

# **Tutorial of TCGA-Assembler 2**

Lin Wei<sup>1</sup>, Zhilin Jin<sup>2</sup>, Shengjie Yang<sup>1</sup>, Yanxun Xu<sup>2</sup>, Yitan Zhu<sup>1\*</sup>, Yuan Ji<sup>1,3\*</sup>

1. Program of Computational Genomics & Medicine, NorthShore University HealthSystem, Evanston, Illinois, USA

2. Department of Applied Mathematics & Statistics, Johns Hopkins University, Baltimore, Maryland, USA

3. Department of Public Health Sciences, The University of Chicago, Chicago, Illinois, USA

\*Correspondence author

Contact: [koaeraser@gmail.com](mailto:koaeraser@gmail.com), [zhuyitan@gmail.com](mailto:zhuyitan@gmail.com)

# Contents

1. System Requirements and Installation.....	3
2. Notice to Windows Users .....	3
3. Notice to Linux/Unix Users.....	4
4. Quick Start Examples.....	4
5. Part 1: Data Downloading.....	5
6. Part 2: Basic Data Processing.....	10
7. Part 3: Advanced Data Processing.....	15

This tutorial provides a quick start to use TCGA-Assembler 2. We will use examples in the script QuickStartExample.R to illustrate how to use TCGA-Assembler 2 functions to download, process, and combine The Cancer Genome Atlas (TCGA) and the Clinical Proteomic Tumor Analysis Consortium (CPTAC) data. Not all TCGA-Assembler 2 functions will be introduced here. For the full set of functions and their details, please see the provided user manual.

## 1 System Requirements and Installation

Downloading and processing TCGA data from Internet may need significant memory space depending on the size of data to be retrieved and processed, we recommend using TCGA-Assembler 2 on computers with 16GB or larger RAM and with a fast and stable Internet connection. However, all the examples in this guide should work well on computers with 4GB~8GB RAM, as we only use a small dataset to demo the functions.

TCGA-Assembler 2 is built using R (<http://www.r-project.org/>), so users need to have basic knowledge of R to use TCGA-Assembler 2. It requires R packages HGNChelper, httr, RCurl, rjson, stringr, and their dependents. We assume that users have a recent R version installed. To start, users should launch R and install the required packages, for example using the following command

```
packages <- c("HGNChelper", "httr", "RCurl", "rjson", "stringr")
install.packages(packages, dependencies = T)
```

Another way to install the R packages is that in R GUI (Graphical User Interface), go to Packages menu and click on Install package(s). Select the best CRAN mirror site for you. And then select the package and click ok to install.

**To download TCGA-Assembler 2**, go to <http://www.compgenome.org/> to register and download the package. The package is compressed as a zip file. Unzip the downloaded file to your desired file directory on the local computer. For example, in our own test, we unzip the package and create the folder /TCGA-Assembler/ for the unzipped files. Then, we set the Present Working Directory (PWD) of R using

```
setwd("/TCGA-Assembler/")
```

**Another way to obtain TCGA-Assembler 2** is to download the package from GitHub (<https://github.com/compgenome365/TCGA-Assembler-2>), which does not require registration.

## 2 Notice to Windows Users

Please make curl command available as a system command for TCGA-Assembler 2 to use. The easiest way to do so is to copy curl.exe in TCGA-Assembler 2 package to the Windows system directory C:\Windows\System32. You can also download latest curl executable file supporting SSL and SSH and compatible with your operating system from <https://curl.haxx.se/download.html>

Windows operating system usually has a limitation on the length of file path, which is 260

characters. TCGA data files usually have a long file name and folder name. So, the downloaded data files may have paths (including both the full directory and file name) longer than the limitation, causing failure when writing the data files to your local hard disk. If you see the following messages in your R console, it most likely indicates a failure caused by the limitation on file path length, so the program can not save the data files and keeps retrying.

```
[1] "metadata file: preparing ..."  
[1] "metadata file: preparing done!"  
[1] "*.tar.gz file: downloading & unzipping ..."  
[1] "cannot open the connection"  
[1] "cannot open the connection"  
[1] "cannot open the connection"  
...
```

To solve this problem, either put TCGA-Assembler 2 package in a directory with a short path (such as the root directory C:\) or change the setting of your operating system to allow long file path. The setting procedure is specific to your Windows version. Please Google on the Internet for solutions about configuring your Windows operating system to allow long file path.

### 3 Notice to Linux/Unix Users

Some users may encounter the following error message on certain operating systems, such as Ubuntu

```
Error in function (type, msg, asError = TRUE):  
gnutls_handshaked: A TLS fatal alert has been received.
```

This error is likely due to missing libcurl4-openssl-dev package that is used for network connection. Please try the following commands to install it.

```
$ sudo apt-get remove libcurl4-gnutls-dev  
$ sudo apt-get install libcurl4-openssl-dev
```

### 4 Quick Start Examples

Let us launch R and use QuickStartExample.R (included in the package) as an example to navigate through TCGA-Assembler 2 package. The code in QuickStartExample.R can be divided into three parts, which are data downloading, basic data processing, and advanced data processing. Part 1, data downloading, uses Module A functions to download TCGA/CPTAC data. Part 2, basic data processing, uses basic data processing functions in Module B to process the data downloaded in Part 1 for quality control and other purposes. Part 3, advanced data processing, uses advanced data processing functions in Module B to further process the data generated in Part 2 to fulfill various data manipulation needs. First of all, we need to set the working directory of R to the package folder using the following command.

```
setwd("/TCGA-Assembler/")
```

Then, load Module A and Module B functions into the working space.

```
source("./Module_A.R")
source("./Module_B.R")
```

Choose breast invasive carcinoma (BRCA) as the cancer type for demo.

```
sCancer <- "BRCA"
```

Choose several patients to form a small data set for the demo. The following are TCGA barcodes of patients.

```
vPatientID <- c("TCGA-A7-A13F", "TCGA-AO-A12B", "TCGA-AR-A1AP", "TCGA-AR-A1AQ", "TCGA-AR-A1AS", "TCGA-AR-A1AV", "TCGA-AR-A1AW", "TCGA-BH-A0BZ", "TCGA-BH-A0DD", "TCGA-BH-A0DG")
```

If no specific patient barcodes are given, the data downloading functions will acquire data of all samples of the cancer type. Set the directory for saving the result data files that are generated by each part of the code.

```
sPath1 <- "./QuickStartExample/Part1_DownloadedData"
sPath2 <- "./QuickStartExample/Part2_BasicDataProcessingResult"
sPath3 <- "./QuickStartExample/Part3_AdvancedDataProcessingResult"
```

## 5 Part 1: Data Downloading

In this part, we use Module A functions to download data of seven different platforms for the samples and save the data files. There are four input arguments taken by every data downloading function, including `cancerType` (specifying the cancer type of interest), `assayPlatform` (specifying the assay platform used to generate the data of interest), `inputPatientIDs` (specifying the barcodes of patients of interest), and `saveFolderName` (specifying the directory to save the downloaded data). The return value of each data downloading function is a character vector of paths for the saved data files. First, we download copy number alteration (CNA) data using the following command.

```
path_copyNumber <- DownloadCNADData(cancerType = sCancer, assayPlatform
= "cna_cnv.hg19", inputPatientIDs = vPatientID, saveFolderName = sPath1)
```

The downloaded data are saved as a tab-delimited .txt file. Its file path is `/QuickStartExample/Part1_DownloadedData/BRCA__cna_cnv.hg19__tissueTypeAll__***.txt`, where `***` is the year, date, and time when the data are downloaded. This file path is return to `path_copyNumber` by the function call. The following picture is an illustration of the data file.

Sample	Chromosome	Start	End	Num_Probes	Segment_Mean
TCGA-AO-A12B-01A-11D-A10L-01	22	25087674	51234455	18566	0.164
TCGA-AO-A12B-01A-11D-A10L-01	X	168477	16319658	10064	0.0094
TCGA-AO-A12B-01A-11D-A10L-01	X	16325524	16411642	56	0.5674
TCGA-AO-A12B-01A-11D-A10L-01	X	16413514	154392382	74244	0.0148
TCGA-AO-A12B-01A-11D-A10L-01	X	154395920	154417401	9	-0.8444
TCGA-AO-A12B-01A-11D-A10L-01	X	154418294	155182354	232	-0.0013
TCGA-AO-A12B-01A-11D-A10L-01	Y	2650438	2985595	152	-2.6868
TCGA-AO-A12B-01A-11D-A10L-01	Y	2985939	6101425	719	-2.2461
TCGA-AO-A12B-01A-11D-A10L-01	Y	6107901	6296423	85	-3.6996
TCGA-AO-A12B-01A-11D-A10L-01	Y	6299046	19571480	4705	-2.7818
TCGA-AO-A12B-01A-11D-A10L-01	Y	19574890	21028957	674	-3.3743
TCGA-AO-A12B-01A-11D-A10L-01	Y	21029230	24515802	1275	-2.8309
TCGA-AO-A12B-01A-11D-A10L-01	Y	24516081	28423938	839	-3.2945
TCGA-AO-A12B-01A-11D-A10L-01	Y	28438116	59018259	183	-2.5149
TCGA-AO-A12B-10A-01D-A10L-01	1	61735	1510801	226	-0.072
TCGA-AO-A12B-10A-01D-A10L-01	1	1627918	1672603	17	-0.8083

This command downloads the CNA data of the patients specified by `inputPatientIDs` if their data are available. In the user manual, we introduce the details about how to use the `DownloadCNADData` function, such as what options are available for `assayPlatform`, and the format information of the downloaded data file, e.g. what each row and each column in the above data table indicate.

Using the following command, we can download DNA methylation data generated by the Illumina Infinium HumanMethylation450 BeadChip.

```
path_methylation_450 <- DownloadMethylationData(cancerType = sCancer,
assayPlatform = "methylation_450", inputPatientIDs = vPatientID,
saveFolderName = sPath1)
```

The downloaded data are saved as a tab-delimited .txt file, whose path is `./QuickStartExample/Part1_DownloadedData/BRCA_methylation_450_tissueTypeAll_***.txt`, where `***` again is the year, date, and time when the data are downloaded. This file path is returned by the function call to `path_methylation_450`. The following picture is an illustration of the data file.

CpG	Gene_Symbol	Chromosome	Genomic_Coordinate	TCGA-A7-A13F-01A-11D-A12R-05	TCGA-A7-A13F-11A-42D-A12R-05
cg00000029	RBL2	16	53468112	0.116363206	0.130402792
cg00000165		1	91194674	0.428333848	0.166828533
cg00000236	VDAC3	8	42263294	0.918171197	0.906248705
cg00000289	ACTN1	14	69341139	0.619009848	0.639440629
cg00000292	ATP2A1	16	28890100	0.840011367	0.690831906
cg00000321	SFRP1	8	41167802	0.326545149	0.2647191
cg00000363		1	230560793	0.669940372	0.165739002
cg00000622	NIPA2	15	23034447	0.010371387	0.012556149
cg00000658	MAN1B1	9	139997924	0.883668613	0.877941951
cg00000714	TSEN34	19	54695678	0.088901683	0.098999075
cg00000721	LRRC16A	6	25282779	0.944658051	0.95901386
cg00000734	CNBP	3	128902377	0.043456285	0.041175528

This command downloads the DNA methylation data of the patients specified by inputPatientIDs if their data are available. For details about how to use the DownloadMethylationData function and the format of the downloaded data file, e.g. what each row and each column of the above data table indicate, please see the user manual.

The following command downloads miRNA expression data.

```
path_miRNAExp <- DownloadmiRNASeqData(cancerType = sCancer,
assayPlatform = "mir_HiSeq.hg19.mirbase20", inputPatientIDs =
vPatientID, saveFolderName = sPath1)
```

assayPlatform = "mir\_HiSeq.hg19.mirbase20" indicates the data to be downloaded are generated using RNA-seq and the miRNAs are defined using human reference genome 19 and miRBase (<http://www.mirbase.org/>) version 20. The downloaded data are saved to ./QuickStartExample/Part1\_DownloadedData/BRCA\_\_mir\_HiSeq.hg19.mirbase20\_\_tissueTypeAll\_\_\*\*\*.txt, indicated by path\_miRNAExp. The following picture is an illustration of the data file.

miRNA_ID	TCGA-A7-A13F-01A-11R-A12O-13	TCGA-A7-A13F-01A-11R-A12O-13	TCGA-A7-A13F-11A-42R-A12O-13	TCGA-A7-A13F-11A-42R-A12O-13
miRNA_ID	read_count	reads_per_million_miRNA_mapped	read_count	reads_per_million_miRNA_mapped
hsa-let-7a-1	39353	14353.0112	30522	13435.3094
hsa-let-7a-2	79715	29074.02963	61377	27017.20021
hsa-let-7a-3	39824	14524.79654	30661	13496.49503
hsa-let-7b	110108	40159.1075	64826	28535.39633
hsa-let-7c	3322	1211.615461	19379	8530.33421
hsa-let-7d	1611	587.571495	1899	835.910246
hsa-let-7e	3557	1297.325766	4553	2004.159743
hsa-let-7f-1	121	44.131689	75	33.013833
hsa-let-7f-2	33099	12072.02292	31026	13657.16235
hsa-let-7g	2369	864.032819	2540	1118.068471
hsa-let-7i	1419	517.544352	1536	676.123296

The following command downloads mRNA expression data.

```
path_geneExp <- DownloadRNASeqData(cancerType = sCancer, assayPlatform
= "gene.normalized_RNAseq", inputPatientIDs = vPatientID,
saveFolderName = sPath1)
```

assayPlatform = "gene.normalized\_RNAseq" indicates the data are normalized read counts of RNA-seq data. The downloaded data are saved to ./QuickStartExample/Part1\_DownloadedData/BRCA\_\_gene.normalized\_RNAseq\_\_tissueTypeAll\_\_\*\*\*.txt. The following picture is an illustration of the data file.

gene_id	TCGA-A7-A13F-01A-11R-A12P-07	TCGA-A7-A13F-11A-42R-A12P-07	TCGA-AO-A12B-01A-11R-A10J-07	TCGA-AR-A1AP-01A-11R-A12P-07
? 8225	391.5663	424.7867	543.8207	280.7018
? 90288	19.2771	13.9201	172.957	74.224
A1BG 1	231.008	67.5258	212.2042	152.0468
A1CF 29974	0	0	0	0
A2BP1 54715	0	0	0	0.8997
A2LD1 87769	45.249	112.7885	39.8055	82.0198
A2ML1 144568	0	9.8788	0.4229	1.7994
A2M 2	13194.3494	30585.357	5225.8928	11859.1813
A4GALT 53947	97.5904	469.2411	467.7027	413.8552
A4GNT 51146	0	0.449	0	0
AAA1 404744	0	0	0	0
AAAS 8086	912.8514	672.6538	481.6577	713.0004

The following command downloads protein expression data generated using the Reverse Phase Protein Array (RPPA).

```
path_protein_RPPA <- DownloadRPPADData(cancerType = sCancer,
assayPlatform = "protein_RPPA", inputPatientIDs = vPatientID,
saveFolderName = sPath1)
```

The downloaded data are saved to ./QuickStartExample/Part1\_DownloadedData/BRCA\_\_protein\_RPPA\_\_tissueTypeAll\_\_\*\*\*.txt. The following picture is an illustration of the data file.

protein	TCGA-A7-A13F-01A-21-A13E-20	TCGA-A7-A13F-11A-51-A43O-20	TCGA-AO-A12B-01A-21-A13E-20	TCGA-AR-A1AP-01A-21-A13E-20
YWHAB 14-3-3_beta-R-V	0.036468674	0.390213481	0.211709774	0.073389932
YWHAE 14-3-3_epsilon-M-C	-0.015836214	0.006679618	0.444883029	0.152640107
YWHAZ 14-3-3_zeta-R-V	0.101070959	0.091580791	-0.028849412	0.089381224
EIF4EBP1 4E-BP1-R-V	0.653155165	-0.061781101	0.321254692	-0.052588945
EIF4EBP1 4E-BP1_pS65-R-V	-0.117276295	-0.511525287	-0.239330676	-0.064308709
EIF4EBP1 4E-BP1_pT37_T46-R-V	-0.265351544	-0.154878295	-1.426977429	0.085029823
EIF4EBP1 4E-BP1_pT70-R-V	0.041832409	-0.07080207	-0.218871436	-0.169120181
TP53BP1 53BP1-R-E	0.50356765	-0.620714551	0.086369173	0.350886034
ARAF A-Raf-R-V	0.145087298	-0.400681235	-0.372555598	-0.01138981
ARAF A-Raf_pS299-R-C	-0.080834561	0.254919731	-0.062274342	-0.276284297
ACACA ACC1-R-E	0.991784532	0.917478696	0.337840911	-0.749329066
ACACA ACACB ACC_pS79-R-V	0.269004627	1.019226585	-1.162631322	-0.652842977

The following command downloads mass spectrometry proteomics data of the specified patients that are generated by the CPTAC using the isobaric Tags for Relative and Absolute Quantification (iTRAQ) assay.

```
path_protein_iTRAQ <- DownloadCPTACData(cancerType = sCancer,
assayPlatform = "proteome_iTRAQ", inputPatientIDs = vPatientID,
saveFolderName = sPath1)
```

proteome\_iTRAQ indicates that we download proteome data generated using the iTRAQ assay. The downloaded data are saved to ./QuickStartExample/Part1\_DownloadedData/BRCA\_\_proteome\_iTRAQ\_\_tissueTypeAll\_\_BI\_\_\*\*\*.txt. This file path is returned to path\_protein\_iTRAQ by the function call. The



following picture is an illustration of the data file.

Gene	Description	Chr	Locus	TCGA-AR-A1AS-01A-Log-Ratio	TCGA-AR-A1AS-01A-Unshared-Log-Ratio	TCGA-BH-A0DD-01A-Log-Ratio	TCGA-BH-A0DD-01A-Unshared-Log-Ratio
A1BG	alpha-1-B glycoprotein	19	19q13.4	-0.2947073	-0.2902306	-0.090228	-0.1068032
A2M	alpha-2-macroglobulin	12	12p13.31	-0.5380259	-0.5117868	-1.0016199	-0.9922238
A2ML1	alpha-2-macroglobulin-like 1	12	12p13.31	-0.7907405	-0.7911393	-0.9200288	-1.4137337
AAAS	achalasia, adrenocortical insufficiency, alacrimia	12	12q13	0.1598317	0.1643084	0.7936982	0.7771231
AACS	acetoacetyl-CoA synthetase	12	12q24.31	-0.1586298	-0.1541531	-0.4207125	-0.4372876
AADAT	aminoadipate aminotransferase	4	4q33	-1.5105599	-1.5060832	-0.0054269	-0.022002
AAED1	AhpC/TSA antioxidant enzyme domain containing 1	9	9q22.32	0.0888495	0.0933262	-1.7370351	-1.7536102
AAGAB	alpha- and gamma-adaptin binding protein	15	15q22.33-q23	0.3492869	0.3537636	0.6925054	0.6759303
AAK1	AP2 associated kinase 1	2	2p14	0.1696641	0.1611207	0.3175102	0.3055456
AAMDC	adipogenesis associated, Mth938 domain containing	11	11q14.1	-0.2888097	-0.284333	-0.4696047	-0.4861798
AAMP	angio-associated, migratory cell protein	2	2q35	-0.0704705	-0.0659938	-0.2155309	-0.232106
AAR2	AAR2 splicing factor homolog (S. cerevisiae)	20	20pter-q12	-0.2160668	-0.2115901	0.7437817	0.7272065

Lastly, we download the somatic mutation data of the patients using the following command.

```
path_somaticMutation <- DownloadSomaticMutationData(cancerType =
sCancer, assayPlatform = "somaticMutation_DNAseq", inputPatientIDs =
vPatientID, saveFolderName = sPath1)
```

This command downloads somatic mutation data of the specified patients and cancer type (i.e. BRCA). There are usually multiple files/versions of somatic mutation data for a cancer type. For each file/version, the `DownloadSomaticMutationData` function selects informative columns from the data table and save them to a tab-delimited .txt file on the local hard disk. The paths of the saved data files are returned to `path_somaticMutation`. The following picture is an illustration of one of the data files.

Hugo_Symbol	Chr	Start_Position	End_Position	Strand	Variant_Classification	Variant_Type	Ref_Allele	Tumor_Seq_Allele1	Tumor_Seq_Allele2	Tumor_Sample_Barcode	Matched_Norm_Sample_Barcode
AASDHPPT	11	105948464	105948464	+	Missense_Mutation	SNP	C	C	G	TCGA-AR-A1AQ-01A-11D-A12Q-09	TCGA-AR-A1AQ-10A-01D-A12Q-09
ABCA12	2	215843156	215843156	+	Frame_Shift_Del	DEL	T	T	-	TCGA-AR-A1AQ-01A-11D-A12Q-09	TCGA-AR-A1AQ-10A-01D-A12Q-09
ABC8	7	150741305	150741305	+	Missense_Mutation	SNP	G	G	T	TCGA-A7-A13F-01A-11D-A12Q-09	TCGA-A7-A13F-10A-01D-A12Q-09
ABC8	7	150741305	150741305	+	Missense_Mutation	SNP	G	G	T	TCGA-A7-A13F-01A-11D-A12Q-09	TCGA-A7-A13F-11A-42D-A12Q-09
ABCC3	17	48746861	48746861	+	Missense_Mutation	SNP	G	G	T	TCGA-BH-A0BZ-01A-31D-A12Q-09	TCGA-BH-A0BZ-11A-61D-A12Q-09
ABCC8	11	17498260	17498260	+	Missense_Mutation	SNP	G	G	A	TCGA-AR-A1AQ-01A-11D-A12Q-09	TCGA-AR-A1AQ-10A-01D-A12Q-09

Some of the original TCGA somatic mutation data files include Ctrl+Z in the text, which is a special character on Windows operating system that indicates the end of a file. An error may occur when reading these files on Windows system (not on other operating systems), because the reading process stops when encountering Ctrl+Z. In such a case, only a part of the original data are imported, processed, and saved to the output file. To cope with this situation, for Windows

system (not other operating systems), the original TCGA somatic mutation data files are also saved in a sub-folder named as "originalSomaticMutationFiles" in the directory indicated by `saveFolderName`. In our example, they are saved in `./QuickStartExample/Part1_DownloadedData/originalSomaticMutationFiles`

## 6 Part 2: Basic Data Processing

For each data platform, Module B provides a corresponding basic data processing function whose name starts with "Process". These functions perform quality control on data (such as validate/correct the gene symbols and draw box plot of data for identifying outliers), remove redundant genomic feature descriptions, and separate different types of measurements into their own data tables (such as separating the raw read counts and the reads per million miRNA mapped (RPM) of miRNA-seq data into two tables). These functions transform data into the matrix format and ensure that each row in the matrix corresponds to a unique genomic feature and each column corresponds to a sample. The copy number processing function also calculates gene-level copy number alternation (CNA), which is an average CNA of DNA fragments within a gene in a sample.

Several common input arguments of the basic processing functions include `inputFilePath` (specifying the input data file that is downloaded by Module A function), `outputFileName` (specifying the name of the file to store processed data), and `outputFileFolder` (the directory where the processed data file should be saved, which is specified by `sPath2` in the demo). The return value of a basic data processing function is usually a list of two matrices, which are `Des` (a character matrix for description of genomic features) and `Data` (a numeric matrix of data). `Des` matrix serves as the description of `Data` matrix, i.e. a row of `Des` provides the genomic feature description of the corresponding row in `Data`. `Des` and `Data` are also merged together and exported to the output data files saved in the directory indicated by `outputFileFolder` with the filename given by `outputFileName`. Two data files, including a `.rda` file (i.e. R data file format) and a `.txt` file, are outputted and saved containing the same processed data. Saving the same data in two different file formats facilitates using the data in different software environments.

We first process the CNA data downloaded in Part 1 of the demo using the following command.

```
list_copyNumber <- ProcessCNADData(inputFilePath = path_copyNumber[1],
outputFileName = paste(sCancer, "copyNumber", sep = "__"),
refGenomeFile = "./SupportingFiles/Hg19GenePosition.txt",
outputFileFolder = sPath2)
```

`refGenomeFile` indicates the path of a support file included in the package that provides the genomic coordinates of genes. The `ProcessCNADData` function uses the gene coordinates to calculate an average copy number of each gene in each sample and outputs the gene-level CNA data. The output data files are `BRCA__copyNumber.txt` and `BRCA__copyNumber.rda` in `./QuickStartExample/Part2_BasicDataProcessingResult/`. In the same folder, a boxplot of the data, whose filename is `BRCA__copyNumber__boxplot.png`, is also saved. The following picture illustrates the `BRCA__copyNumber.txt` file.

Gene Symbol	Chr	Strand	TCGA-A7- A13F-01A- 11D-A12N-01	TCGA-A7- A13F-10A- 01D-A12N-01	TCGA-A7- A13F-11A- 42D-A12N-01	TCGA-AO- A12B-01A- 11D-A10L-01	TCGA-AO- A12B-10A- 01D-A10L-01
OR4F5	CHR1	+	-0.0283	-0.048	0.0297	-0.1729	-0.072
LOC729737	CHR1	-	-0.0283	0.4706	0.0297	-0.1729	-0.072
LOC100132287	CHR1	+	-0.0283	0.4706	0.0297	-0.1729	-0.072
OR4F29	CHR1	+	-0.0283	0.4706	0.0297	-0.1729	-0.072
OR4F16	CHR1	-	-0.6362	0.4706	0.0297	-0.1729	-0.072
LOC100133331	CHR1	-	-0.6362	0.4706	0.0297	-0.1729	-0.072
LOC100288069	CHR1	-	-0.6362	0.4706	0.0297	-0.1729	-0.072
LINC00115	CHR1	-	-0.6362	0.4706	0.0297	-0.1729	-0.072
LOC643837	CHR1	+	-0.6362	0.4706	0.0297	-0.1729	-0.072
FAM41C	CHR1	-	-0.6362	0.4706	0.0297	-0.1729	-0.072
LOC100130417	CHR1	-	-0.6362	0.4706	0.0297	-0.1729	-0.072
SAMD11	CHR1	+	-0.6362	0.4706	0.0297	-0.1729	-0.072

For details about how to use the `ProcessCNData` function and the format information of the output data file, e.g. what each row and each column in the above data table indicate, please see the user manual.

We use the following command to process the DNA methylation data downloaded in Part 1 of this demo.

```
list_methylation_450 <- ProcessMethylation450Data(inputFilePath =
path_methylation_450[1], outputFileFolder = paste(sCancer,
"methylation_450", sep = "__"), outputFileName = sPath2)
```

The above command outputs three files in `./QuickStartExample/Part2_BasicDataProcessingResult/`, which are `BRCA__methylation_450.txt`, `BRCA__methylation_450.rda`, and `BRCA__methylation_450_boxplot.png`. The following picture illustrates the `BRCA__methylation_450.txt` file.

REF	Gene Symbol	Chr	Coordinate	TCGA-A7- A13F-01A- 11D-A12R-05	TCGA-A7- A13F-11A- 42D-A12R-05	TCGA-AO- A12B-01A- 11D-A10N-05	TCGA-AR- A1AP-01A- 11D-A12R-05
cg13869341	WASH5P	1	15865	0.886879799	0.892384394	0.881069106	0.800113151
cg14008030	WASH5P	1	18827	0.803315101	0.668286673	0.792004138	0.667987594
cg12045430	WASH5P	1	29407	0.055560655	0.03014451	0.026930373	0.111632628
cg20826792	WASH5P	1	29425	0.175869891	0.144824814	0.134526675	0.191239781
cg00381604	WASH5P	1	29435	0.01632793	0.020460064	0.016672168	0.022166467
cg20253340	OR4F5	1	68849	0.307208579	0.457821082	0.844804344	0.517295056
cg21870274	OR4F5	1	69591	0.405556818	0.839738895	0.908156718	0.81109047
cg03130891		1	91550	0.337245083		0.352986103	0.119438408
cg24335620		1	135252	0.827559416	0.803281362	0.757140384	0.696226956
cg17149495		1	530959	0.689006496	0.936729387	0.907259466	0.884542837
cg22802167		1	533950	0.860614094	0.962215629	0.941343783	0.94646847
cg17308840		1	542758	0.489858806	0.385950041	0.126352575	0.620673121

Again, for details about how to use the `ProcessMethylation450Data` function and the format information of the output data file, e.g. what each row and each column in the above data table indicate, please see the user manual.

We then process the miRNA expression data downloaded in Part 1 of the demo using the following command.

```
list_miRNAExp <- ProcessmiRNASeqData(inputFilePath = path_miRNAExp[1],
outputFileName = paste(sCancer, "miRNAExp", sep = "__"),
outputFileFolder = sPath2)
```

Because miRNA expression data include two kinds of measurements, i.e. raw read counts and RPMs, this function call generates a .rda data file and a .txt data file for raw read counts and another set of .rda file and .txt file for RPMs. The following picture is an illustration of the .txt RPM data file.

GeneSymbol	TCGA-A7-A13F-01A-11R-A12O-13	TCGA-A7-A13F-11A-42R-A12O-13	TCGA-AO-A12B-01A-11R-A10I-13	TCGA-AR-A1AP-01A-11R-A12O-13	TCGA-AR-A1AQ-01A-11R-A12O-13	TCGA-AR-A1AS-01A-11R-A12O-13
hsa-let-7a-1	14353.0112	13435.3094	18237.04197	5863.07824	4944.16177	12548.33172
hsa-let-7a-2	29074.02963	27017.20021	36414.33107	11577.05145	9987.644152	24958.29718
hsa-let-7a-3	14524.79654	13496.49503	18091.92785	5868.237584	5057.150355	12476.11666
hsa-let-7b	40159.1075	28535.39633	48523.98767	32370.75457	18872.73841	42980.64085
hsa-let-7c	1211.615461	8530.33421	360.069251	2639.004329	2486.660057	739.79094
hsa-let-7d	587.571495	835.910246	225.819293	345.676031	1172.712163	539.683555
hsa-let-7e	1297.325766	2004.159743	588.99259	1128.348479	868.371944	1915.628552
hsa-let-7f-1	44.131689	33.013833	34.144498	18.573638	17.312767	22.050401
hsa-let-7f-2	12072.02292	13657.16235	4980.440635	4769.297364	6482.264437	9746.828326
hsa-let-7g	864.032819	1118.068471	688.322038	591.776728	986.827718	943.205886
hsa-let-7i	517.544352	676.123296	265.395871	626.860266	944.0014	401.317291
hsa-mir-1-1	0	0	0	0	0	0

We use the following command to process the gene expression data downloaded in Part 1 of the demo.

```
list_geneExp <- ProcessRNASeqData(inputFilePath = path_geneExp[1],
outputFileName = paste(sCancer, "geneExp", sep = "__"), dataType =
"geneExp", outputFileFolder = sPath2)
```

The output data files of this function call are BRCA\_\_geneExp.rda and BRCA\_\_geneExp.txt in ./QuickStartExample/Part2\_BasicDataProcessingResult/. BRCA\_\_geneExp\_\_boxplot.png is a boxplot of the data saved in the same folder. The following picture is an illustration of the BRCA\_\_geneExp.txt file.

GeneSymbol	EntrezID	TCGA-A7-A13F-01A-11R-A12P-07	TCGA-A7-A13F-11A-42R-A12P-07	TCGA-AO-A12B-01A-11R-A10J-07	TCGA-AR-A1AP-01A-11R-A12P-07	TCGA-AR-A1AQ-01A-11R-A12P-07	TCGA-AR-A1AS-01A-11R-A12P-07
?	8225	391.5663	424.7867	543.8207	280.7018	347.1074	371.0041
?	90288	19.2771	13.9201	172.957	74.224	4.1322	169.2931
A1BG	1	231.008	67.5258	212.2042	152.0468	72.9184	71.4768
A1CF	29974	0	0	0	0	0.5165	0
RBFOX1	54715	0	0	0	0.8997	0	0
GGACT	87769	45.249	112.7885	39.8055	82.0198	47.1384	93.5254
A2ML1	144568	0	9.8788	0.4229	1.7994	1134.2975	4.9527
A2M	2	13194.3494	30585.357	5225.8928	11859.1813	12341.4101	4301.6614
A4GALT	53947	97.5904	469.2411	467.7027	413.8552	112.0868	137.3255
A4GNT	51146	0	0.449	0	0	0.5165	0
NPSR1-AS1	404744	0	0	0	0	0	0
AAAS	8086	912.8514	672.6538	481.6577	713.0004	626.0331	778.9284
AACSP1	729522	0	1.7961	0	1.3495	0	0

We use the following command to process the RPPA protein data downloaded in Part 1 of the demo.

```
list_protein_RPPA <- ProcessRPPADDataWithGeneAnnotation(inputFilePath =
path_protein_RPPA[1], outputFileName = paste(sCancer, "protein_RPPA",
sep = "__"), outputFolder = sPath2)
```

This command generates BRCA\_protein\_RPPA.rda, BRCA\_protein\_RPPA.txt, and BRCA\_protein\_RPPA\_boxplot.png in ./QuickStartExample/Part2\_BasicDataProcessingResult/. The following picture is an illustration of the BRCA\_protein\_RPPA.txt file.

Gene Symbol	Protein Antibody	TCGA-A7-A13F-01A-21-A13E-20	TCGA-A7-A13F-11A-51-A43O-20	TCGA-AO-A12B-01A-21-A13E-20	TCGA-AR-A1AP-01A-21-A13E-20	TCGA-AR-A1AQ-01A-21-A13E-20	TCGA-AR-A1AS-01A-21-A13E-20
YWHAB	14-3-3_beta-R-V	0.036468674	0.390213481	0.211709774	0.073389932	-0.23396641	-0.13818132
YWHAE	14-3-3_epsilon-M-C	-0.01583621	0.006679618	0.444883029	0.152640107	-0.10132455	-0.00172421
YWHAZ	14-3-3_zeta-R-V	0.101070959	0.091580791	-0.02884941	0.089381224	-0.1616772	-0.49634248
EIF4EBP1	4E-BP1-R-V	0.653155165	-0.0617811	0.321254692	-0.05258895	0.58246435	-0.1618214
EIF4EBP1	4E-BP1_pS65-R-V	-0.11727629	-0.51152529	-0.23933068	-0.06430871	0.032928173	-0.24237861
EIF4EBP1	4E-BP1_pT37_T46-R-V	-0.26535154	-0.1548783	-1.42697743	0.085029823	0.430937445	-0.15678691
EIF4EBP1	4E-BP1_pT70-R-V	0.041832409	-0.07080207	-0.21887144	-0.16912018	0.177298443	0.174726896
TP53BP1	53BP1-R-E	0.50356765	-0.62071455	0.086369173	0.350886034	0.009330409	0.683512207
ARAF	A-Raf-R-V	0.145087298	-0.40068124	-0.3725556	-0.01138981	-0.14748498	-0.01856132
ARAF	A-Raf_pS299-R-C	-0.08083456	0.254919731	-0.06227434	-0.2762843	-0.24957733	0.074887456
ACACA	ACC1-R-E	0.991784532	0.917478696	0.337840911	-0.74932907	-0.41307134	0.421179662
ACACA	ACC_pS79-R-V	0.269004627	1.019226585	-1.16263132	-0.65284298	-0.18060298	0.673584129

The following command processes the iTRAQ protein expression data downloaded in Part 1 of the demo.

```
list_protein_iTRAQ <- ProcessCPTACData(inputFilePath =
path_protein_iTRAQ[1], outputFileName = paste(sCancer, "protein_iTRAQ",
sep = "__"), outputFolder = sPath2)
```

The iTRAQ protein expressions include two types of data, which are measurements based on all peptides and measurements based on only unshared peptides that are uniquely mapped to a protein. For each type of data, a .rad file and a .txt file are generated to store the data. Data files of measurements based on all peptides are BRCA\_protein\_iTRAQ\_allPeptides.rda and BRCA\_protein\_iTRAQ\_allPeptides.txt, while data files of measurements based on only



unshared peptides are BRCA\_\_protein\_iTRAQ\_unsharedPeptides.rda and BRCA\_\_protein\_iTRAQ\_unsharedPeptides.txt. Their boxplots are also saved. This function call returns a list of two elements, named AllPeptidesData and UnsharedPeptidesData. Each of them is a list object of two matrices, i.e. the numeric data matrix Data and its genomic feature description matrix Des. AllPeptidesData is the data counting all peptides and UnsharedPeptidesData is the data considering only peptides uniquely mapped to one protein. The following picture illustrates the BRCA\_\_protein\_iTRAQ\_allPeptides.txt file.

Gene Symbol	Description	Chr	Locus	TCGA-AR-A1AS-01A	TCGA-BH-A0DD-01A	TCGA-BH-A0DG-01A
A1BG	alpha-1-B glycoprotein	19	19q13.4	-0.2947073	-0.090228	-0.1771301
A2M	alpha-2-macroglobulin	12	12p13.31	-0.5380259	-1.0016199	-0.3247701
A2ML1	alpha-2-macroglobulin-like 1	12	12p13.31	-0.7907405	-0.9200288	-0.0731056
AAAS	achalasia, adrenocortical insufficiency, alacrimia	12	12q13	0.15983171	0.79369822	0.02173431
AACS	acetoacetyl-CoA synthetase	12	12q24.31	-0.1586298	-0.4207125	-0.1893885
AADAT	aminoadipate aminotransferase	4	4q33	-1.5105599	-0.0054269	
AAED1	AhpC/TSA antioxidant enzyme domain containing 1	9	9q22.32	0.08884948	-1.7370351	-0.2900469
AAGAB	alpha- and gamma-adaptin binding protein	15	15q22.33-q2	0.3492869	0.69250542	0.25698443
AAK1	AP2 associated kinase 1	2	2p14	0.16966414	0.31751023	0.19355384
AAMDC	adipogenesis associated, Mth938 domain containing	11	11q14.1	-0.2888097	-0.4696047	1.25835521
AAMP	angio-associated, migratory cell protein	2	2q35	-0.0704705	-0.2155309	0.23531298
AAR2	AAR2 splicing factor homolog (S. cerevisiae)	20	20pter-q12	-0.2160668	0.74378166	-0.1029208

We use the following command to process somatic mutation data downloaded in Part 1 of the demo.

```
list_somaticMutation <- ProcessSomaticMutationData(inputFilePath =
path_somaticMutation[1], outputFileName = paste(sCancer,
"somaticMutation", sep = "__"), outputFileFolder = sPath2)
```

This function call first transfers the somatic mutation data into the matrix format, where each row is a somatic mutation and each column corresponds to a tumor sample, which is called individual mutation-level data labeled by "mutationLevel" in their filenames. Then, it further transforms the data into summary of number of mutations occurring for each gene in each sample, no matter what somatic mutations they are and where they occur in the gene, with the filenames labeled by "geneLevel". Four data files are generated by this command, which are BRCA\_\_somaticMutation\_geneLevel.txt, BRCA\_\_somaticMutation\_geneLevel.rda, BRCA\_\_somaticMutation\_mutationLevel.txt, and BRCA\_\_somaticMutation\_mutationLevel.rda, all saved in ./QuickStartExample/Part2\_BasicDataProcessingResult/. The following picture is an illustration of BRCA\_\_somaticMutation\_geneLevel.txt.

Gene Symbol	Variant Classification	TCGA-AR- A1AQ-01A- 11D-A12Q-09/ TCGA-AR- A1AQ-10A- 01D-A12Q-09	TCGA-A7- A13F-01A-11D- A12Q-09/ TCGA-A7- A13F-10A-01D- A12Q-09	TCGA-A7- A13F-01A-11D- A12Q-09/ TCGA-A7- A13F-11A-42D- A12Q-09	TCGA-BH- A0BZ-01A-31D- A12Q-09/ TCGA-BH- A0BZ-11A-61D- A12Q-09	TCGA-AO- A12B-01A-11D- A10M-09/ TCGA-AO- A12B-10A-01D- A10M-09	TCGA-BH- A0DG-01A- 21D-A12Q-09/ TCGA-BH- A0DG-11A- 43D-A12Q-09
AASDHPPT	Missense_Mutation	1	0	0	0	0	0
ABCA12	Frame_Shift_Del	1	0	0	0	0	0
ABCB8	Missense_Mutation	0	1	1	0	0	0
ABCC3	Missense_Mutation	0	0	0	1	0	0
ABCC8	Missense_Mutation	1	0	0	0	0	0
ADAMTS6	Missense_Mutation	0	0	0	0	1	0
ADGRG7	Missense_Mutation	0	0	0	0	0	0
AHDC1	Frame_Shift_Ins	0	0	0	1	0	0
AIDA	Missense_Mutation	0	0	0	0	0	1
ALDH1L2	Missense_Mutation	0	0	0	1	0	0
ALDOA	Missense_Mutation	0	0	0	1	0	0
ALYREF	In_Frame_Ins	0	0	0	1	0	0

This command returns a list of two matrices representing the gene-level mutation data, including a numeric matrix `Data` and its genomic feature description matrix `Des`. Each row of `Des` includes the gene symbol and the classification information of somatic mutations in the gene across samples. Each element of `Data` is the total number of somatic mutations in a gene for a tumor sample.

## 7 Part 3: Advanced Data Processing

In this part, we will demo two of the advanced data processing functions `CalculateSingleValueMethylationData` and `CombineMultiPlatformData`. `CalculateSingleValueMethylationData` calculates average DNA methylation levels of CpG sites in particular regions of genes. For example, we can use the following command to calculate average DNA methylation levels in promoter regions of genes.

```
list_methylation_450_OverallAverage <-
CalculateSingleValueMethylationData(input = list_methylation_450,
regionOption = c("TSS1500", "TSS200"), DHSOption = "Both",
outputFileName = paste(sCancer, "methylation_450", sep = "__"),
outputFileFolder = sPath3)
```

Let us look at the input arguments of this function. `input` is the DNA methylation data generated by the basic data processing function in Part 2 of the demo. `regionOption` specifies for what genomic region the average DNA methylation value should be calculated. `TSS200` is with 200 base pairs (bps) ahead of the Transcription Start Site (TSS). `TSS1500` is with 1500 bps ahead of the TSS, but not including the 200 bps ahead of the TSS. `regionOption = c("TSS1500", "TSS200")` indicates the calculation is based on CpG sites within 1500 bps ahead of the TSS. `DHSOption = "Both"` indicates that the calculation is based on both CpG sites that are DNase hypersensitive and CpG sites that are not. `outputFileName` forms a part of the output file names. `outputFileFolder = sPath3` indicates the directory to save the output files, which is `./QuickStartExample/Part3_AdvancedDataProcessingResult/`. A `.rda` file (`BRCA__methylation_450__All__Both.rda`) and a `.txt` file (`BRCA__methylation_450__All__Both.txt`) containing the same output data are saved with a boxplot picture of the data (`BRCA__methylation_450__All__Both__boxplot.png`). For details about how to use the function and the data format information of the output files, please see the user manual. The following picture is an illustration of the

BRCA\_\_methylation\_450\_\_All\_\_Both.txt file.

Gene Symbol	Single Value Type	TCGA-A7-A13F-01A-11D-A12R-05	TCGA-A7-A13F-11A-42D-A12R-05	TCGA-AO-A12B-01A-11D-A10N-05	TCGA-AR-A1AP-01A-11D-A12R-05	TCGA-AR-A1AQ-01A-11D-A12R-05
WASH5P	TSS200-TSS1500 Both	0.082586159	0.065143129	0.059376405	0.108346292	0.091649417
OR4F5	TSS200-TSS1500 Both	0.307208579	0.457821082	0.844804344	0.517295056	0.443467191
MIR1977	TSS200-TSS1500 Both	0.246066762	0.230759862	0.292331299	0.225536604	0.302423444
LOC643837	TSS200-TSS1500 Both	0.027867415	0.01846254	0.020563189	0.02067391	0.017966856
LINC00115	TSS200-TSS1500 Both	0.027867415	0.01846254	0.020563189	0.02067391	0.017966856
FAM41C	TSS200-TSS1500 Both	0.665004981	0.6472216	0.756839562	0.757721883	0.655978581
FLJ39609	TSS200-TSS1500 Both	0.591407856	0.510484476	0.465551279	0.499879601	0.487835268
SAMD11	TSS200-TSS1500 Both	0.060422222	0.054630144	0.081233286	0.058586646	0.070809586
KLHL17	TSS200-TSS1500 Both	0.590826736	0.49669863	0.538430477	0.519952344	0.542681214
NOC2L	TSS200-TSS1500 Both	0.401008197	0.277318958	0.311197665	0.317572022	0.358960801
PLEKHN1	TSS200-TSS1500 Both	0.469582666	0.501508696	0.554389091	0.443429759	0.453118586
PERM1	TSS200-TSS1500 Both	0.388331615	0.594596072	0.387907755	0.464605805	0.444242646

This function call returns a list of two matrices, including a numeric matrix `Data` and its genomic feature description matrix `Des`. Each row of `Data` includes the average methylation values for a gene and each column is a sample. The corresponding row of `Des` gives the gene symbol and indicates how the average methylation value is calculated.

Now, it is ready to combine all data from seven different platforms into a mega data table. First, we organize the data of each platform into a list object that includes three elements, i.e. `Des`, `Data`, and `dataType`. `Des` is the description of genomic features. `Data` is the data. `dataType` indicates the data platform.

```
l_somaticMutation <- list(Des = list_somaticMutation$Des, Data =
list_somaticMutation$Data, dataType = "somaticMutation")

l_copyNumber <- list(Des = list_copyNumber$Des, Data =
list_copyNumber$Data, dataType = "copyNumber")

l_methylation <- list(Des = list_methylation_450_OverallAverage$Des,
Data = list_methylation_450_OverallAverage$Data, dataType =
"methylation")

l_miRNAExp <- list(Des = list_miRNAExp$Des, Data = list_miRNAExp$Data,
dataType = "miRNAExp")

l_geneExp <- list(Des = list_geneExp$Des, Data = list_geneExp$Data,
dataType = "geneExp")

l_protein_RPPA <- list(Des = list_protein_RPPA$Des, Data =
list_protein_RPPA$Data, dataType = "protein_RPPA")

l_protein_iTRAQ <- list(Des = list_protein_iTRAQ$AllPeptidesData$Des,
Data = list_protein_iTRAQ$AllPeptidesData$Data, dataType =
"protein_iTRAQ")
```

Second, the seven list objects are organized into a vector of list.

```
inputDataList <- vector("list", 7)
inputDataList[[1]] <- l_somaticMutation
inputDataList[[2]] <- l_copyNumber
```



```
inputDataList[[3]] <- l_methylation
inputDataList[[4]] <- l_miRNAExp
inputDataList[[5]] <- l_geneExp
inputDataList[[6]] <- l_protein_RPPA
inputDataList[[7]] <- l_protein_iTRAQ
```

Third, input the list vector into the `CombineMultiPlatformData` function to merge the multiplatform data.

```
list_CombinedData <- CombineMultiPlatformData(inputDataList =
inputDataList)
```

This command identifies the samples that are covered by all seven platforms and merges their data. With a different option, it can also use the union approach to merge data, which includes a sample in the output data as long as it is covered by at least one platform. It returns a list object of two variables `Des` and `Data`. `Des` is a character matrix of genomic feature descriptions and `Data` is the numeric data.

Fourth, write the merged multiplatform data to a tab-delimited .txt file.

```
write.table(cbind(list_CombinedData$Des, list_CombinedData$Data), file
= paste(sPath3, "CombinedMultiPlatformData.txt", sep = "/"), quote =
FALSE, sep = "\t", na = "", col.names = TRUE, row.names = FALSE)
```

GeneSymbol	Platform	Description	TCGA-A7-A13F-01	TCGA-AO-A12B-01
A1BG	copyNumber	CHR19-	0.1049	0.1516
A1BG	geneExp	1	231.008	212.2042
A1BG	methylation	TSS200-TSS1500 Both	0.896628452	0.896430547
A1BG	protein_iTRAQ	alpha-1-B glycoprotein	0.038663917	-0.402473958
A1BG-AS1	copyNumber	CHR19+	0.1049	0.1516
A1BG-AS1	geneExp	503538	131.8916	151.4029
A1BG-AS1	methylation	TSS200-TSS1500 Both	0.896628452	0.896430547
A1CF	copyNumber	CHR10-	-0.132	-0.1463
A1CF	geneExp	29974	0	0
A1CF	methylation	TSS200-TSS1500 Both	0.454185354	
A2M	copyNumber	CHR12-	0.1419	-0.1786
A2M	geneExp	2	13194.3494	5225.8928
A2M	methylation	TSS200-TSS1500 Both	0.293855362	0.248844969
A2M	protein_iTRAQ	alpha-2-macroglobulin	-0.991478486	-0.43370134
A2M-AS1	copyNumber	CHR12+	0.1419	-0.1786

The above picture illustrates the merged multiplatform data, i.e. the `CombinedMultiPlatformData.txt` file

in `./QuickStartExample/Part3_AdvancedDataProcessingResult/`. The first three columns are descriptions of genomic features, and the other two columns are data with TCGA patient barcodes being the column header. Rows in the table are ordered so that the multiplatform data of the same gene are adjacent. The first column in the table is gene symbol. The second column is data platform, which can be "methylation", "copyNumber", "somaticMatation", "geneExp", "miRNAExp", "protein\_RPPA" and "protein\_iTRAQ" representing DNA methylation, gene copy number, somatic mutation, gene expression, miRNA expression, protein expression produced by RPPA (from GDC) and protein expression generated by iTRAQ (from CPTAC), respectively.

The third column is additional description of genomic features depending on data platform. If the platform is "geneExp", the description is Entrez ID of the gene. If the platform is "protein\_RPPA", the description is the name of the protein antibody used in the RPPA assay. For "copyNumber" platform, the description shows the chromosome ID and strand of gene. For "miRNAExp" platform, the description column is empty. For "methylation" 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 are calculated (refer to the introduction of the `CalculateSingleValueMethylationData` function in the user manual 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. For the "somaticMutation" platform, the description is the classification information of the mutations in this gene. For the "protein\_iTRAQ" platform, the description is the full protein name. Other columns in the table are data.