‘Bioinformatics tools to integrate and understand molecular changes associated with Immune Response, Stemness and Oncogenic processes: A PanCancer study.’

# Workshop: overview

In this workshop we will show the capability of TCGAbiolinks and Moonlight, to integrate multi -omics data from different consortium and to reproduce the six immune subtypes from TCGA PanCancer and how features (Immune Subtypes, Oncogenic Processes, Driver Genes and Stemness) can be used by the end user to expand their understating of their own un-published data.

The workshop is organized in 4 subsections:

1. Data retrieval from (TCGA, GTEx, GEO and IHEC)
2. Immune Subtypes
3. Oncognic Processes and Driver Genes
4. Stemness scores

# Data retrieval from (GDC, TCGA, GEO and IHEC)

# GDC and TCGA using TCGAbiolinks

You can easily query - download - prepare multi -omics data from GDC: . Gene expression . Copy number . Protein expression (RRPA) . Methylation . Clinical data . microRNA

## GDC TCGA Gene Expression data (IlluminaHiSeq\_RNASeqV2) using TCGAbiolinks

You can easily search TCGA samples, download and prepare a matrix of gene expression.

#DataDir <- "~/Dropbox (Personal)/Umiami/TCGAanalysis/GDCdata/TCGA-BRCA"  
  
# Firstly we install TCGAbiolinks from bioconductor or from github  
  
if (!requireNamespace("BiocManager", quietly = TRUE))  
 install.packages("BiocManager")  
BiocManager::install("TCGAbiolinks")  
  
require(TCGAbiolinks)  
  
  
cancerType <- "BRCA"  
  
# Query platform Illumina HiSeq with a list of barcode   
query <- GDCquery(project = paste0("TCGA-",cancerType),  
 data.category = "Gene expression",  
 data.type = "Gene expression quantification",  
 experimental.strategy = "RNA-Seq",  
 platform = "Illumina HiSeq",  
 file.type = "results",  
 legacy = TRUE)  
  
# We select 10 tumor and 10 normal samples  
Sample\_sel <- query$results[[1]]$cases  
  
Sample\_sel\_TP <- TCGAquery\_SampleTypes(barcode = Sample\_sel,typesample = "TP")  
Sample\_sel\_NT <- TCGAquery\_SampleTypes(barcode = Sample\_sel,typesample = "NT")   
  
Sample\_sel\_short <- c(Sample\_sel\_TP[1:10],Sample\_sel\_NT[1:10])  
  
# we need to create a new query with the selected barcodes  
query\_down <- GDCquery(project = paste0("TCGA-",cancerType),  
 data.category = "Gene expression",  
 data.type = "Gene expression quantification",  
 experimental.strategy = "RNA-Seq",  
 platform = "Illumina HiSeq",  
 file.type = "results",  
 barcode = Sample\_sel\_short,  
 legacy = TRUE)  
  
  
# Download a list of barcodes with platform IlluminaHiSeq\_RNASeqV2  
GDCdownload(query\_down,directory = DataDir)  
  
# Prepare expression matrix with geneID in the rows and samples (barcode) in the columns  
# rsem.genes.results as values  
BRCARnaseqSE <- GDCprepare(query\_down,directory = DataDir)  
#   
# For gene expression if you need to see a boxplot correlation and AAIC plot to define outliers you can run  
dataPrep <- TCGAanalyze\_Preprocessing(BRCARnaseqSE)

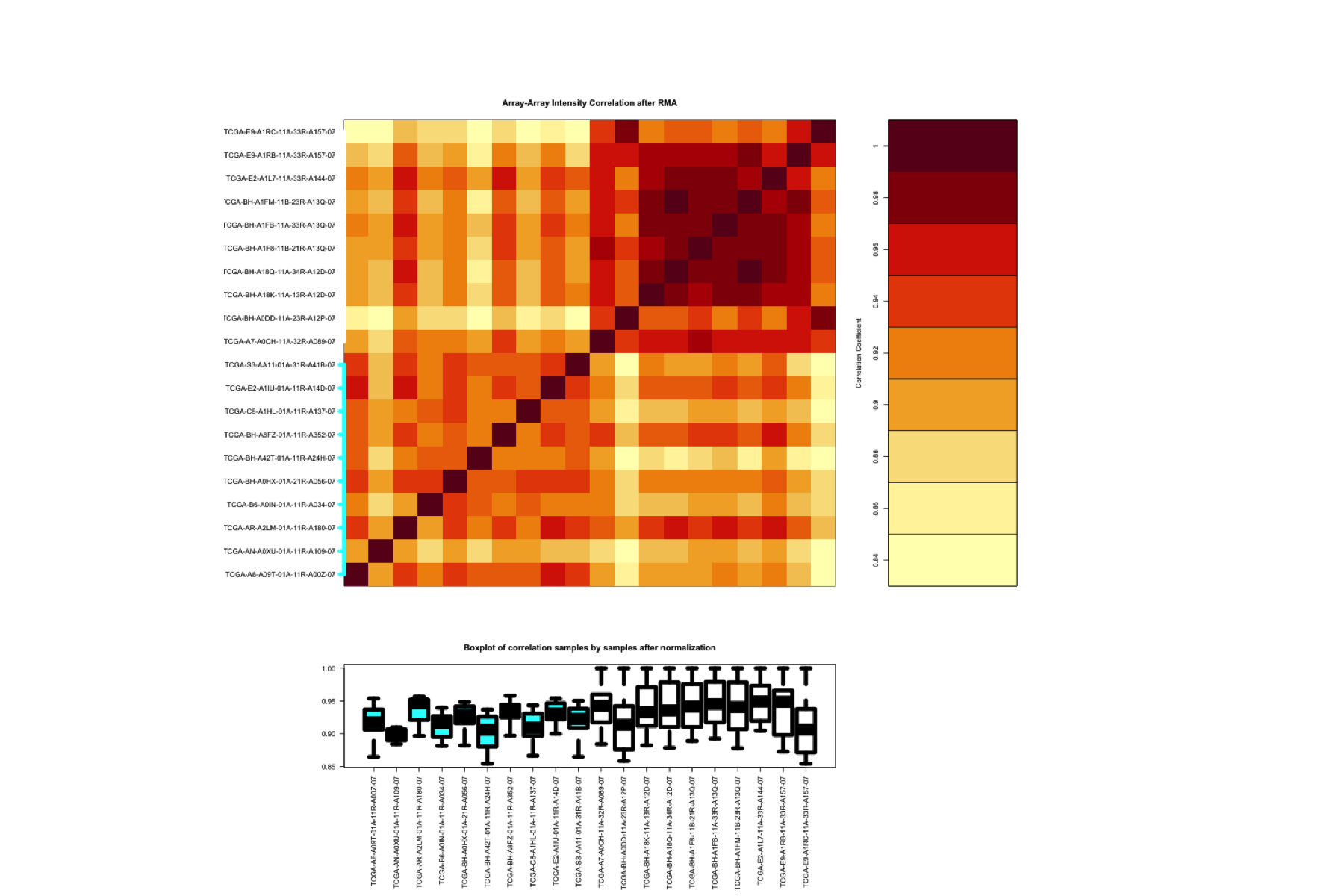
The result is shown below:

## Registered S3 methods overwritten by 'ggplot2':  
## method from   
## [.quosures rlang  
## c.quosures rlang  
## print.quosures rlang

## Registered S3 method overwritten by 'R.oo':  
## method from   
## throw.default R.methodsS3

Example of a matrix of gene expression (10 genes in rows and 2 samples in columns)

|  |  |  |
| --- | --- | --- |
|  | TCGA-BH-A0DK-11A-13R-A089-07 | TCGA-BH-A0AU-01A-11R-A12P-07 |
| SAMSN1|64092 | 152.00 | 437.00 |
| RFPL1|5988 | 5.75 | 8.85 |
| DNAJA1|3301 | 11365.00 | 6937.00 |
| UBQLN4|56893 | 5435.00 | 9690.00 |
| OR11H6|122748 | 0.00 | 0.00 |
| WNT9A|7483 | 106.00 | 34.00 |
| SCRIB|23513 | 4862.00 | 8292.00 |
| WBP4|11193 | 1064.00 | 502.00 |
| LOC93622|93622 | 1762.00 | 1198.00 |
| GLRA4|441509 | 51.00 | 2.00 |

The result from TCGAanalyze\_Preprocessing is shown below: 

## GDC TCGA Protein Expression data (RPPA) using TCGAbiolinks

You can easily search TCGA samples, download and prepare a matrix of protein RPPA expression.

cancerType <- "BRCA"  
  
query.RPPA <- GDCquery(project = paste0("TCGA-",cancerType),   
 legacy = TRUE,  
 data.category = "Protein expression",  
 platform = "MDA\_RPPA\_Core",   
 data.type = "Protein expression quantification",  
 file.type = "expression",   
 sample.type = c("Primary solid Tumor"))  
  
samples.RPPA <- query.RPPA$results[[1]]$cases  
  
  
query.RPPA.down <- GDCquery(project = paste0("TCGA-",cancerType),   
 legacy = TRUE,  
 data.category = "Protein expression",  
 data.type = "Protein expression quantification",  
 platform = "MDA\_RPPA\_Core",   
 file.type = "expression",   
 sample.type = c("Primary solid Tumor"),  
 barcode = samples.RPPA)  
  
  
GDCdownload(query.RPPA.down,   
 directory = PathDir)  
  
data.RPPA <- GDCprepare(query.RPPA.down,   
 directory = PathDir)

## GDC TCGA Copy number variation (CNV) Affymetrix SNP Array 6.0 using TCGAbiolinks

You can easily search TCGA samples, download and prepare a matrix of CN segments.

require(TCGAbiolinks)  
  
cancerType <- "BRCA"  
query.cnv <- GDCquery(project = cancerType,  
 data.category = "Copy number variation",  
 legacy = TRUE,  
 data.type = "Copy number segmentation",  
 platform = "Affymetrix SNP Array 6.0",  
 file.type = "nocnv\_hg19.seg",  
 sample.type = c("Primary solid Tumor"))  
  
samples.cnv <- query.cnv$results[[1]]$cases[1:20]  
  
query.cnv.down <- GDCquery(project = cancerType,  
 data.category = "Copy number variation",  
 legacy = TRUE,  
 platform = "Affymetrix SNP Array 6.0",  
 data.type = "Copy number segmentation",  
 file.type = "nocnv\_hg19.seg",  
 sample.type = c("Primary solid Tumor"),  
 barcode = samples.cnv)  
  
GDCdownload(query.cnv.down,  
 directory = PathDir)  
  
data.cnv <- GDCprepare(query.cnv.down,   
 directory = PathDir)  
  
data.cnv <- as.data.frame(data.cnv)

## GDC TCGA DNA methylation Illumina Human Methylation 450 using TCGAbiolinks

You can easily search TCGA samples, download and prepare a matrix of DNA methylation probes.

require(TCGAbiolinks)  
cancerType <- "BRCA"  
query.met <- GDCquery(project = cancerType,   
 legacy = TRUE,  
 data.category = "DNA methylation",  
 platform = "Illumina Human Methylation 450",  
 sample.type = c("Primary solid Tumor"))  
  
samples.met <- query.met$results[[1]]$cases[1:20]  
  
query.met.down <- GDCquery(project = cancerType,   
 legacy = TRUE,  
 data.category = "DNA methylation",  
 platform = "Illumina Human Methylation 450",  
 sample.type = c("Primary solid Tumor"),  
 barcode = samples.met)  
  
GDCdownload(query.met.down,  
 directory = PathDir)  
  
data.met <- GDCprepare(query.met.down,   
 directory = PathDir)

## GDC TCGA Mutation MAF using TCGAbiolinks

You can easily search TCGA samples, download and prepare a matrix of Mutations.

cancerType <- "TCGA-BRCA"  
query.mut <- GDCquery(project = cancerType,   
 data.category = "Simple nucleotide variation",   
 data.type = "Simple somatic mutation",  
 access = "open",   
 sample.type = c("Primary solid Tumor"),  
 legacy = TRUE)  
  
# Check maf availables  
query.mut$results[[1]]$file\_name  
  
query.mut <- GDCquery(project = cancerType,   
 data.category = "Simple nucleotide variation",   
 data.type = "Simple somatic mutation",  
 access = "open",   
 sample.type = c("Primary solid Tumor"),  
 legacy = TRUE,  
 file.type = query.mut$results[[1]]$file\_name[1])  
  
samples.mut <- query.mut$results[[1]]$cases  
  
  
query.mut.down <- GDCquery(project = cancerType,   
 data.category = "Simple nucleotide variation",   
 data.type = "Simple somatic mutation",  
 access = "open",   
 sample.type = c("Primary solid Tumor"),  
 legacy = TRUE,  
 barcode = samples.mut,  
 file.type = query.mut$results[[1]]$file\_name[1])  
  
GDCdownload(query.mut.down,  
 directory = PathDir)  
  
data.mut <- GDCprepare(query.mut.down,   
 directory = PathDir)

## GDC TCGA microRNA Expression using TCGAbiolinks

You can easily search TCGA samples, download and prepare a matrix of microRNA Expression

cancerType <- "TCGA-BRCA"  
query.miR <- GDCquery(project = cancerType,   
 data.category = "Gene expression",  
 data.type = "miRNA gene quantification",  
 platform = "Illumina HiSeq",  
 file.type = "hg19.mirbase20.mirna.quantification",  
 legacy = TRUE)  
samples.miR <- query.miR$results[[1]]$cases  
  
query.miR.down <- GDCquery(project = cancerType,   
 data.category = "Gene expression",  
 data.type = "miRNA gene quantification",  
 platform = "Illumina HiSeq",  
 file.type = "hg19.mirbase20.mirna.quantification",  
 legacy = TRUE,  
 barcode = TNBCsamplesmiRlong)  
  
GDCdownload(query.miR.down,  
 directory = PathDir)  
  
data.miR <- GDCprepare(query.miR.down,   
 directory = PathDir)

## GDC TCGA Clinical Data using TCGAbiolinks

You can easily search TCGA samples, download and prepare a matrix of clinical data

cancerType <- "BRCA"  
dataClin <- GDCquery\_clinic(project = paste0("TCGA-",cancerType),type = "clinical")

## GTEx Data using TCGAbiolinks

You can easily search GTEx samples, download and prepare a matrix of gene expression.

data\_gtex\_brain <- TCGAquery\_recount2("gtex", tissue = "brain")

## IHEC gene expression data using DeepBlueR

You can easily search IHEC Blueprint samples, download and prepare a matrix of gene expression.

require(DeepBlueR)  
require(dplyr)  
  
  
# List all BLUEPRINT samples  
blueprint\_samples <- deepblue\_list\_samples(  
 extra\_metadata = list("source" = "BLUEPRINT Epigenome"))  
  
# Extract their ids  
blueprint\_samples\_ids <- deepblue\_extract\_ids(blueprint\_samples)  
  
# Select gene expression data. We assign gene names using Gencode 22  
gene\_exprs\_query <- deepblue\_select\_expressions(sample\_ids = blueprint\_samples\_ids,  
 expression\_type = "gene",  
 gene\_model = "gencode v22")  
  
gene\_exprs\_query <- deepblue\_select\_expressions(sample\_ids = blueprint\_samples\_ids,  
 expression\_type = "gene",  
 gene\_model = "gencode v19")  
  
  
  
# We request the data and define the output format  
request = deepblue\_get\_regions(query\_id = gene\_exprs\_query,  
 "@GENE\_ID(gencode v19),FPKM,@BIOSOURCE,@SAMPLE\_ID")  
  
# We download the data  
gene\_regions <- deepblue\_download\_request\_data(request)  
  
# We retain a table mapping sample ids to bisources  
sample\_names <- dplyr::select(gene\_regions, `@BIOSOURCE`, `@SAMPLE\_ID`) %>%  
 dplyr::distinct()  
  
# We filter out duplicated gene entries  
genes\_one\_sample <- dplyr::filter(gene\_regions, `@SAMPLE\_ID` == "s10678")  
duplicated\_genes <- genes\_one\_sample[  
 which(duplicated(genes\_one\_sample$`@GENE\_ID(gencode v22)`)),  
 "@GENE\_ID(gencode v22)"]  
  
  
genes\_one\_sample <- dplyr::filter(gene\_regions, `@SAMPLE\_ID` == "s10678")  
duplicated\_genes <- genes\_one\_sample[  
 which(duplicated(genes\_one\_sample$`@GENE\_ID(gencode v19)`)),  
 "@GENE\_ID(gencode v19)"]  
  
  
# We convert the gene expression from a list to a data frame and subsequently...  
genes\_matrix = dplyr::filter(gene\_regions,  
 !(`@GENE\_ID(gencode v22)` %in% duplicated\_genes)) %>%  
 dplyr::select(-`@BIOSOURCE`) %>%  
 tidyr::spread(key = `@SAMPLE\_ID`, value = FPKM)  
  
# ...to a numeric matrix  
genes <- genes\_matrix[,1]  
genes\_matrix <- data.matrix(genes\_matrix[,-1])  
rownames(genes\_matrix) <- genes  
  
### OUTPUT  
### genes\_matrix : The gene expression matrix for all 276 BLUEPRINT samples  
### sample\_names : A mapping table from sample id to cell type / biosource  
  
save(genes\_matrix, file = "IHEC\_genes\_matrix.Rdata")  
save(sample\_names, file = "IHEC\_Sample\_names.rdata")

## GEO gene expression data using MoonlightR and GEOquery

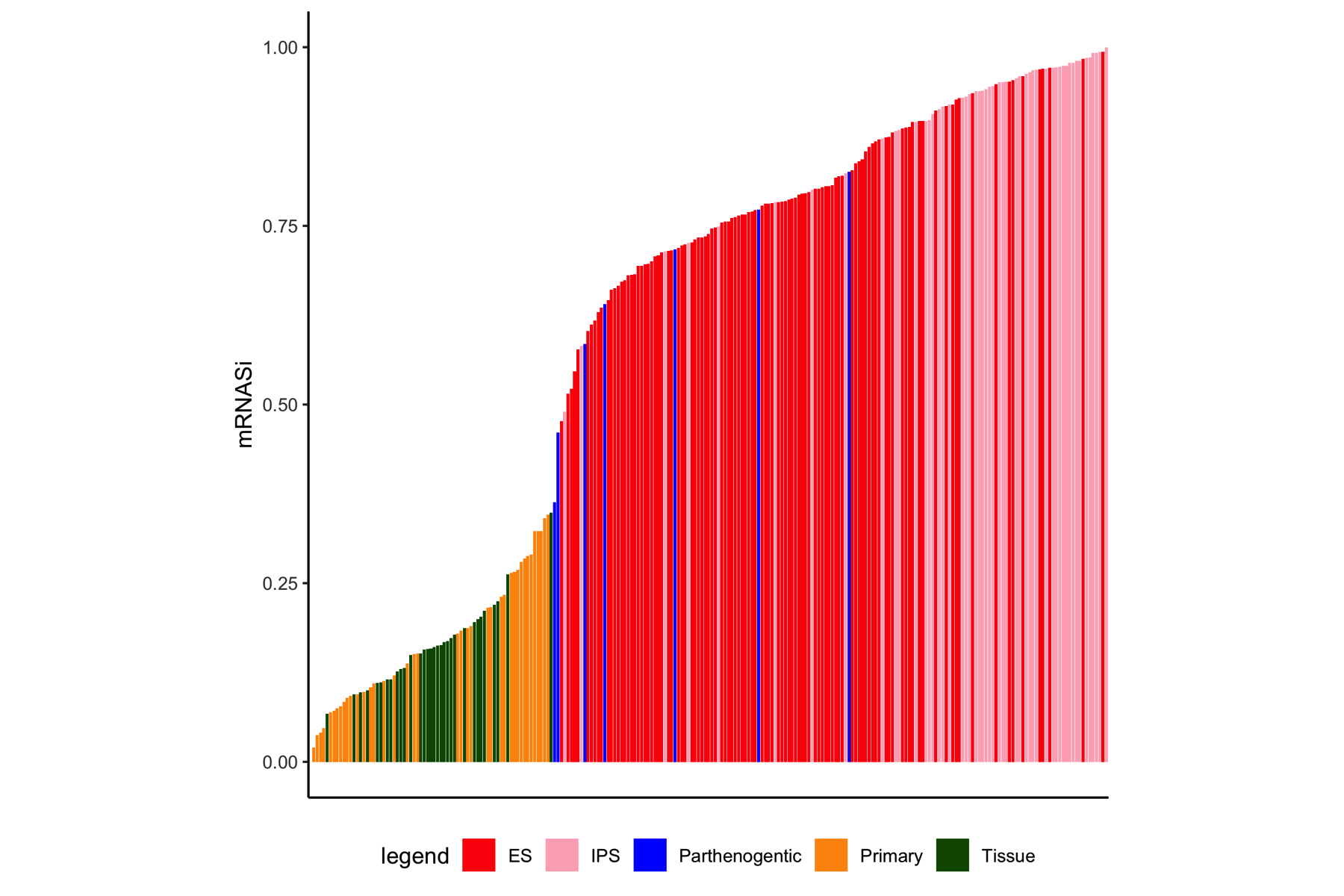
## Preparing Gene Expression for GEO (platform and GSE) for Nazor et al., Cell Stem Cell 2012

You can easily search GEO samples, download and prepare a matrix of gene expression.

require(MoonlightR)  
library(devtools)  
install\_github("ibsquare/MoonlightR")   
  
dataNazor <- getDataGEO(GEOobject = "GSE30652",  
 platform = "GPL6947")  
  
require(SummarizedExperiment)  
GSE30652 <- as.data.frame(exprs(dataNazor))  
GSE30652\_non\_norm <- cbind(ILMN = rownames(GSE30652),  
 IDmean = rowMeans(GSE30652),  
 GSE30652)  
  
  
  
dataNazor\_samples <- pData(dataNazor)  
dataNazor\_samples <- as.data.frame(dataNazor\_samples)  
dataNazor\_samples <- subset(dataNazor\_samples,  
 select = c("geo\_accession","characteristics\_ch1.2"))  
colnames(dataNazor\_samples)[2] <- "CellType"  
  
dataNazor\_samples$CellType <- gsub("cell type: ","",dataNazor\_samples$CellType)  
dataNazor\_samples$CellType <- gsub(", undifferentiated","",dataNazor\_samples$CellType)  
  
GPL6947\_13512 <- fData(dataNazor)  
GPL6947\_13512\_annot <- as.data.frame(GPL6947\_13512)  
GPL6947\_13512\_annot <- subset(GPL6947\_13512\_annot,  
 select = c("ID","Gene.symbol"))  
  
  
  
GSE30652\_merge <- merge(x = GPL6947\_13512\_annot,  
 y = GSE30652\_non\_norm,  
 by.x = "ID",  
 by.y = "ILMN")  
  
  
GSE30652\_merge <- GSE30652\_merge[order(GSE30652\_merge$IDmean,decreasing = TRUE),]  
GSE30652\_merge <- GSE30652\_merge[!duplicated(GSE30652\_merge$Gene.symbol),]  
NazorMatrix <- GSE30652\_merge  
rownames(NazorMatrix) <- NazorMatrix$Gene.symbol  
  
NazorMatrix <- NazorMatrix[,dataNazor\_samples$geo\_accession]

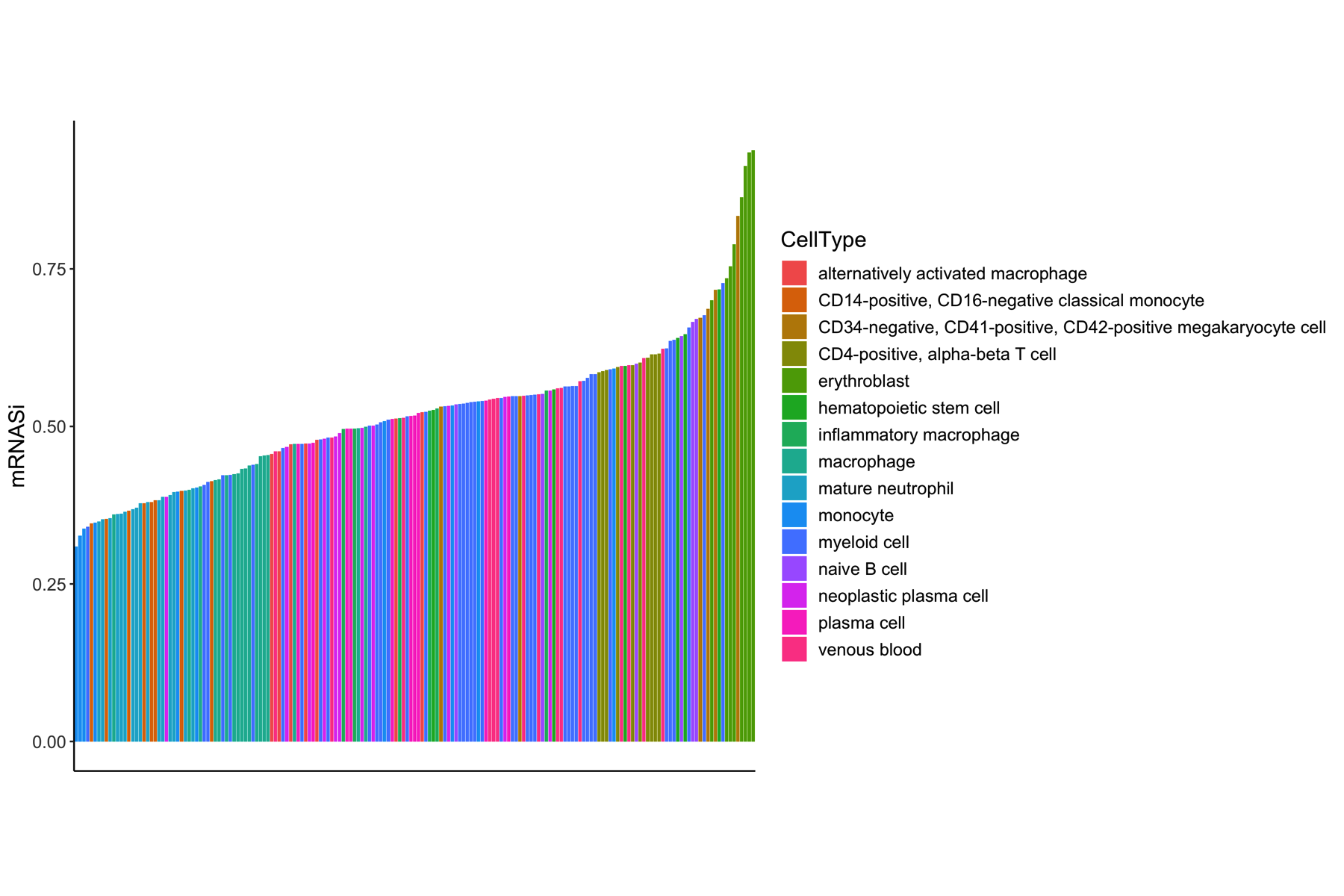
# Validation of Stemness signature in external dataset. Nazor et al., Cell Stem Cell 2012

# Firstly we have previously prepared the Gene Expression matrix (genes in rows , samples in columns)   
# for Nazor's dataset using Moonlight and getGEO's function.  
  
TCGA\_mRNA\_StemScoreTable <- TCGAanalyze\_Stemness(stemSig = PCBC\_stemSig,  
 dataGE = NazorMatrix)  
  
  
  
tab\_mRNASi <- TCGA\_mRNA\_StemScoreTable  
  
tab\_mRNASi\_merged <- merge(x = tab\_mRNASi,  
 y = dataNazor\_samples,  
 by.x = "Sample",  
 by.y = "geo\_accession")  
require(ggplot2)  
require(ggpubr)  
  
colnames(tab\_mRNASi\_merged)[3] <- "mRNASi"  
  
tab\_mRNASi\_merged[tab\_mRNASi\_merged$CellType %in% "embryonic stem cell","CellType"] <- "ES"  
tab\_mRNASi\_merged[tab\_mRNASi\_merged$CellType %in% "induced pluripotent stem cell","CellType"] <- "IPS"  
tab\_mRNASi\_merged[tab\_mRNASi\_merged$CellType %in% "parthenogentic embryonic stem cell","CellType"] <- "Parthenogentic"  
tab\_mRNASi\_merged[tab\_mRNASi\_merged$CellType %in% "Somatic.Primary","CellType"] <- "Primary"  
tab\_mRNASi\_merged[tab\_mRNASi\_merged$CellType %in% "Somatic.Tissue","CellType"] <- "Tissue"  
  
tab\_mRNASi\_merged <- tab\_mRNASi\_merged[order(tab\_mRNASi\_merged$mRNASi, decreasing = FALSE),]  
library(forcats)  
p <- ggplot(data=tab\_mRNASi\_merged, aes(x=Sample, y=mRNASi, fill = CellType)) +  
 geom\_bar(stat="identity")+  
 scale\_colour\_gradient2()+  
 #coord\_flip()+  
 # ylim(0, 15)+  
 #scale\_x\_discrete(limits = df1$Tissue)+  
 theme\_classic() +  
 theme(axis.title.x=element\_blank(),  
 axis.text.x=element\_blank(),  
 axis.ticks.x=element\_blank())+  
 scale\_fill\_manual("legend", values = c("Primary" = "orange",  
 "Parthenogentic" = "blue",  
 "Tissue" = "darkgreen",  
 "ES" = "red",  
 "IPS" = "pink"))  
p <- p + theme(legend.position="bottom")  
  
p <- p + aes(x = fct\_inorder(Sample))  
ggsave(p, file = "Validation\_mRNASi\_Nazor.png", width = 6,height = 6)

The result from TCGAanalyze\_Stemness validation in Nazor’s Dataset is shown below: 

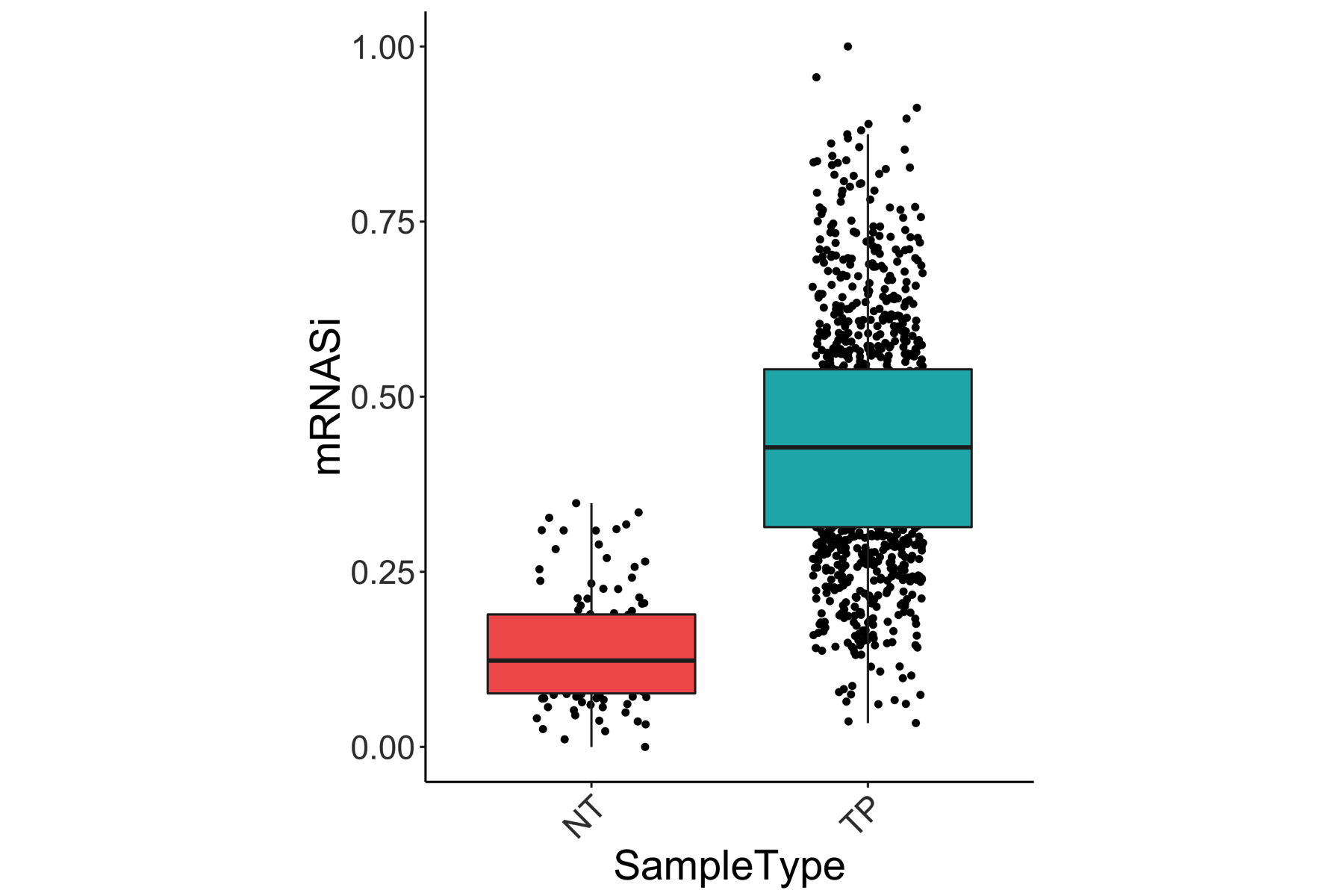
# Validation of Stemness signature in external dataset. IHEC Author et al., Cell 2016

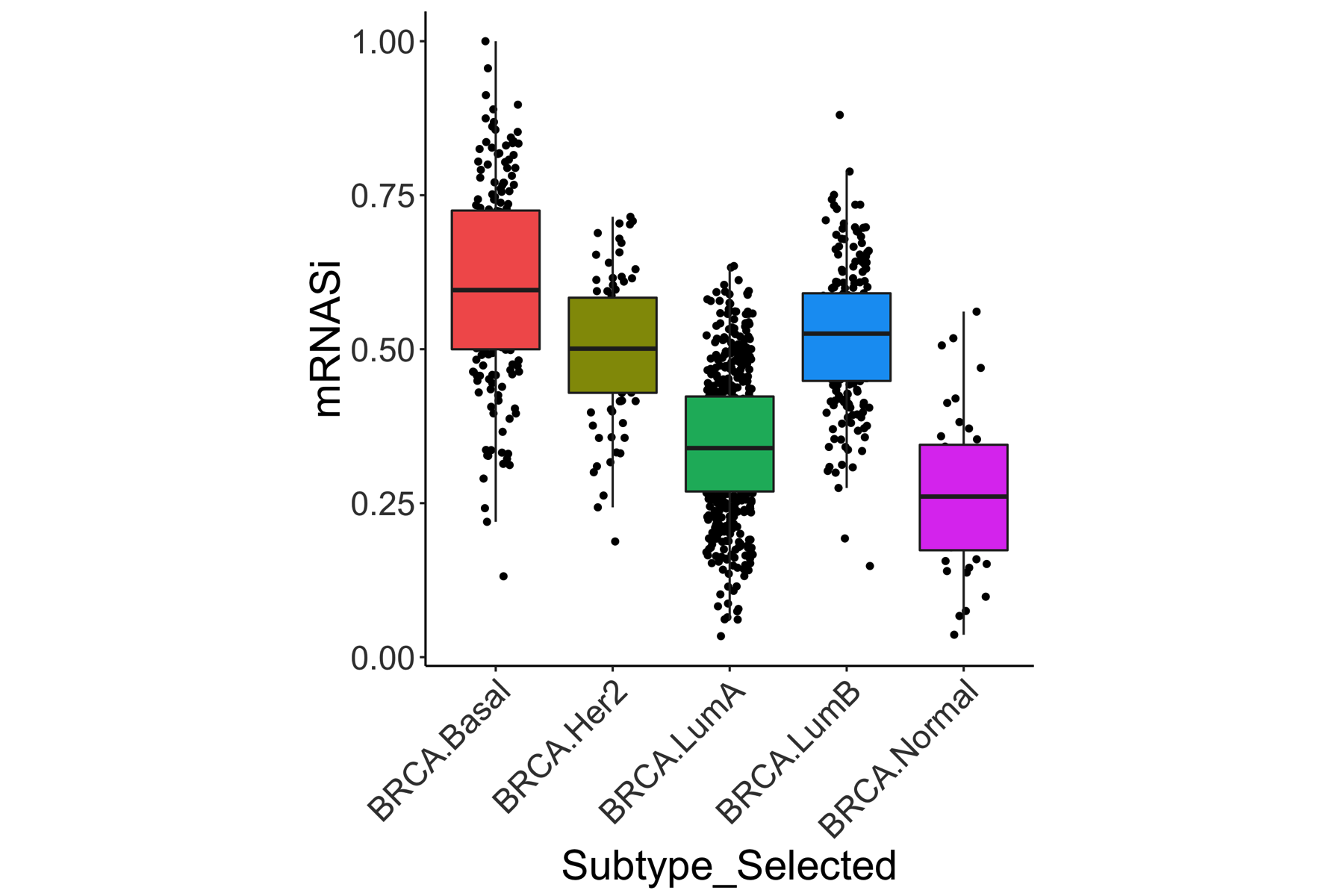
# stemness score with IHEC samples  
load("./IHEC\_genes\_matrix.Rdata")  
load("./IHEC\_Sample\_names.rdata")  
IHECMatrix <- genes\_matrix  
require(stringr)  
IHEC\_annot <- sample\_names  
colnames(IHEC\_annot) <- c("Diffname\_short","X1")  
IHEC\_annot <- as.data.frame(IHEC\_annot)  
  
library("biomaRt")  
ensemblsIDS <- str\_split\_fixed(rownames(IHECMatrix),"\\.",2)[,1]  
rownames(IHECMatrix)<- ensemblsIDS  
  
IHEC\_annot\_common <- IHEC\_annot[IHEC\_annot$X1 %in% colnames(IHECMatrix),]  
  
  
# mapping Ensembl IDs to gene Symbol  
  
require(TCGAbiolinks)  
library(clusterProfiler)  
library(org.Hs.eg.db)  
gene.df <- bitr(ensemblsIDS, fromType = "ENSEMBL",  
 toType = c( "ENTREZID", "SYMBOL"),  
 OrgDb = org.Hs.eg.db)  
  
  
IHECMatrix\_sel <- IHECMatrix[gene.df$ENSEMBL,]  
rownames(IHECMatrix\_sel) <-gene.df$SYMBOL  
IHECMatrix\_sel <- as.data.frame(IHECMatrix\_sel)  
  
IHECMatrix\_sel\_filt <- TCGAanalyze\_Filtering(tabDF = IHECMatrix\_sel,  
 method = "quantile",  
 qnt.cut = 0.25)  
  
TCGA\_mRNA\_StemIHEC <- TCGAanalyze\_Stemness(stemSig = PCBC\_stemSig,  
 dataGE = IHECMatrix\_sel\_filt)  
  
colnames(TCGA\_mRNA\_StemIHEC)[3] <- "mRNASi"  
colnames(IHEC\_annot)[1] <- "CellType"  
colnames(IHEC\_annot)[2] <- "Sample"  
TCGA\_mRNA\_StemIHEC\_merge<- merge(x = TCGA\_mRNA\_StemIHEC,  
 y = IHEC\_annot,  
 by.x = "Sample",  
 by.y = "Sample")  
tab\_mRNASi\_merged <- TCGA\_mRNA\_StemIHEC\_merge  
  
CellType <- table(tab\_mRNASi\_merged$CellType)  
  
  
  
CellType\_filt <- CellType[CellType > 5]  
  
tab\_mRNASi\_merged\_filt <- tab\_mRNASi\_merged[tab\_mRNASi\_merged$CellType %in% names(CellType\_filt), ]  
tab\_mRNASi\_merged <- tab\_mRNASi\_merged\_filt  
  
  
tab\_mRNASi\_merged <- tab\_mRNASi\_merged[order(tab\_mRNASi\_merged$mRNASi, decreasing = FALSE),]  
require(ggplot2)  
library(forcats)  
p <- ggplot(data=tab\_mRNASi\_merged, aes(x=Sample, y=mRNASi, fill = CellType)) +  
 geom\_bar(stat="identity")+  
 scale\_colour\_gradient2()+  
 #coord\_flip()+  
 # ylim(0, 15)+  
 #scale\_x\_discrete(limits = df1$Tissue)+  
 theme\_classic() +  
 theme(axis.title.x=element\_blank(),  
 axis.text.x=element\_blank(),  
 axis.ticks.x=element\_blank())  
p <- p + theme(legend.position="right")  
  
p <- p + aes(x = fct\_inorder(Sample))  
p <- p + theme(text = element\_text(size=14))  
ggsave(p, file = "Validation\_mRNASi\_IHEC.png", width = 12,height = 6)

The result from TCGAanalyze\_Stemness validation in IHEC’s Dataset is shown below: 

# Generating Stemness score for TCGA gene expression data (mRNASi). Malta et al. Cell, 2018

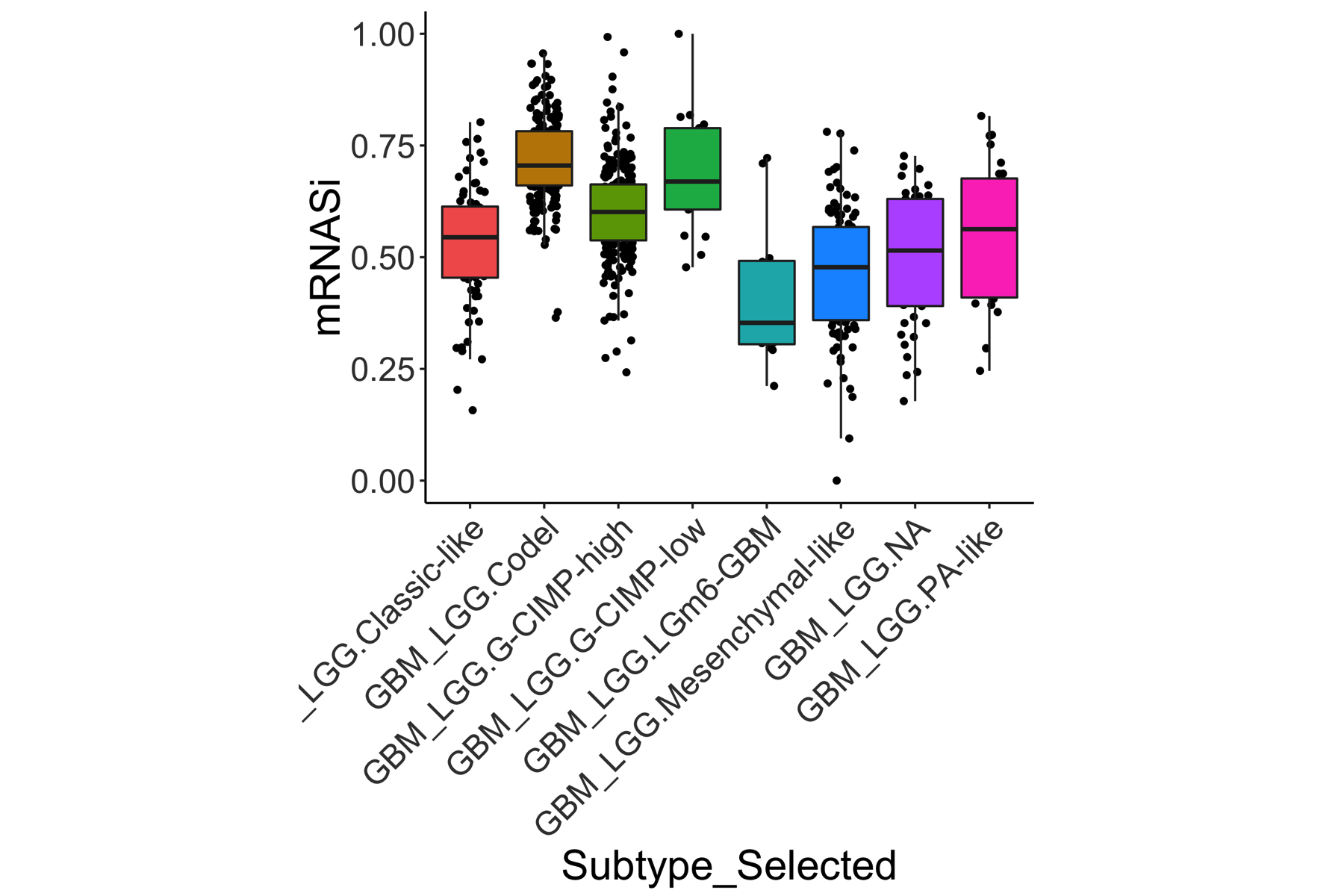
# Firstly we load Gene Expression matrix (genes in rows , samples in columns)   
# from a previous generated pancancer gene expression matrix  
load("~/Dropbox (Personal)/Umiami/TCGAanalysis/Stemness/dataFilt\_panCancer33new.Rdata")  
  
curCancer <- "BRCA"  
# We have previously generated a table with 33 cancer types barcodes and molecular subtypes.  
  
dataSubt\_PanCancer <- PanCancerAtlas\_subtypes()  
  
# Selecting TCGA breast cancer  
  
dataSubt\_curCancer <- dataSubt\_PanCancer[dataSubt\_PanCancer$cancer.type %in% "BRCA",]  
  
commonSamples <- intersect(dataSubt\_curCancer$pan.samplesID, colnames(dataFilt))  
  
dataFilt\_curCancer <- dataFilt[,commonSamples]  
load("~/Dropbox (Personal)/Umiami/Github/TCGAbiolinks/data/PCBC\_stemSig.rda")  
  
TCGA\_mRNA\_StemScoreTable <- TCGAanalyze\_Stemness(stemSig = PCBC\_stemSig,  
 dataGE = dataFilt\_curCancer)  
  
colnames(TCGA\_mRNA\_StemScoreTable)[1] <- "barcode"  
  
sampleNT <- TCGAquery\_SampleTypes(barcode = TCGA\_mRNA\_StemScoreTable$barcode,typesample = "NT")  
sampleTP <- TCGAquery\_SampleTypes(barcode = TCGA\_mRNA\_StemScoreTable$barcode,typesample = "TP")  
#sampleTM <- TCGAquery\_SampleTypes(barcode = TCGA\_mRNA\_StemScoreTable$barcode,typesample = "TM")  
#sampleTAM <- TCGAquery\_SampleTypes(barcode = TCGA\_mRNA\_StemScoreTable$barcode,typesample = "TAM")  
#sampleTB <- TCGAquery\_SampleTypes(barcode = TCGA\_mRNA\_StemScoreTable$barcode,typesample = "TB")  
  
rownames(TCGA\_mRNA\_StemScoreTable) <- TCGA\_mRNA\_StemScoreTable$barcode  
  
TCGA\_mRNA\_StemScoreTable[sampleNT,"SampleType"] <- "NT"  
TCGA\_mRNA\_StemScoreTable[sampleTP,"SampleType"] <- "TP"  
#TCGA\_mRNA\_StemScoreTable[sampleTM,"SampleType"] <- "TM"  
#TCGA\_mRNA\_StemScoreTable[sampleTAM,"SampleType"] <- "TM"  
#TCGA\_mRNA\_StemScoreTable[sampleTB,"SampleType"] <- "TB"  
  
tab\_mRNASi <- TCGA\_mRNA\_StemScoreTable  
  
tab\_mRNASi\_merged <- merge(x = tab\_mRNASi,  
 y = dataSubt\_PanCancer,  
 by.x = "barcode",  
 by.y = "pan.samplesID")  
require(ggplot2)  
require(ggpubr)  
  
colnames(tab\_mRNASi\_merged)[3] <- "mRNASi"  
  
  
  
p<-ggplot(tab\_mRNASi\_merged, aes(x=SampleType, y=mRNASi, fill=SampleType))  
p <- p + theme\_classic()  
  
p <- p + theme(legend.position="none")  
p <- p + rotate\_x\_text(45)  
p <- p + geom\_jitter(shape=16, position=position\_jitter(0.2), color = "black")  
p <- p + geom\_boxplot(position=position\_dodge(1),outlier.colour = NA)  
p <- p + theme(text = element\_text(size=20))  
ggsave(p , filename = "TCGA\_BRCA\_mRNASi\_TP\_NT.png",width = 5,height = 6)  
  
  
tab\_mRNASi\_merged\_TP <- tab\_mRNASi\_merged[tab\_mRNASi\_merged$SampleType %in% "TP",]  
  
p<-ggplot(tab\_mRNASi\_merged\_TP, aes(x=Subtype\_Selected, y=mRNASi, fill=Subtype\_Selected))  
p <- p + theme\_classic()  
  
p <- p + theme(legend.position="none")  
p <- p + rotate\_x\_text(45)  
p <- p + geom\_jitter(shape=16, position=position\_jitter(0.2), color = "black")  
p <- p + geom\_boxplot(position=position\_dodge(1),outlier.colour = NA)  
p <- p + theme(text = element\_text(size=20))  
ggsave(p , filename = "TCGA\_BRCA\_mRNASi\_subtypes.png",width = 5,height = 6)

The result from TCGAanalyze\_Stemness is shown below: 

The result from TCGAanalyze\_Stemness with molecular subtypes is shown below: 

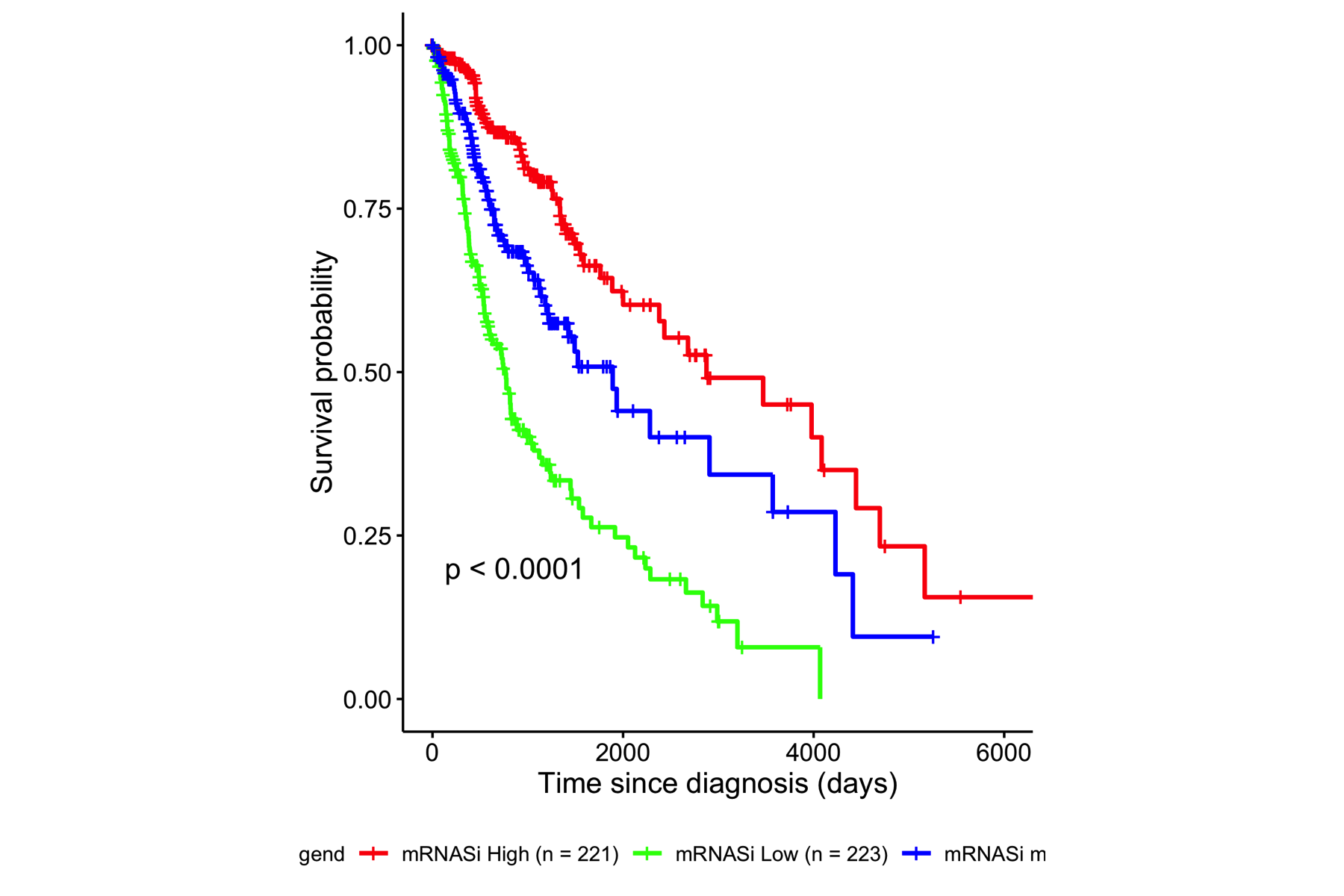
# Generating Stemness score for TCGA LGG and GBM gene expression data (mRNASi). Malta et al. Cell, 2018

# working with LGG and GBM  
curCancer <- c("LGG","GBM")  
# We have previously generated a table with 33 cancer types barcodes and molecular subtypes.  
require(TCGAbiolinks)  
dataSubt\_PanCancer <- PanCancerAtlas\_subtypes()  
  
# Selecting TCGA breast cancer  
  
dataSubt\_curCancer <- dataSubt\_PanCancer[dataSubt\_PanCancer$cancer.type %in% curCancer,]  
  
sampleCurCancer <- colnames(dataFilt)  
sampleCurCancer <- sampleCurCancer[substr(sampleCurCancer,1,12) %in% dataSubt\_curCancer$pan.samplesID]  
  
dataFilt\_curCancer <- dataFilt[,sampleCurCancer]  
load("~/Dropbox (Personal)/Umiami/Github/TCGAbiolinks/data/PCBC\_stemSig.rda")  
  
TCGA\_mRNA\_StemScoreTable <- TCGAanalyze\_Stemness(stemSig = PCBC\_stemSig,  
 dataGE = dataFilt\_curCancer)  
  
colnames(TCGA\_mRNA\_StemScoreTable)[1] <- "barcode"  
  
  
tab\_mRNASi <- TCGA\_mRNA\_StemScoreTable  
tab\_mRNASi <- cbind(barcode12 = substr(tab\_mRNASi$barcode,1,12),  
 tab\_mRNASi)  
tab\_mRNASi$barcode12 <- as.character(tab\_mRNASi$barcode12)  
  
  
tab\_mRNASi\_merged <- merge(x = tab\_mRNASi,  
 y = dataSubt\_PanCancer,  
 by.x = "barcode12",  
 by.y = "pan.samplesID")  
require(ggplot2)  
require(ggpubr)  
  
colnames(tab\_mRNASi\_merged)[4] <- "mRNASi"  
  
p<-ggplot(tab\_mRNASi\_merged, aes(x=Subtype\_Selected, y=mRNASi, fill=Subtype\_Selected))  
p <- p + theme\_classic()  
p <- p + theme(legend.position="none")  
p <- p + rotate\_x\_text(45)  
p <- p + geom\_jitter(shape=16, position=position\_jitter(0.2), color = "black")  
p <- p + geom\_boxplot(position=position\_dodge(1),outlier.colour = NA)  
p <- p + theme(text = element\_text(size=20))  
ggsave(p , filename = "TCGA\_LGG\_GBM\_mRNASi\_subtypes.png",width = 5,height = 6)

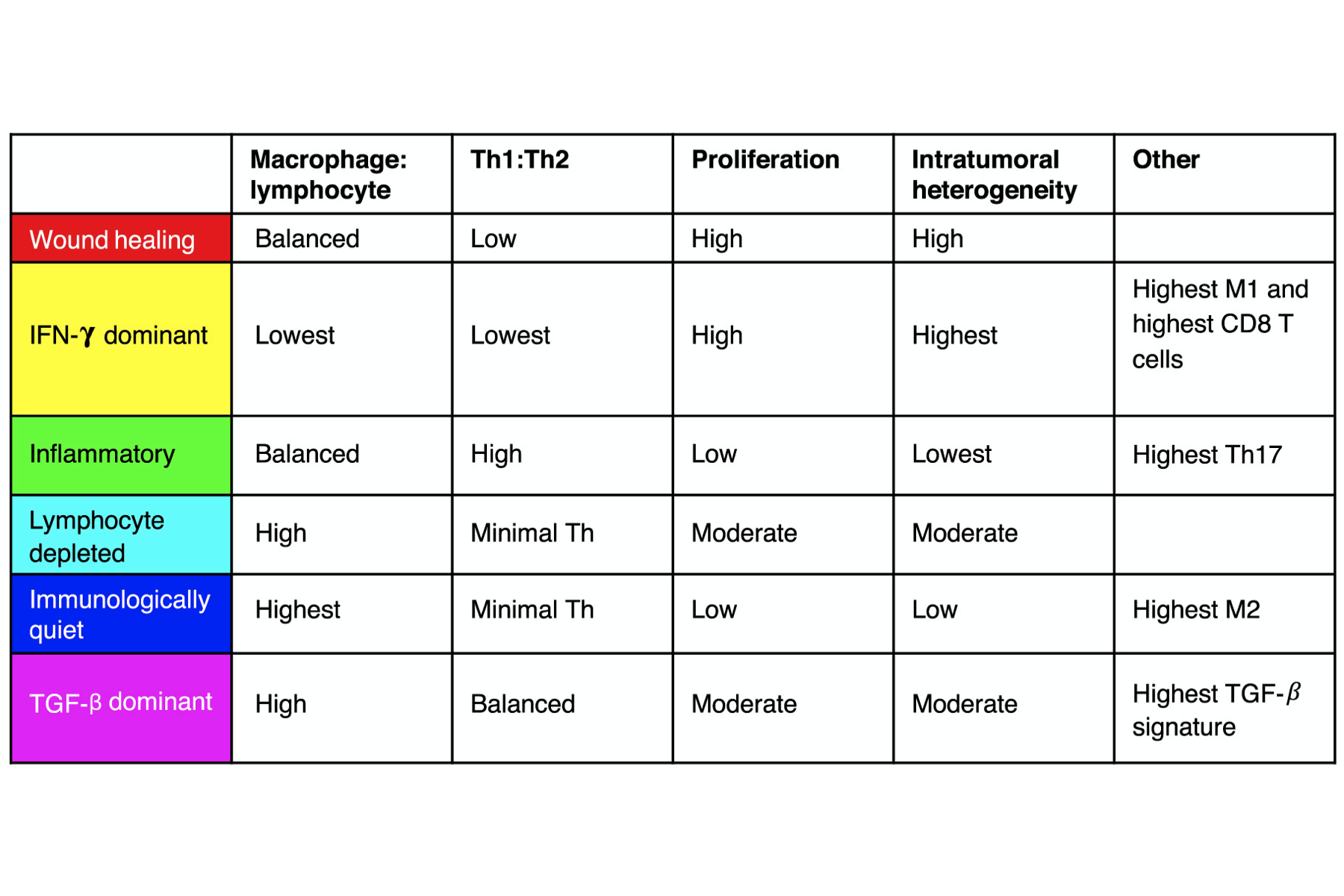
The result from TCGAanalyze\_Stemness for LGG and GBM is shown below: 

# Stemness index is associated with overall survival (Glioma example)

#working with survival glioma  
  
tab\_mRNASi\_merged <- cbind(mRNASi\_level = rep("mRNASi mod",nrow(tab\_mRNASi\_merged)),  
 tab\_mRNASi\_merged)  
  
tabSi<- tab\_mRNASi\_merged  
tabSi$mRNASi\_level <- as.character(tabSi$mRNASi\_level)  
  
tabSi[tabSi$mRNASi < quantile(tabSi$mRNASi,1/3),"mRNASi\_level"] <- "mRNASi Low"  
tabSi[tabSi$mRNASi > quantile(tabSi$mRNASi,2/3),"mRNASi\_level"] <- "mRNASi High"  
  
  
require(TCGAbiolinks)  
  
dataClin\_LGG <- GDCquery\_clinic(project = "TCGA-LGG",type = "clinical")  
dataClin\_GBM <- GDCquery\_clinic(project = "TCGA-GBM",type = "clinical")  
  
dataClin\_LGG\_GBM <- rbind(dataClin\_LGG,dataClin\_GBM)  
  
dataClin\_LGG\_GBM <- dataClin\_LGG\_GBM[dataClin\_LGG\_GBM$submitter\_id %in% tabSi$barcode12,]  
  
dataClin\_merged <- merge(x = dataClin\_LGG\_GBM,  
 y = tabSi,  
 by.x = "submitter\_id",  
 by.y = "barcode12")  
  
  
  
  
p <- TCGAanalyze\_survival(dataClin\_merged,  
 clusterCol = "mRNASi\_level",  
 conf.int = FALSE,  
 main = "TCGA LGG GBM mRNASi",   
 height = 10,  
 width=10,  
 risk.table = TRUE,  
 filename = NULL)  
  
p <- p$plot  
  
p <- p + theme(legend.position = "bottom")  
  
ggsave(p , filename = "TCGA\_LGG\_GBM\_mRNASi\_survival.png",width = 5,height = 6)

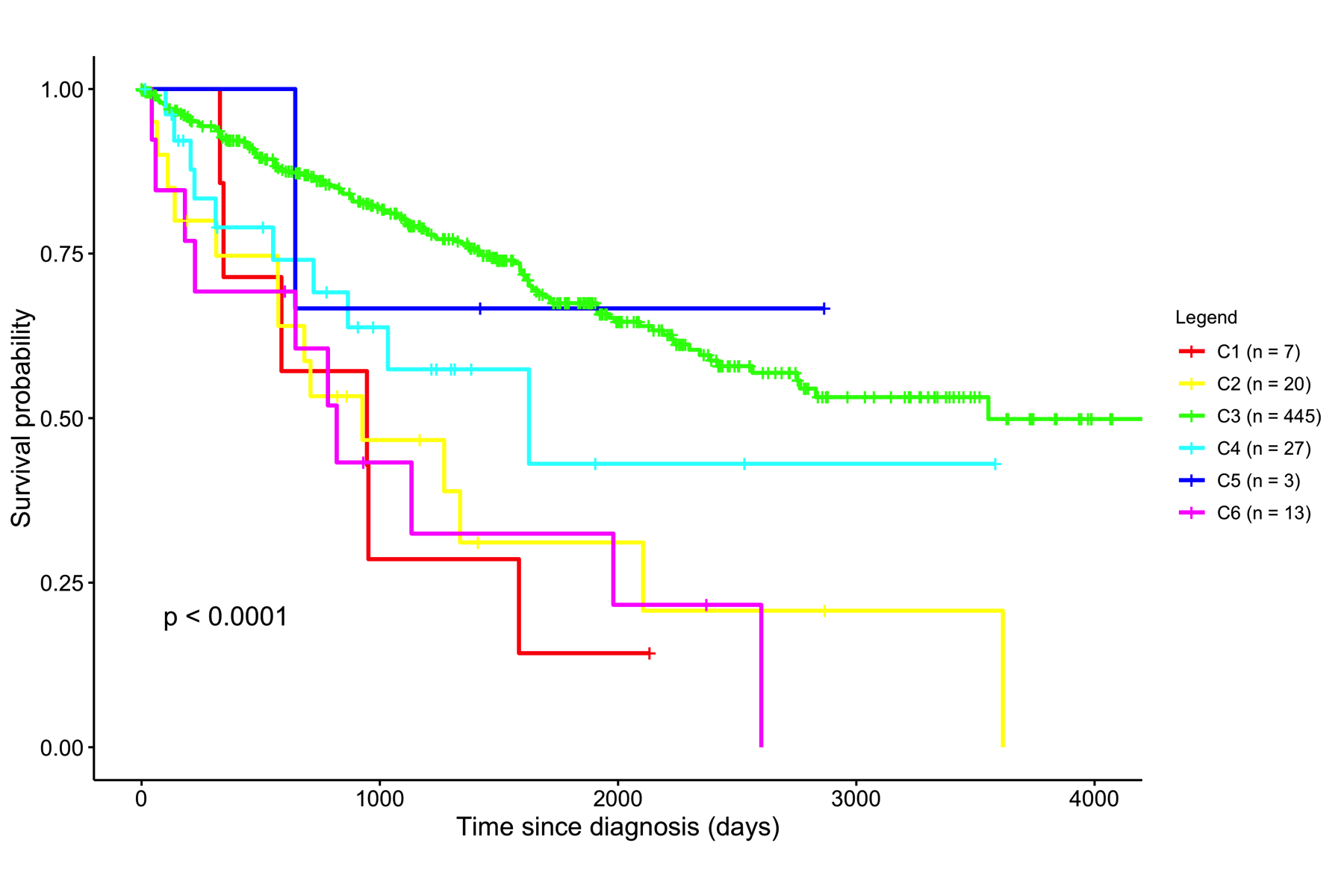
The result from the Stemness-survival association for LGG and GBM is shown below: 

In this section we generate the Immune subtypes for TCGA and GEO tumors. Figure1C readapted from Thorsson et al., Immunity, 2018, to summarize features of six different immune subtypes.



# KM survival analysis of TCGA KIRC samples with Immune Subtypes from Thorsson et al., Immunity, 2018

download.file(url = "https://ars.els-cdn.com/content/image/1-s2.0-S1074761318301213-mmc2.xlsx",  
 destfile = "X1\_s2\_0\_S1074761318301213\_mmc2")  
X1\_s2\_0\_S1074761318301213\_mmc2 <- read\_excel("X1\_s2\_0\_S1074761318301213\_mmc2")  
X1\_s2\_0\_S1074761318301213\_mmc2 <- as.data.frame(X1\_s2\_0\_S1074761318301213\_mmc2)  
  
CancerType <- c("KIRC")  
  
ImmuneSubtypes <- as.data.frame(X1\_s2\_0\_S1074761318301213\_mmc2)  
  
ImmuneSubtypes <- ImmuneSubtypes[ImmuneSubtypes$`TCGA Study` %in% CancerType,]  
ImmuneSubtypes <- ImmuneSubtypes[ImmuneSubtypes$`Immune Subtype`!="NA",]  
  
dataClin\_KIRC <- GDCquery\_clinic(project = "TCGA-KIRC",type = "clinical")  
  
dataClin\_merged <- merge(x = dataClin\_KIRC,  
 y = ImmuneSubtypes,  
 by.x = "submitter\_id",  
 by.y = "TCGA Participant Barcode")  
  
  
p <- TCGAanalyze\_survival(dataClin\_merged,  
 clusterCol = "Immune Subtype",  
 conf.int = FALSE,  
 main = "TCGA KIRC Immune Subtypes",  
 height = 10,  
 width=10,  
 risk.table = TRUE,  
 filename = NULL)  
  
p <- p$plot  
  
p <- p + theme(legend.position = "right")  
  
ggsave(p , filename = "TCGA\_KIRC\_ImmuneSubtypes\_survival.png",width = 10,height = 6)

The result from the Immune subtypes -survival association for BRCA samples is shown below: 

# Immune Subtypes for GEO samples (KIRC example) using data from Jones et al.,Clin Cancer Res 2005 (16115910)

require(MoonlightR)  
   
#Jones J, Otu H, Spentzos D, Kolia S et al. Gene signatures of progression and metastasis in renal cell cancer.   
# Clin Cancer Res 2005 Aug 15;11(16):5730-9. PMID: 16115910  
  
dataGEO\_KIRC<- getDataGEO(GEOobject = "GSE15641",  
 platform = "GPL96")  
require(SummarizedExperiment)  
  
GSE15641 <- as.data.frame(exprs(dataGEO\_KIRC))  
GSE15641\_non\_norm <- cbind(ILMN = rownames(GSE15641),  
 IDmean = rowMeans(GSE15641),  
 GSE15641)  
  
GSE15641\_annot <- fData(dataGEO\_KIRC)  
GSE15641\_annot <- as.data.frame(GSE15641\_annot)  
GSE15641\_annot <- subset(GSE15641\_annot,  
 select = c("ID","Gene.symbol"))  
  
  
dataGEO\_samples <- pData(dataGEO\_KIRC)  
dataGEO\_samples <- as.data.frame(dataGEO\_samples)  
dataGEO\_samples <- subset(dataGEO\_samples,  
 select = c("geo\_accession","source\_name\_ch1"))  
colnames(dataGEO\_samples)[2] <- "CellType"  
  
dataGEO\_samples <- dataGEO\_samples[dataGEO\_samples$CellType %in% "cancerous human kidney tissue, clear cell RCC",]  
  
GSE15641\_merge <- merge(x = GSE15641\_annot,  
 y = GSE15641\_non\_norm,  
 by.x = "ID",  
 by.y = "ILMN")  
  
  
GSE15641\_merge <- GSE15641\_merge[order(GSE15641\_merge$IDmean,decreasing = TRUE),]  
GSE15641\_merge <- GSE15641\_merge[!duplicated(GSE15641\_merge$Gene.symbol),]  
GSE15641\_Matrix <- GSE15641\_merge  
rownames(GSE15641\_Matrix) <- GSE15641\_Matrix$Gene.symbol  
  
GSE15641\_Matrix <- GSE15641\_Matrix[,dataGEO\_samples$geo\_accession]  
  
dataImmuneSubtype <- TCGAanalyze\_ImmuneSubtypes(ImmuneMW = ImmuneMW,  
 dataGE = GSE15641\_Matrix)

## Oncogenic processes and driver genes using MoonlightR

To include some examples from <http://bioconductor.org/packages/release/bioc/vignettes/MoonlightR/inst/doc/Moonlight.html>

And some figures from <https://www.sciencedirect.com/science/article/pii/S0092867418303131>

# Session Information

sessionInfo()

## R version 3.6.0 (2019-04-26)  
## Platform: x86\_64-apple-darwin15.6.0 (64-bit)  
## Running under: macOS High Sierra 10.13.5  
##   
## Matrix products: default  
## BLAS: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRblas.0.dylib  
## LAPACK: /Library/Frameworks/R.framework/Versions/3.6/Resources/lib/libRlapack.dylib  
##   
## locale:  
## [1] en\_US.UTF-8/en\_US.UTF-8/en\_US.UTF-8/C/en\_US.UTF-8/en\_US.UTF-8  
##   
## attached base packages:  
## [1] grid parallel stats4 stats graphics grDevices utils   
## [8] datasets methods base   
##   
## other attached packages:  
## [1] png\_0.1-7 TCGAbiolinks\_2.13.1   
## [3] dplyr\_0.8.1 SummarizedExperiment\_1.14.0  
## [5] DelayedArray\_0.10.0 BiocParallel\_1.18.0   
## [7] matrixStats\_0.54.0 Biobase\_2.44.0   
## [9] GenomicRanges\_1.36.0 GenomeInfoDb\_1.20.0   
## [11] IRanges\_2.18.0 S4Vectors\_0.22.0   
## [13] BiocGenerics\_0.30.0   
##   
## loaded via a namespace (and not attached):  
## [1] colorspace\_1.4-1 selectr\_0.4-1   
## [3] rjson\_0.2.20 hwriter\_1.3.2   
## [5] circlize\_0.4.6 XVector\_0.24.0   
## [7] GlobalOptions\_0.1.0 clue\_0.3-57   
## [9] ggpubr\_0.2 matlab\_1.0.2   
## [11] ggrepel\_0.8.1 bit64\_0.9-7   
## [13] AnnotationDbi\_1.46.0 xml2\_1.2.0   
## [15] codetools\_0.2-16 splines\_3.6.0   
## [17] R.methodsS3\_1.7.1 doParallel\_1.0.14   
## [19] DESeq\_1.36.0 geneplotter\_1.62.0   
## [21] knitr\_1.23 jsonlite\_1.6   
## [23] Rsamtools\_2.0.0 km.ci\_0.5-2   
## [25] broom\_0.5.2 annotate\_1.62.0   
## [27] cluster\_2.0.8 R.oo\_1.22.0   
## [29] readr\_1.3.1 compiler\_3.6.0   
## [31] httr\_1.4.0 backports\_1.1.4   
## [33] assertthat\_0.2.1 Matrix\_1.2-17   
## [35] lazyeval\_0.2.2 limma\_3.40.0   
## [37] htmltools\_0.3.6 prettyunits\_1.0.2   
## [39] tools\_3.6.0 gtable\_0.3.0   
## [41] glue\_1.3.1 GenomeInfoDbData\_1.2.1   
## [43] ggthemes\_4.2.0 ShortRead\_1.42.0   
## [45] Rcpp\_1.0.1 Biostrings\_2.52.0   
## [47] nlme\_3.1-139 rtracklayer\_1.44.0   
## [49] iterators\_1.0.10 xfun\_0.7   
## [51] stringr\_1.4.0 rvest\_0.3.4   
## [53] XML\_3.98-1.19 edgeR\_3.26.1   
## [55] zoo\_1.8-5 zlibbioc\_1.30.0   
## [57] scales\_1.0.0 aroma.light\_3.14.0   
## [59] hms\_0.4.2 RColorBrewer\_1.1-2   
## [61] ComplexHeatmap\_2.0.0 yaml\_2.2.0   
## [63] memoise\_1.1.0 gridExtra\_2.3   
## [65] KMsurv\_0.1-5 ggplot2\_3.1.1   
## [67] downloader\_0.4 biomaRt\_2.40.0   
## [69] latticeExtra\_0.6-28 stringi\_1.4.3   
## [71] RSQLite\_2.1.1 highr\_0.8   
## [73] genefilter\_1.66.0 foreach\_1.4.4   
## [75] GenomicFeatures\_1.36.0 shape\_1.4.4   
## [77] rlang\_0.3.4 pkgconfig\_2.0.2   
## [79] bitops\_1.0-6 evaluate\_0.13   
## [81] lattice\_0.20-38 purrr\_0.3.2   
## [83] cmprsk\_2.2-7 GenomicAlignments\_1.20.0   
## [85] bit\_1.1-14 tidyselect\_0.2.5   
## [87] plyr\_1.8.4 magrittr\_1.5   
## [89] R6\_2.4.0 generics\_0.0.2   
## [91] DBI\_1.0.0 mgcv\_1.8-28   
## [93] pillar\_1.4.0 survival\_2.44-1.1   
## [95] RCurl\_1.95-4.12 tibble\_2.1.1   
## [97] EDASeq\_2.18.0 crayon\_1.3.4   
## [99] survMisc\_0.5.5 rmarkdown\_1.12   
## [101] GetoptLong\_0.1.7 progress\_1.2.2   
## [103] locfit\_1.5-9.1 sva\_3.32.0   
## [105] data.table\_1.12.2 blob\_1.1.1   
## [107] ConsensusClusterPlus\_1.48.0 digest\_0.6.18   
## [109] xtable\_1.8-4 tidyr\_0.8.3   
## [111] R.utils\_2.8.0 munsell\_0.5.0   
## [113] survminer\_0.4.3