda file contain the same data. Having the same data in two different file formats is for the convenience of using data under different software environments.

Input arguments

input1: a list object containing a methylation dataset to be merged. It is a list of two variables. One variable is Des, which is a four-column character matrix including Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. The other variable is Data, a numeric matrix containing methylation data. Each column in Data corresponds to a sample with column names showing the TCGA barcodes of samples. Des serves as the description of Data. This list object of input data can be generated by the ProcessMethylation27Data, ProcessMethylation450Data, or MergeMethylationData function. Another way to generate this list object is to load the .rda data file produced by the ProcessMethylation27Data or ProcessMethylation450Data function, and form a list using the loaded Des and Data.

input2: a list object containing the other methylation dataset to be merged, which should be generated in the same way as input1.

outputFileName: a character string to form the names of output files.

outputFileFolder: a character string indicating the path of the directory in which the output files will be saved.

Output argument

A list object formed by two variables. The first variable is Des, which is a character matrix including a four-column description of CpG sites, i.e. Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in Data corresponds to a sample and the column names of Data are TCGA barcodes of samples.

Note

MergeMethylationData function can be used to merge only methylation datasets that include Illumina IDs of CpG sites as description. It can NOT be used to merge methylation dataset without Illumina IDs of CpG sites.

Example

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download humanmethylation27 data of six rectum adenocarcinoma (READ) patient samples

Methylation27RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "humanmethylation27", inputPatientIDs = c("TCGA-AG-3583-01", "TCGA-AG-A032-01", "TCGA-AF-2692-11", "TCGA-AG-4001-01", "TCGA-AG-3608-01", "TCGA-AG-3574-01"));

# Process the downloaded humanmethylation27 data and import the data into R

Methylation27Data = ProcessMethylation27Data(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation27__Jan-30-2014.txt", outputFileName = "READ__humanmethylation27", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Download humanmethylation450 data of six READ patient samples

Methylation450RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "humanmethylation450", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-A01W-11", "TCGA-AG-3731-11"));

# Process the downloaded humanmethylation450 data and import the data into R

Methylation450Data = ProcessMethylation450Data(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation450__Jan-30-2014.txt", outputFileName = "READ__humanmethylation450", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Merge the humanmethylation27 data and humanmethylation450 data

Methylation27_450_Merged = MergeMethylationData(input1 = Methylation27Data, input2 = Methylation450Data, outputFileName = "READ__humanmethylation27_450_merged", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");
```

ProcessCNAData<-function(inputFilePath, outputFileName, outputFileFolder, refGenomeFile)

This function processes DNA copy number data files acquired by TCGA-Assembler Module A or downloaded from Firehose website, and imports the data into R. It does the following:

(1) Calculate gene-level copy number value, which is the average copy number of the genomic region of a gene.

(2) For gene-level copy number data, check and correct the gene identifiers to official gene symbols.

(3) Draw and save a box plot of the gene-level copy number data for quality control purpose. The picture file name is composed of outputFileName and "__boxplot.png".

(4) Save the gene-level copy number data as a tab-delimited .txt file. The first three columns are descriptions of genomic features including gene symbol, chromosome ID, and strand. The other columns are data of individual samples. The first row gives TCGA sample barcode (see reference at https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode). The file name is composed of outputFileName and ".txt".

(5) Save the gene-level copy number data as an R data file (.rda), which includes two variables. The first variable is Des, a character matrix including gene symbol, chromosome ID, and strand in the first, second, and third column, respectively. It describes the second variable, Data, which is a numeric matrix. Each column in Data is a sample and each row corresponds to a gene. The column names of Data are sample barcodes. The file name is composed of outputFileName and ".rda".

The output .txt file and .rda file contain the same data. Having the same data in two different file formats is for the convenience of using data under different software environments.

Input arguments

inputFilePath: a character string indicating the path of the input DNA copy number data file acquired by TCGA-Assembler Module A or downloaded from Firehose.

outputFileName: a character string to form the names of output files.

outputFileFolder: a character string indicating the path of the directory to which the output files will be saved.

refGenomeFile: a character string indicating the path of gene genomic position file that will be used. There are two gene position files in the SupportingFiles folder of the package. Use Hg18GenePosition.txt or Hg19GenePosition.txt, if the input copy number data was generated based on reference genome Hg18 or Hg19, respectively.

Output argument

A list object formed by two variables Des and Data. Des is a character matrix of three columns including gene symbol, chromosome ID, and strand in the first, second, and third column, respectively. It describes the second variable, Data, which is a numeric matrix of gene-level copy number. Each column in Data is a sample and each row corresponds to a gene. The column names of Data are sample barcodes.

Examples:

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions
```

```
source("Module_B.r");   # Load Module B functions

# Download DNA copy number data of six rectum adenocarcinoma (READ) patient samples.

CNARawData = DownloadCNAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda",
saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ",
assayPlatform = "genome_wide_snp_6", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01",
"TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AF-2692-10", "TCGA-AG-4021-10"));

# Process the downloaded copy number data and calculate an average copy number for each gene.
# Save results to subfolder ProcessedData.TCGA-Assembler in the UserManualExampleData folder.

READ.GeneLevel.CNA = ProcessCNAData(inputFilePath = "./UserManualExampleData/RawData.TCGA-
Assembler/READ__broad.mit.edu__genome_wide_snp_6__hg19__Jan-30-2014.txt", outputFileName =
"READ__genome_wide_snp_6__GeneLevelCNA", outputFileFolder =
"./UserManualExampleData/ProcessedData.TCGA-Assembler", refGenomeFile =
"./SupportingFiles/Hg19GenePosition.txt");

# Process READ copy number data downloaded from Firehose website and calculate average copy
# number value. Save results to ./UserManualExampleData/ProcessedData.Firehose/.

READ.GeneLevel.CNA = ProcessCNAData(inputFilePath =
"./UserManualExampleData/RawData.Firehose/READ.snp__genome_wide_snp_6__broad_mit_edu__hg18
__Level_3__segmented_scna_hg18__seg.seg.txt", outputFileName =
"READ__genome_wide_snp_6__GeneLevelCNA", outputFileFolder =
"./UserManualExampleData/ProcessedData.Firehose", refGenomeFile =
"./SupportingFiles/Hg18GenePosition.txt");
```

ProcessMethylation27Data<-function(inputFilePath, outputFileName, outputFileFolder, fileSource = "TCGA-Assembler")

This function processes HumanMethylation27 BeadChip data file either acquired by TCGA-Assembler Module A or downloaded from Firehose, and imports the data into R. It does the following.

(1) For data files downloaded from Firehose, remove redundant columns in the data. Firehose HumanMethylation27 data file includes replicated columns of probe descriptions, i.e. gene symbol, chromosome ID, genomic coordinate, which are identical for each sample.

(2) Check whether the gene symbols are official HGNC gene symbols. If not, correct them.

(3) When a CpG site corresponds to more than one gene symbol, duplicate the measurements of the CpG site (a row in the data matrix) for each gene symbol.

(4) Draw and save a box plot of the data for quality control purpose. The picture file name is composed of outputFileName and "__boxplot.png".

(5) Save the processed data as a tab-delimited .txt file. The first four columns are Illumina ID of CpG site, gene symbol, chromosome ID, and genome coordinate. The other columns are data of individual samples. The top row gives the TCGA sample barcodes (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode).

(6) Save the processed data as an R data file (.rda), which includes two variables Des and Data. Des is a character matrix including the four-column description of CpG sites. It serves as the description of Data, which is a numeric matrix. Each column in the data matrix is a sample and the column names are sample barcodes.

The output .txt file and .rda file contain the same data. Having the same data in two different file formats is for the convenience of using data under different software environments.

Input arguments

inputFilePath: a character string indicating the path of input data file acquired by TCGA-Assembler Module A or downloaded from Firehose website. It should be a tab-delimited .txt file.

outputFileName: a character string to form the names of output files.

outputFileFolder: a character string indicating the path of the directory to which the output files will be saved.

fileSource: a character string, either TCGA-Assembler or Firehose. TCGA-Assembler, which is the default setting, indicates that the input data file was acquired by TCGA-Assembler Module A and Firehose indicates that the input data file was downloaded from Firehose website.

Output argument:

A list object of two variables, which are Des and Data. Des is a character matrix of the four-column description of CpG sites including Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in the matrix is a sample and the column names are the TCGA sample barcodes.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace
```

```
source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download humanmethylation27 data of six rectum adenocarcinoma (READ) patient samples

Methylation27RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "humanmethylation27", inputPatientIDs = c("TCGA-AG-3583-01", "TCGA-AG-A032-01", "TCGA-AF-2692-11", "TCGA-AG-4001-01", "TCGA-AG-3608-01", "TCGA-AG-3574-01"));

# Process the downloaded humanmethylation27 data and save the results to
# ./UserManualExampleData/ProcessedData.TCGA-Assembler

Methylation27Data = ProcessMethylation27Data(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation27__Jan-30-2014.txt", outputFileName = "READ__humanmethylation27", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Process READ HumanMethylation27 data downloaded from Firehose website and
# save the results to ./UserManualExampleData/ProcessedData.Firehose

Methylation27Data = ProcessMethylation27Data(inputFilePath = "./UserManualExampleData/RawData.Firehose/READ.methylation__humanmethylation27__jhu_usc_edu__Level_3__within_bioassay_data_set_function__data.data.txt", outputFileName = "READ__humanmethylation27", outputFileFolder = "./UserManualExampleData/ProcessedData.Firehose", fileSource = "Firehose");
```

ProcessMethylation450Data<-function(inputFilePath, outputFileName, outputFileFolder, fileSource = "TCGA-Assembler")

This function processes HumanMethylation450 BeadChip data file either acquired by TCGA-Assembler Module A or downloaded from Firehose website, and imports the data into R. It does the following.

(1) For HumanMethylation450 data file downloaded from Firehose website, remove redundant columns in the data. Firehose HumanMethylation450 data file includes replicated columns of probe descriptions, i.e. gene symbol, chromosome ID, genomic coordinate, which are identical for each sample. This redundant information makes the file size very large, more than 10GB for some cancer types, which produces difficulties when loading the data into software environment for data analysis, such as R.

(2) Check whether the gene symbols are official HGNC gene symbols. If not, correct them.

(3) When a CpG site corresponds to more than one gene symbol, duplicate the measurements of the CpG site (a row in the data matrix) for each gene symbol that it stands for.

(4) Draw and save a box plot of the data for quality control purpose. The picture file name is composed of outputFileName and "__boxplot.png".

(5) Save the processed data as a tab-delimited .txt file. The first four columns are Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. The other columns are data of individual samples. And the top row shows the TCGA barcodes of samples (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode).

(6) Save the processed data as an R data file (.rda), which contains two variables Des and Data. Des is a character matrix including the four-column description of CpG sites. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in Data is a sample and the column names of Data are TCGA sample barcodes.

The output .txt file and .rda file contain the same data. Having the same data in two different file formats is for the convenience of using data under different software environments.

Input arguments

inputFilePath: a character string indicating the path of the input data file, either acquired by TCGA-Assembler Module A or downloaded from Firehose website. It should be a tab-delimited .txt file.

outputFileName: a character string to form the names of output files.

outputFileFolder: a character string indicating the directory in which the output files will be saved.

fileSource: a character string, either TCGA-Assembler or Firehose. TCGA-Assembler, which is the default setting, indicates that the input data file was acquired by TCGA-Assembler Module A, and Firehose indicates that the input data file was downloaded from Firehose website.

Output argument

A list object formed by Des and Data. Des is a character matrix of four-column descriptions of CpG sites including Illumina ID of CpG site, gene symbol, chromosome ID, and genomic coordinate. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in Data is a sample and the column names of Data are TCGA sample barcodes.

Note

For HumanMethylation450 data file downloaded from Firehose, this function calls GetMethylation450Data function to read in the data file and get rid of the redundant columns of CpG site descriptions.

GetMethylation450Data function reads and processes the data file block by block to circumvent potential memory limitation problem caused by large sizes of Firehose data files (>10GB for some cancer types).

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download humanmethylation450 data of six rectum adenocarcinoma (READ) patient samples

Methylation450RawData = DownloadMethylationData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "humanmethylation450", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-A01W-11", "TCGA-AG-3731-11"));

# Process the downloaded humanmethylation450 data and save the results to
# ProcessedData.TCGA-Assembler subfolder.

Methylation450Data = ProcessMethylation450Data(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__jhu-usc.edu__humanmethylation450__Jan-30-2014.txt", outputFileName = "READ__humanmethylation450", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Process READ HumanMethylation450 data downloaded from Firehose website.

Methylation450Data = ProcessMethylation450Data(inputFilePath = "./UserManualExampleData/RawData.Firehose/READ.methylation__humanmethylation450__jhu_usc_edu__Level_3__within_bioassay_data_set_function__data.data.txt", outputFileName = "READ__humanmethylation450", outputFileFolder = "./UserManualExampleData/ProcessedData.Firehose", fileSource = "Firehose");
```

ProcessmiRNASeqData<-function(inputFilePath, outputFileName, outputFileFolder, fileSource = "TCGA-Assembler")

This function processes miRNASeq data file either acquired by TCGA-Assembler module A or downloaded from Firehose website, and imports the data into R. It does the following.

(1) Save the miRNASeq read count data and reads per million miRNA mapped (RPM) data as two separate tab-delimited .txt files. In each file, the first column is the miRNA name. The other columns are data of individual samples. The first row in the file gives the TCGA sample barcodes (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode). The file names are composed of outputFileName and either "ReadCount.txt" or "RPM.txt", with double-underscore "__" separating the two.

(2) Save the miRNASeq read count data and RPM data as two separate R data file (.rda), each of which contains two variables. The first variable is Des, which is a single-column character matrix including the miRNA names. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in Data is a sample and each row corresponds to a miRNA. The file names are composed of outputFileName and either "ReadCount.rda" or "RPM.rda", with double-underscore "__" separating the two parts.

For both read count and RPM, the .txt file and .rda file contain the same data. Having the same data in two different file formats is just for the convenience of using the data in different software environments.

Input arguments

inputFilePath: a character string indicating the path of the input miRNAseq data file acquired by TCGA-Assembler module A or downloaded from Firehose website.

outputFileName: a character string to form the names of output data files.

outputFileFolder: a character string indicating the directory to which the output files will be saved.

fileSource: a character string that can be either TCGA-Assembler or Firehose. TCGA-Assembler, which is the default setting, indicates that the input data file was acquired by TCGA-Assembler Module A and Firehose indicates that the input data file was downloaded from Firehose website.

Output argument

A list object of two variables Des and Data. Des is a single-column character matrix including miRNA names. It serves as the description of the second variable, Data, which is a numeric matrix of RPM values. Each column in Data corresponds to a sample and each row corresponds to a miRNA. And the column names are TCGA sample barcodes.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download miRNA-seq data of six rectum adenocarcinoma (READ) patient samples
```

```
miRNASeqRawData = DownloadmiRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "miRNASeq", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AF-2689-11", "TCGA-AF-2691-11"));

# Process the downloaded READ miRNA-seq data and save the results to
# the ProcessedData.TCGA-Assembler subfolder.

miRNASeqData = ProcessmiRNASeqData(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__bcgsc.ca__illuminahiseq_mirnaseq__NCBI36__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_mirnaseq", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Process READ miRNA-seq data downloaded from Firehose website,
# and save the results to the ProcessedData.Firehose subfolder.

miRNASeqData = ProcessmiRNASeqData(inputFilePath = "./UserManualExampleData/RawData.Firehose/READ.mirnaseq__illuminaga_mirnaseq__bcgsc_ca__Level_3__miR_gene_expression__data.data.txt", outputFileName = "READ__illuminaga_mirnaseq", outputFileFolder = "./UserManualExampleData/ProcessedData.Firehose", fileSource = "Firehose");
```

ProcessRNASeqData<-function(inputFilePath, outputFileName, outputFileFolder, dataType, verType)

This function processes RNASeq data files and imports data into R, including (1) gene expression data file generated by RNASeqV1 pipeline, (2) exon expression data file generated by RNASeqV1 pipeline, (3) normalized gene expression data file generated by RNASeqV2 pipeline, and (4) exon expression data file generated by RNASeqV2 pipeline. The input data files should be either acquired by TCGA-Assembler Module A or downloaded from Firehose website. This function does the following.

(1) For gene expression data, check whether the gene symbols are official HGNC gene symbols. If not, correct them.

(2) Extract most useful measurements from the input data files. For RNASeqV1 gene expression data and exon expression data, extract the RPKM values. For RNASeqV2 normalized gene expression data, extract the normalized count values. For RNASeqV2 exon expression data, extract the RPKM values.

(3) Draw and save a box plot of log2 transferred gene expression data for quality control purpose. The picture file name is composed of outputFileName and "__boxplot.png".

(4) Save the extracted data as a tab-delimited .txt file. For gene expression data, the first two columns are gene symbol and Entrez ID. And the other columns are samples, with TCGA sample barcodes in the top row (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode). For exon expression data, the first column is exon ID. And the other columns are samples, with TCGA sample barcodes in the top row.

(5) Save the extracted data as an R data file (.rda), which includes two variables Des and Data. Des is a character matrix including gene symbols and Entrez IDs for gene expression data, and exon IDs for exon expression data. Des describes the second variable, Data, which is a numeric matrix. Each column in Data is a sample and each row corresponds to a gene or exon. The column names of Data are TCGA sample barcodes.

Input arguments

inputFilePath: a character string indicating the path of input RNASeq data file acquired by TCGA-Assembler Module A or downloaded from Firehose website. It should be a tab-delimited .txt file.

outputFileName: a character string to form the names of output files.

outputFileFolder: a character string indicating the directory in which the output files will be saved.

dataType: a character string indicating the type of input data file. GeneExp indicates that the input file is gene expression data. ExonExp indicates that the input file is exon expression data.

verType: a character string indicating the data processing pipeline used to generate the data. Options include RNASeqV1 and RNASeqV2.

Output argument

A list object of two variables Des and Data. If dataType is GeneExp, Des is a character matrix of two columns including gene symbols and Entrez IDs. It describes the second variable, Data, which is a numeric matrix of gene expressions. Each column in the matrix is a sample and each row corresponds to a gene. The column names of Data are TCGA sample barcodes. If dataType is ExonExp, Des is a single-column character matrix of exon IDs. Data is a numeric matrix of exon expressions. Each column in the data matrix is a sample and each row corresponds to an exon.

Note

RNASeqV1 and RNASeqV2 are two different pipelines used to process RNA-seq read count data, and generate different types of data. For details of RNASeqV1 and RNASeqV2 pipelines and the data files

generated by them, please refer to https://tcga-data.nci.nih.gov/tcgafiles/ftp_auth/distro_ftpusers/anonymous/tumor/brca/cgcc/unc.edu/illuminahiseq_rnaseqv2/rnaseqv2/unc.edu_BRCA.IlluminaHiSeq_RNASeqV2.mage-tab.1.7.0/DESCRIPTION.txt

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions

source("Module_B.r");  # Load Module B functions

# Download normalized gene expression data and exon expression data of six rectum
# adenocarcinoma (READ) patient samples, which were generated by RNASeqV2 pipeline.

RNASeqRawData = DownloadRNASeqData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda", saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ", assayPlatform = "RNASeqV2", dataType = c("rsem.genes.normalized_results", "exon_quantification"), inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-6572-01", "TCGA-F5-6812-01", "TCGA-AG-3732-11", "TCGA-AG-3742-11"));

# Process the downloaded normalized gene expression data and save the results to the
# ProcessedData.TCGA-Assembler subfolder.

GeneExpData = ProcessRNASeqData(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__unc.edu__illuminahiseq_rnaseqv2__rsem.genes.normalized_results__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_rnaseqv2__GeneExp", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler", dataType = "GeneExp", verType = "RNASeqV2");

# Process the downloaded exon expression data and save the results to the
# ProcessedData.TCGA-Assembler subfolder.

ExonExpData = ProcessRNASeqData(inputFilePath = "./UserManualExampleData/RawData.TCGA-Assembler/READ__unc.edu__illuminahiseq_rnaseqv2__exon_quantification__Jan-30-2014.txt", outputFileName = "READ__illuminahiseq_rnaseqv2__ExonExp", outputFileFolder = "./UserManualExampleData/ProcessedData.TCGA-Assembler", dataType = "ExonExp", verType = "RNASeqV2");

# Process RNASeqV1 gene expression data of READ samples,
# which were downloaded from Firehose website.

GeneExpData = ProcessRNASeqData(inputFilePath = "./UserManualExampleData/RawData.Firehose/READ.rnaseq__illuminaga_rnaseq__unc_edu__Level_3__gene_expression__data.data.txt", outputFileName = "READ__illuminaga_rnaseq__GeneExp", outputFileFolder = "./UserManualExampleData/ProcessedData.Firehose", dataType = "GeneExp", verType = "RNASeqV1");
```

ProcessRPPADataWithGeneAnnotation<-function(inputFilePath, outputFileName, outputFileFolder)

This function processes RPPA data files either acquired by TCGA-Assembler Module A or downloaded from Firehose website, and imports the data into R. It does the following.

(1) Split the gene symbol and protein antibody name into two separate columns.

(2) Check whether the gene symbols are official HGNC gene symbols. If not, correct them.

(3) When a protein is encoded by more than one gene, duplicate the measurement of the protein (a row in the data matrix) for each gene.

(4) Draw and save a box plot picture of the data for quality control purpose. The picture file name is composed of outputFileName and "__boxplot.png".

(5) Save protein expression data as a tab-delimited .txt file. The first column gives gene symbols. The second column gives protein antibody names. The other columns are data of individual samples. The first row gives TCGA sample barcodes (refer to https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode).

(6) Save the processed data as an R data file (.rda), which includes two variables Des and Data. Des is a character matrix including two columns, i.e. gene symbol and protein antibody name. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in the data matrix is a sample and column names are TCGA sample barcodes.

The output .txt file and .rda file contain the same protein expression data. Having the same data in two different file formats is just for the convenience of using the data under different software environments.

Input arguments

inputFilePath: a character string indicating the path of the input RPPA protein expression data file acquired by TCGA-Assembler Module A or downloaded from Firehose website. It should be a tab-delimited .txt file.

outputFileName: a character string to form the names of output data files and box plot picture file.

outputFileFolder: a character string indicating the path of the directory to which the output files will be saved.

Output argument

A list object of two matrix variables. The first variable is Des, which is a character matrix including two columns, i.e. gene symbol and protein antibody name. It serves as the description of the second variable, Data, which is a numeric matrix. Each column in the data matrix is a sample and the column names are TCGA sample barcodes.

Note

Before applying the function, check the input data file to see whether gene symbols corresponding to the same protein antibody are separated by a single white space. Sometimes the gene symbols are not correctly separated due to errors inherited from the original TCGA data files.

Examples

```
# The present working directory of R must be TCGA-Assembler, i.e. the package folder,
# for running the examples.

rm(list = ls());     # Clear workspace

source("Module_A.r");  # Load Module A functions
```

```
source("Module_B.r");   # Load Module B functions

# Download RPPA protein expression data of six rectum adenocarcinoma (READ) patient samples
# and save the acquired data in ./UserManualExampleData/RawData.TCGA-Assembler

RPPARawData = DownloadRPPAData(traverseResultFile = "./DirectoryTraverseResult_Jan-30-2014.rda",
saveFolderName = "./UserManualExampleData/RawData.TCGA-Assembler", cancerType = "READ",
assayPlatform = "mda_rppa_core", inputPatientIDs = c("TCGA-EI-6884-01", "TCGA-DC-5869-01", "TCGA-G5-
6572-01", "TCGA-F5-6812-01", "TCGA-AG-3582-01", "TCGA-AG-4001-01"));

# Process the downloaded protein expression data and save the results in
# ./UserManualExampleData/ProcessedData.TCGA-Assembler

RPPAData = ProcessRPPADataWithGeneAnnotation(inputFilePath =
"./UserManualExampleData/RawData.TCGA-Assembler/READ__mdanderson.org__mda_rppa_core__Jan-
30-2014.txt", outputFileName = "READ__mda_rppa_core", outputFileFolder =
"./UserManualExampleData/ProcessedData.TCGA-Assembler");

# Process RPPA protein expression data file downloaded from Firehose website.

RPPAData = ProcessRPPADataWithGeneAnnotation(inputFilePath =
"./UserManualExampleData/RawData.Firehose/READ.RPPA_AnnotateWithGene.txt", outputFileName =
"READ__mda_rppa_core", outputFileFolder = "./UserManualExampleData/ProcessedData.Firehose");
```