Statistical Genetics: TCGA Software Packages

Buffalo Wild Wings

March 16, 2016

RTCGAToolbox

- blurbs all taken from: `http://omictools.com/tcga-data-access-category`
- "An open source and extensible R based data client for pre-processed data from the Firehose, and demonstrate its use with sample case studies. Results show that our RTCGAToolbox can facilitate data management for researchers interested in working with TCGA data. The RTCGAToolbox can also be integrated with other analysis pipelines for further data processing."

TCGA-assembler

- "An open-source, freely available tool that automatically downloads, assembles and processes public The Cancer Genome Atlas (TCGA) data, to facilitate downstream data analysis by relieving investigators from the burdens of data preparation. TCGA-Assembler includes two modules. Module A acquires public TCGA data from TCGA Data Coordinating Center and assembles individual data files into locally stored data tables. Module B does various manipulations on the data tables to prepare them for downstream analysis."

TCGAbiolinks

- (I think this one looks promising)
- "Aids in querying, downloading, analyzing and integrating TCGA data within a single collective Bioconductor package. TCGAbiolinks can: i) facilitate the TCGA open-access data retrieval, ii) prepare the data using the appropriate pre-processing strategies, iii) provide the means to carry out different standard analyses and iv) allow the user to download a specific version of the data and thus to easily reproduce earlier research results. In more detail, the package provides multiple methods for analysis (e.g., differential expression analysis, identifying differentially methylated regions) and methods for visualization (e.g., survival plots, volcano plots, starburst plots) in order to easily develop complete analysis pipelines."

TCGA2STAT

- (This is the one I chose to look into in more detail.)
- "An open source software package to obtain the TCGA data, wrangle it, and pre-process it into a format ready for multivariate and integrated statistical analysis in the R environment. In a user-friendly format with one single function call, our package downloads and fully processes the desired TCGA data to be seamlessly integrated into a computational analysis pipeline. **No further technical or biological knowledge is needed to utilize our software, thus making TCGA data easily accessible to data scientists without specific domain knowledge.**"

TCGA2STAT: Helpful Links:

- CRAN:
  `https://cran.r-project.org/web/packages/TCGA2STAT/index.html`
- User Guide:
  `http://www.liuzlab.org/TCGA2STAT/`
- Grid of cancers and available data (with cancer acronyms):
  `http://www.liuzlab.org/TCGA2STAT/CancerDataChecklist.pdf`
- Grid of data.type / type:
  `http://www.liuzlab.org/TCGA2STAT/DataPlatforms.pdf`
- Grid of common clinical variables:
  `http://www.liuzlab.org/TCGA2STAT/ClinicalVariables.pdf`

EXAMPLE: PROSTATE ADENOCARCINOMA

- The package can be installed from CRAN; it also depends on a package CNTools which needs to be downloaded from Bioconductor. The command for loading data is very straightforward, and the command produces a list of three matrices.

```
# source("https://bioconductor.org/biocLite.R")
# biocLite("CNTools")
# install.packages("TCGA2STAT")
library("TCGA2STAT")

## load in data -- specify the disease, the desired data type,
## and the "measurements" desired (typically there is a non-normalized
## and a normalized version for each data type)
pr.rna <- getTCGA(disease="PRAD", data.type="RNASeq2", type="RPKM")


## RNASeqV2 data will be imported! This may take some time!

## 20501 genes have been imported!
```

```
names(pr.rna)


## [1] "dat"        "clinical"   "merged.dat"
```

EXAMPLE: PROSTATE ADENOCARCINOMA

- The default behavior of the command does not load in any clinical data, so the clinical data set is NULL as is the merged data set.

```
class(pr.rna$dat)
```

```
## [1] "matrix"
```

```
pr.rna$dat[1:2,1:2]
```

```
##      TCGA-2A-A8VL-01A-21R-A37L-07
## A1BG                      67.9197
## A1CF                       0.0000
##      TCGA-2A-A8VO-01A-11R-A37L-07
## A1BG                      17.9448
## A1CF                       0.0000
```

```
pr.rna$clinical
```

```
## NULL
```

```
pr.rna$merged.dat
```

```
## NULL
```

EXAMPLE: PROSTATE ADENOCARCINOMA

- You can request all clinical data. But note that the merged.dat seems to only allow one clinical variable to be merged in. The default variable to load is overall survival.

```
## load in data requesting clinical variables
pr.rna <- getTCGA(disease="PRAD", data.type="RNASeq2", type="RPKM", clinical=TR
```

```
## RNASeqV2 data will be imported! This may take some time!
```

```
## 20501 genes have been imported!
```

```
## Clinical data will be imported.
```

```
## some variables are available for all diseases, some disease-specific
colnames(pr.rna$clinical)[1:10]
```

```
##  [1] "Composite Element REF"
##  [2] "yearstobirth"
##  [3] "vitalstatus"
##  [4] "daystodeath"
##  [5] "daystolastfollowup"
##  [6] "tumortissuesite"
##  [7] "pathologicstage"
##  [8] "pathologyTstage"
##  [9] "pathologyNstage"
## [10] "pathologyMstage"
```

EXAMPLE: PROSTATE ADENOCARCINOMA

- You can see here the default variables merged into merged.dat

```
pr.rna$merged.dat[1:10,1:10]
```

```
##              bcr status   OS    A1BG   A1CF
## 1  TCGA-2A-A8VL      0  621 67.9197 0.0000
## 2  TCGA-2A-A8VO      0 1701 17.9448 0.0000
## 3  TCGA-2A-A8VT      0 1373 17.1029 0.3676
## 4  TCGA-2A-A8VV      0  671 17.6115 0.0000
## 5  TCGA-2A-A8VX      0 1378 43.5168 0.0000
## 6  TCGA-2A-A8W1      0  112  7.5646 0.0000
## 7  TCGA-2A-A8W3      0  863 27.3482 0.0000
## 8  TCGA-2A-AAYF      0 1364 13.0900 0.0000
## 9  TCGA-2A-AAYO      0 1272 35.1337 0.0000
## 10 TCGA-2A-AAYU      0  615 16.5171 0.0000
##       A2BP1    A2LD1   A2ML1        A2M
## 1  25.2317 136.0453 45.3141  4827.853
## 2  17.9448 169.1662  2.6585 20000.691
## 3   1.8382 159.0037  1.1029  4906.496
## 4   3.7651 175.2959  0.9413  8427.490
## 5   0.5038 263.1062  2.5192  8330.823
## 6   0.0000 198.5503  4.7220  1511.217
## 7   2.4535 193.6618  2.0446 10273.506
## 8   0.4244 366.9228 20.3735  7582.029
## 9   4.1951 300.0577 23.0729 15474.851
## 10  2.5391 225.0148  3.8087  9980.643
```

EXAMPLE: PROSTATE ADENOCARCINOMA

- You can request certain clinical variables to be merged instead – but it doesn't seem like you can merge all of them in with these commands.

```
pr.rna <- getTCGA(disease="PRAD", data.type="RNASeq2", type="RPKM", clinical=TR
```

```
## RNASeqV2 data will be imported! This may take some time!

## 20501 genes have been imported!

## Clinical data will be imported.
```

```
data <- data.frame(pr.rna$merged.dat)
data[1:10,1:10]
```

```
##              bcr GLEASONSCORE    A1BG   A1CF
## 1  TCGA-2A-A8VL             6 67.9197 0.0000
## 2  TCGA-2A-A8VO             6 17.9448 0.0000
## 3  TCGA-2A-A8VT             9 17.1029 0.3676
## 4  TCGA-2A-A8VV             6 17.6115 0.0000
## 5  TCGA-2A-A8VX             8 43.5168 0.0000
## 6  TCGA-2A-A8W1             7  7.5646 0.0000
## 7  TCGA-2A-A8W3             9 27.3482 0.0000
## 8  TCGA-2A-AAYF             7 13.0900 0.0000
## 9  TCGA-2A-AAYO             6 35.1337 0.0000
## 10 TCGA-2A-AAYU             6 16.5171 0.0000
##      A2BP1    A2LD1   A2ML1       A2M
## 1  25.2317 136.0453 45.3141  4827.853
```

Available Diseases, Data Types, and Global Clinical Variables

- Grid of cancers and available data (with cancer acronyms): `http://www.liuzlab.org/TCGA2STAT/CancerDataChecklist.pdf`
- Grid of data.type / type: `http://www.liuzlab.org/TCGA2STAT/DataPlatforms.pdf`
- Grid of common clinical variables: `http://www.liuzlab.org/TCGA2STAT/ClinicalVariables.pdf`

Some Other Examples (from the Tutorial)

- They note that the Mutation Data is saved as lists of mutations available for each patient, so they aggregate the data into a matrix of dimension PXN, with a 1 in the $(i,j)^{th}$ entry of the matrix if a mutation is found in gene i from patient j, and a 0 otherwise.

```
mut.pr <- getTCGA(disease="PRAD", data.type="Mutation", type="somatic")
```

```
## Mutation data will be imported! This may take some time!
```

```
head(mut.pr$dat[,1:6])
```

```
##         TCGA-G9-6353 TCGA-CH-5772
## HIVEP3             1            0
## KIRREL             1            0
## CAMKMT             1            0
## CLIC1              1            0
## PPP3CC             1            0
## MGAT5B             1            0
##         TCGA-HC-A8CY TCGA-KK-A8IG
## HIVEP3             0            0
## KIRREL             0            0
## CAMKMT             0            0
## CLIC1              0            0
## PPP3CC             0            0
## MGAT5B             0            0
##         TCGA-HC-7749 TCGA-KK-A7B4
```

Some Other Examples (from the Tutorial)

- Methylation data is provided at the probe level, and the probes are annotated with Gene name (because for methylation data, genes have multiple CpG sites with different amounts of methylation).

```
methyl.ov <- getTCGA(disease="OV", data.type="Methylation", type="27K")
```

```
## Methylation data will be imported! This may take some time!

## 27578 CPG probes have been imported!
```

```
head(methyl.ov$dat[,1:3])
```

```
##            TCGA-01-0628-11A-01D-0383-05
## cg00000292                   0.79940858
## cg00002426                   0.33900444
## cg00003994                   0.02811930
## cg00005847                   0.60116497
## cg00006414                           NA
## cg00007981                   0.01881682
##            TCGA-01-0630-11A-01D-0383-05
## cg00000292                   0.62039417
## cg00002426                   0.18030460
## cg00003994                   0.03607298
## cg00005847                   0.64955777
## cg00006414                           NA
## cg00007981                   0.01803597
```

Some Other Examples (from the Tutorial)

```
head(methyl.ov$cpgs)
```

```
##              Gene_Symbol Chromosome
## cg00000292         ATP2A1         16
## cg00002426          SLMAP          3
## cg00003994          MEOX2          7
## cg00005847          HOXD3          2
## cg00006414 ZNF425;ZNF398          7
## cg00007981          PANX1         11
##            Genomic_Coordinate
## cg00000292           28890100
## cg00002426           57743543
## cg00003994           15725862
## cg00005847          177029073
## cg00006414          148822837
## cg00007981           93862594
```

Combining OMICS Data

- They provide a function for binding together different OMICS data

```
# Get the RNA-Seq, methylation, and mutation profiles for OV cancer patients
seq <- getTCGA(disease="OV", data.type="RNASeq2")
```

```
## RNASeqV2 data will be imported! This may take some time!

## 20501 genes have been imported!
```

```
meth <- getTCGA(disease="OV", data.type="Methylation", type="27K")
```

```
## Methylation data will be imported! This may take some time!

## 27578 CPG probes have been imported!
```

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

```
## Mutation data will be imported! This may take some time!
```

```
# Now, merge the three OMICs-data into one R object
# step 1: merge RNA-Seq and mutation data
m1 <- OMICSBind(dat1 = seq$dat, dat2 = mut$dat)
# step 2: further concatenate the methylation data to the merged data-object
m2 <- OMICSBind(dat1 = m1$merged.data, dat2 = meth$dat)
```

Splitting Tumor and Normal Data

- "The TCGA OMICs profiles downloaded for a cancer type often include samples of different types, such as from tumor tissue, normal tissue, or other controls. The type of sample is encoded in the sample's BCR code. Users who are interested in studying only tumor samples, for example, would need to parse the codes to extract only these samples. To spare users the difficulties of decoding the sample ID (BCR code) to extract particular types of samples, our package includes a function, SampleSplit to accomplish this task in a user-friendly manner. Specifically, SampleSplit will take the omics data matrix as input and separate the matrix according to the sample types. The object returned is a list of three matrices corresponding to the omics profiles of primary solid tumor, recurrent solid tumor, and normal tissues/blood samples."

Splitting Tumor and Normal Data

```
pr.2 <- SampleSplit(pr.rna$dat)
names(pr.2)
```

```
## [1] "primary.tumor"   "recurrent.tumor"
## [3] "normal"
```

Splitting Tumor and Normal Data

```
pr.2$normal[1:10,1:2]
```

```
##        TCGA-CH-5761-11A-01R-1580-07
## A1BG                        47.4871
## A1CF                         0.0000
## A2BP1                       11.9061
## A2LD1                      113.0807
## A2ML1                      160.2748
## A2M                       5594.0057
## A4GALT                     922.7247
## A4GNT                        0.0000
## AAA1                         0.0000
## AAAS                       836.6342
##        TCGA-CH-5767-11B-01R-1789-07
## A1BG                        36.3934
## A1CF                         0.0000
## A2BP1                       37.0455
## A2LD1                      102.9231
## A2ML1                      159.6002
## A2M                       9124.2072
## A4GALT                     586.1311
## A4GNT                        0.2537
## AAA1                         0.0000
## AAAS                       709.1933
```

Matching Tumor and Normal

- "Our package also includes a utility function, TumorNormalMatch to prepare the downloaded omics-profiles for pairwise analysis between tumor tissues and normal tissues/blood samples in a user-friendly manner. In the example below, we show how to first get the RNA-Seq data of analysis pipeline 2 from LUSC patients and then split the data into tumor tissues and normal tissues/blood samples for the same set of patients."

Splitting Tumor and Normal Data

```
pr.tum.norm <- TumorNormalMatch(pr.rna$dat)
names(pr.tum.norm)
```

```
## [1] "primary.tumor" "normal"
```

```
pr.tum.norm$primary.tumor[1:2,1:2]
```

```
##      TCGA-CH-5761 TCGA-CH-5767
## A1BG      12.2164      25.9445
## A1CF       0.0000       0.0000
```

```
pr.tum.norm$normal[1:2,1:2]
```

```
##      TCGA-CH-5761 TCGA-CH-5767
## A1BG      47.4871      36.3934
## A1CF       0.0000       0.0000
```