TCGA analysis

Paulyna Magana

Contents

0.1 TCGA Analysis

```
library(DT) #we will use it to visualize the results
```

```
## Warning: package 'DT' was built under R version 4.1.3
```

```
library(TCGAbiolinks)
library("limma")
library("edgeR")
library("caret")
```

```
## Warning: package 'caret' was built under R version 4.1.3

## Loading required package: ggplot2

## Warning: package 'ggplot2' was built under R version 4.1.3

## Loading required package: lattice
```

```
library("SummarizedExperiment")
```

```
## Loading required package: MatrixGenerics

## Loading required package: matrixStats

## Warning: package 'matrixStats' was built under R version 4.1.3

##
## Attaching package: 'MatrixGenerics'
```

```
## The following objects are masked from 'package:matrixStats':
##
##     colAlls, colAnyNAs, colAnys, colAvgsPerRowSet, colCollapse,
##     colCounts, colCummaxs, colCummins, colCumprods, colCumsums,
##     colDiffs, colIQRDiffs, colIQRs, colLogSumExps, colMadDiffs,
##     colMads, colMaxs, colMeans2, colMedians, colMins, colOrderStats,
##     colProds, colQuantiles, colRanges, colRanks, colSdDiffs, colSds,
##     colSums2, colTabulates, colVarDiffs, colVars, colWeightedMads,
##     colWeightedMeans, colWeightedMedians, colWeightedSds,
##     colWeightedVars, rowAlls, rowAnyNAs, rowAnys, rowAvgsPerColSet,
##     rowCollapse, rowCounts, rowCummaxs, rowCummins, rowCumprods,
##     rowCumsums, rowDiffs, rowIQRDiffs, rowIQRs, rowLogSumExps,
##     rowMadDiffs, rowMads, rowMaxs, rowMeans2, rowMedians, rowMins,
##     rowOrderStats, rowProds, rowQuantiles, rowRanges, rowRanks,
##     rowSdDiffs, rowSds, rowSums2, rowTabulates, rowVarDiffs, rowVars,
##     rowWeightedMads, rowWeightedMeans, rowWeightedMedians,
##     rowWeightedSds, rowWeightedVars

## Loading required package: GenomicRanges

## Loading required package: stats4

## Loading required package: BiocGenerics

## Loading required package: parallel

##
## Attaching package: 'BiocGenerics'

## The following objects are masked from 'package:parallel':
##
##     clusterApply, clusterApplyLB, clusterCall, clusterEvalQ,
##     clusterExport, clusterMap, parApply, parCapply, parLapply,
##     parLapplyLB, parRapply, parSapply, parSapplyLB

## The following object is masked from 'package:limma':
##
##     plotMA

## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs

## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min
```

```
## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following object is masked from 'package:grDevices':
##
##     windows

## Loading required package: GenomeInfoDb

## Loading required package: Biobase

## Welcome to Bioconductor
##
##     Vignettes contain introductory material; view with
##     'browseVignettes()'. To cite Bioconductor, see
##     'citation("Biobase")', and for packages 'citation("pkgname")'.

##
## Attaching package: 'Biobase'

## The following object is masked from 'package:MatrixGenerics':
##
##     rowMedians

## The following objects are masked from 'package:matrixStats':
##
##     anyMissing, rowMedians
```

```
library("gplots")
```

```
## Warning: package 'gplots' was built under R version 4.1.3

##
## Attaching package: 'gplots'

## The following object is masked from 'package:IRanges':
##
##     space
```

```
## The following object is masked from 'package:S4Vectors':
##
##     space

## The following object is masked from 'package:stats':
##
##     lowess
```

```
library("survival")
```

```
##
## Attaching package: 'survival'

## The following object is masked from 'package:caret':
##
##     cluster
```

```
library("survminer")
```

```
## Warning: package 'survminer' was built under R version 4.1.3

## Loading required package: ggpubr

## Warning: package 'ggpubr' was built under R version 4.1.2
```

```
library("RColorBrewer")
```

```
## Warning: package 'RColorBrewer' was built under R version 4.1.3
```

```
library("genefilter")
```

```
##
## Attaching package: 'genefilter'

## The following objects are masked from 'package:MatrixGenerics':
##
##     rowSds, rowVars

## The following objects are masked from 'package:matrixStats':
##
##     rowSds, rowVars
```

To download TCGA data with TCGAbiolinks, you need to follow 3 steps. First, you will query the TCGA database through R with the function GDCquery. This will allow you to investigate the data available at the TCGA database. Next, we use GDCdownload to download raw version of desired files into your computer. Finally GDCprepare will read these files and make R data structures so that we can further analyse them.

Before we get there however we need to know what we are searching for. We can check all the available projects at TCGA with the command bellow. Since there are many lets look at the first 6 projects using the command head().

```
GDCprojects = getGDCprojects()

head(GDCprojects[c("project_id", "name")])
```

```
##   project_id
## 1 TARGET-NBL
## 2 GENIE-GRCC
## 3 GENIE-DFCI
## 4  GENIE-NKI
## 5 GENIE-VICC
## 6  GENIE-UHN
##                                                                    name
## 1                                                          Neuroblastoma
## 2         AACR Project GENIE - Contributed by Institut Gustave Roussy
## 3    AACR Project GENIE - Contributed by Dana-Farber Cancer Institute
## 4    AACR Project GENIE - Contributed by Netherlands Cancer Institute
## 5 AACR Project GENIE - Contributed by Vanderbilt-Ingram Cancer Center
## 6 AACR Project GENIE - Contributed by Princess Margaret Cancer Centre
```

```
TCGAbiolinks:::getProjectSummary("TCGA-LUSC")
```

```
## $file_count
## [1] 23893
##
## $data_categories
##   file_count case_count               data_category
## 1       3146        504       Copy Number Variation
## 2       3350        504            Sequencing Reads
## 3       7791        497 Simple Nucleotide Variation
## 4       1719        503             DNA Methylation
## 5        577        504                    Clinical
## 6       2148        504     Transcriptome Profiling
## 7       2630        504                 Biospecimen
## 8        328        328          Proteome Profiling
## 9       2204        501        Structural Variation
##
## $case_count
## [1] 504
##
## $file_size
## [1] 3.568685e+13
```

Of note, not all patients were measured for all data types. Also, some data types have more files than samples. This is the case when more experiments were performed per patient, i.e. transcriptome profiling was performed both in mRNA and miRNA, or that data have been analysed by distinct computational strategies.

Let us start by querying all RNA-seq data from LIHC project. When using GDCquery we always need to specify the id of the project, i.e. "TCGA_LIHC", and the data category we are interested in, i.e. "Transcriptome Profiling". Here, we will focus on a particular type of data summarization for mRNA-seq data (workflow.type), which is based on raw counts estimated with HTSeq.

Note that performing this query will take a few of minutes

```
library(maftools)
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 4.1.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following object is masked from 'package:Biobase':
##
##     combine
```

```
## The following objects are masked from 'package:GenomicRanges':
##
##     intersect, setdiff, union
```

```
## The following object is masked from 'package:GenomeInfoDb':
##
##     intersect
```

```
## The following objects are masked from 'package:IRanges':
##
##     collapse, desc, intersect, setdiff, slice, union
```

```
## The following objects are masked from 'package:S4Vectors':
##
##     first, intersect, rename, setdiff, setequal, union
```

```
## The following objects are masked from 'package:BiocGenerics':
##
##     combine, intersect, setdiff, union
```

```
## The following object is masked from 'package:matrixStats':
##
##     count
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(TCGAWorkflowData)

# recovering data from TCGAWorkflowData package.
data(mafMutect2LGGGBM)

# To prepare for maftools we will also include clinical data
```

```r
# For a mutant vs WT survival analysis
# get indexed clinical patient data for GBM samples
gbm_clin <- GDCquery_clinic(project = "TCGA-GBM", type = "Clinical")
# get indexed clinical patient data for LGG samples
lgg_clin <- GDCquery_clinic(project = "TCGA-LGG", type = "Clinical")
# Bind the results, as the columns might not be the same,
# we will will plyr rbind.fill, to have all columns from both files
clinical <- plyr::rbind.fill(gbm_clin,lgg_clin)
colnames(clinical)[1] <- "Tumor_Sample_Barcode"

# we need to create a binary variable 1 is dead 0 is not dead
plyr::count(clinical$vital_status)
```

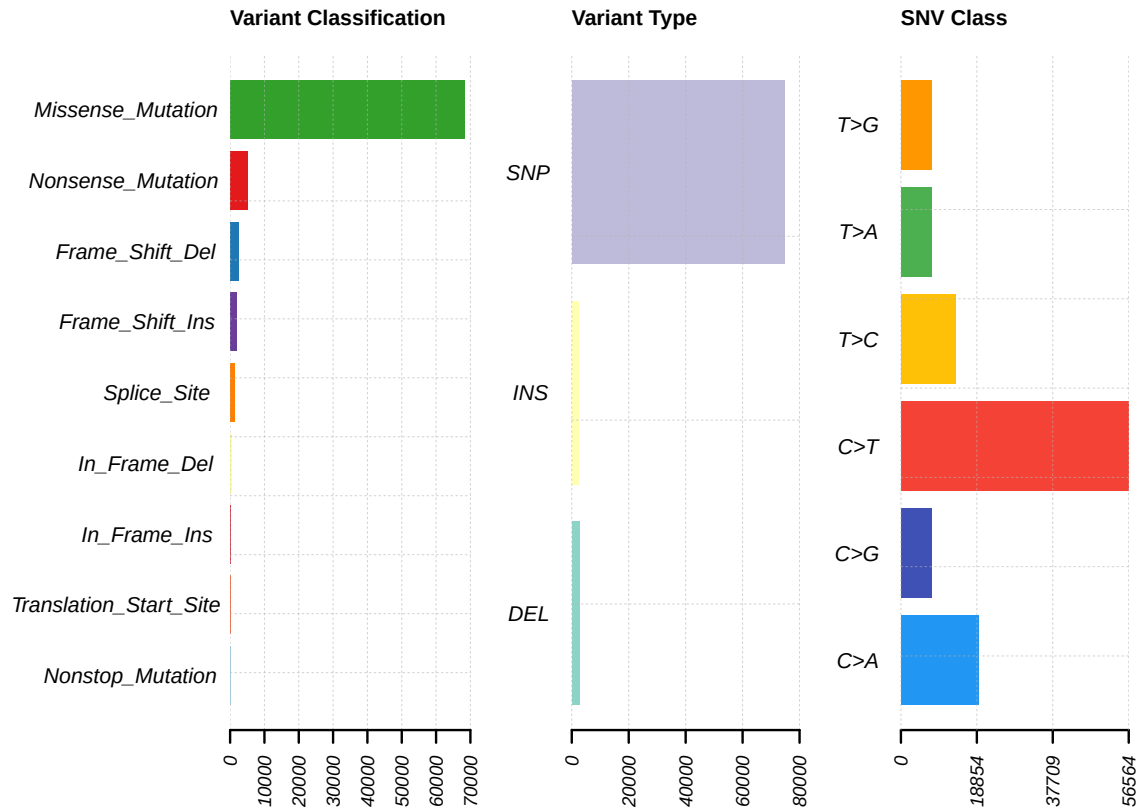
```
##              x freq
## 1        Alive  489
## 2         Dead  618
## 3 Not Reported    7
## 4         <NA>   19
```

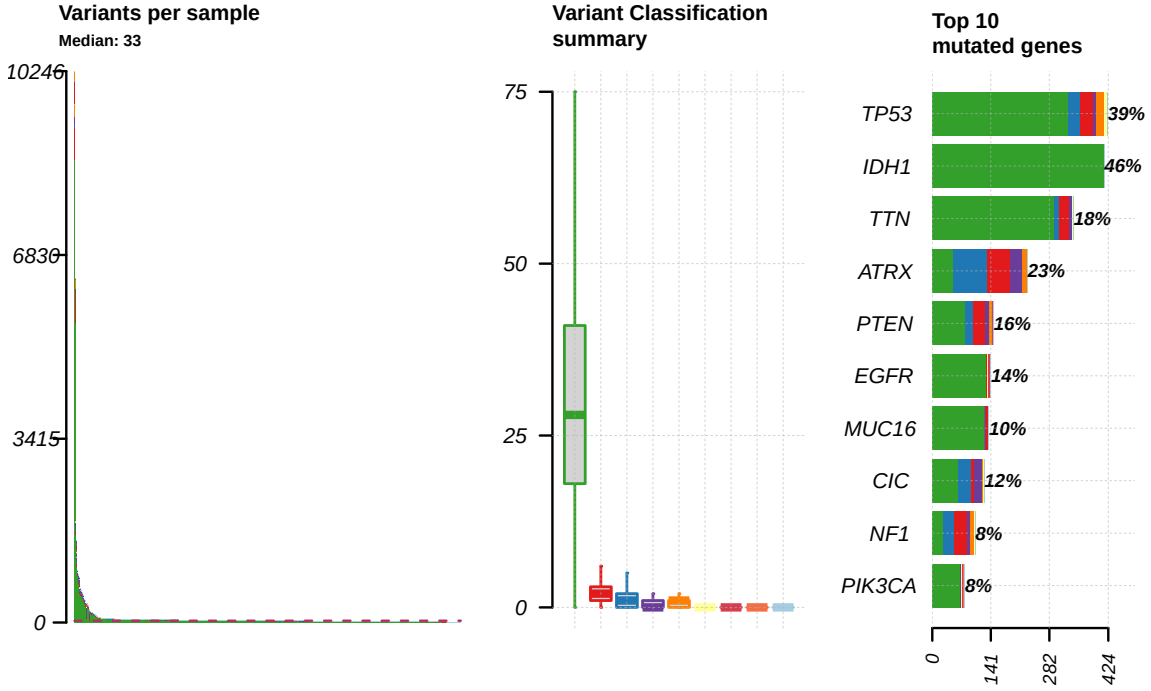
```r
clinical$Overall_Survival_Status <- 1 # dead
clinical$Overall_Survival_Status[which(clinical$vital_status != "Dead")] <- 0

# If patient is not dead we don't have days_to_death (NA)
# we will set it as the last day we know the patient is still alive
clinical$time <- clinical$days_to_death
clinical$time[is.na(clinical$days_to_death)] <- clinical$days_to_last_follow_up[is.na(clinical$days_to_
```

```r
# Create object to use in maftools
maf <- read.maf(maf = mut, clinicalData = clinical, isTCGA = TRUE)
```

```
## -Validating
## -Silent variants: 38433
## -Summarizing
## --Possible FLAGS among top ten genes:
##   TTN
##   MUC16
## -Processing clinical data
## -Finished in 13.8s elapsed (12.2s cpu)
```

```r
plotmafSummary(
  maf = maf,
  rmOutlier = TRUE,
  addStat = 'median',
  dashboard = TRUE
)
```


Variant Classification
Missense_Mutation
Nonsense_Mutation
Frame_Shift_Del
Frame_Shift_Ins
Splice_Site
In_Frame_Del
In_Frame_Ins
Translation_Start_Site
Nonstop_Mutation
0
10000
20000
30000
40000
50000
60000
70000
Variant Type
SNP
INS
DEL
0
20000
40000
60000
80000
SNV Class
T>G
T>A
T>C
C>T
C>G
C>A
0
18854
37709
56564


Variants per sample
Median: 33
10246
6830
3415
0
Variant Classification summary
75
50
25
0
Top 10 mutated genes
TP53
IDH1
TTN
ATRX
PTEN
EGFR
MUC16
CIC
NF1
PIK3CA
39%
46%
18%
23%
16%
14%
10%
12%
8%
8%
0
141
282
424