Tumors that respond poorly to bevacizumab show upregulation of angiogenesis genes.

Roshan Lodha

27 April, 2023

```
knitr::opts_chunk$set(warning = FALSE, # turn off warnings
                      message = FALSE,
                      results = 'hide') # hide console output
knitr::opts_chunk$set(fig.width = 10, fig.height = 7) # set figure height and width
```

todo: * add SOX10 images * look into EGR1 expression in blood cancers * look into 3D vasculature as a predictor of response to bevacizumab * look into IDH1 data

Loading packages and tools for bulk RNA-sequencing analysis

Load packages

```
for (package in c('BiocManager', 'tidyverse', 'matrixStats', 'cowplot', 'DT', 'plotly', 'gt', 'ggrepel'
  if (!require(package, character.only = T, quietly = T)) {
    install.packages(package,
      repos = "http://cran.us.r-project.org")
    library(package, character.only = T)
  }
}

bio_pkgs <- c("biomaRt", "tximport", "ensembldb", "EnsDb.Hsapiens.v86", "edgeR", "DESeq2", "limma", "ap
#BiocManager::install(bio_pkgs)
invisible(lapply(bio_pkgs, function (x) library(x, character.only = T)))
```

Load MART, design matrix, and counts

```
design <- read_tsv("input/prefilterstudydesign.txt")
sampleLabels <- design$sample
group <- factor(design$group)

gbmexpr <- read_csv("input/prefiltergbmexpr.csv")[2: 13] #pre-filtered
```

Preprocessing raw counts data

```
gbmexpr.matrix <- as.matrix(gbmexpr[, -1])
rownames(gbmexpr.matrix) <- unlist(gbmexpr[, 1])
myDGEList <- DGEList(gbmexpr.matrix)
myDGEList.filtered.norm <- calcNormFactors(myDGEList, method = "TMM") #normalize using TMM
log2.tpm.filtered.norm <- log2(gbmexpr.matrix + 1)
log2.tpm.filtered.norm.df <- as_tibble(log2.tpm.filtered.norm, rownames = "geneID")
```

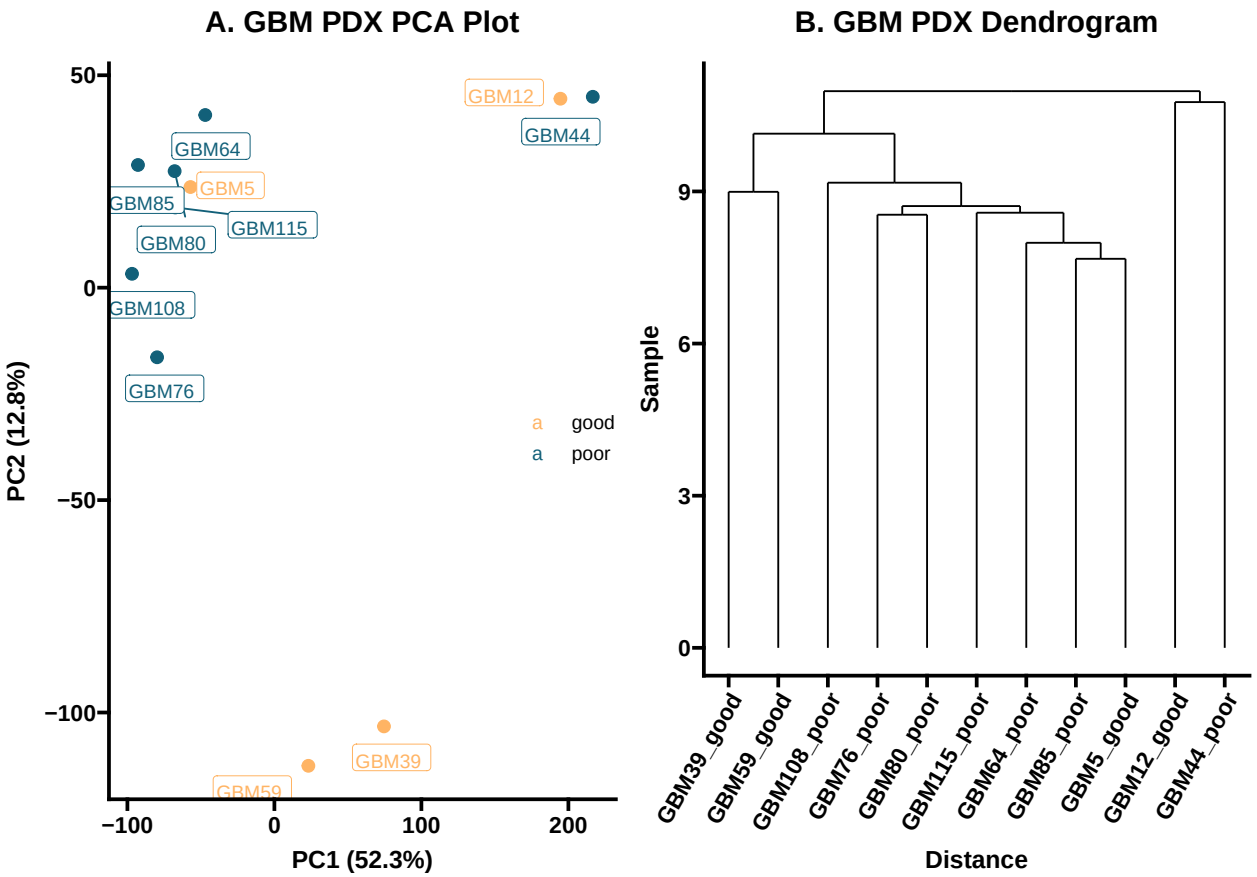
Preprocessed PCA plot and clustering dendrogram

```
distance <- dist(t(log2.tpm.filtered.norm), method = "maximum")
clusters <- hclust(distance, method = "average")
den <- ggdendrogram(clusters) +
  labs(title = "B. GBM PDX Dendrogram") +
  xlab("Distance") +
  ylab("Sample") +
  theme_prism() +
  theme(axis.text.x = element_text(angle = 60, vjust = 1, hjust = 1))
##ggsave(path = "./plots/", filename = "dendrogram.png", plot = den, height = 5, width = 5)

sampleLabels <- substr(sampleLabels, 1, nchar(sampleLabels) - 5)
pca.res <- prcomp(t(log2.tpm.filtered.norm), scale. = F, retx = T)
pc.var <- pca.res$sdev ^ 2
pc.per <- round(pc.var / sum(pc.var) * 100, 1)
pca.res.df <- as_tibble(pca.res$x)
pca.plot <- ggplot(pca.res.df) +
  aes(x = PC1, y = PC2, label = sampleLabels, color = group) +
  geom_point(size = 3) +
  geom_label_repel(aes(label = sampleLabels), hjust = 0, vjust = 0) +
  scale_color_manual(values = c("#ffb464", "#126079")) +
  xlab(paste0("PC1 (", pc.per[1], "%", ")")) +
  ylab(paste0("PC2 (", pc.per[2], "%", ")")) +
  labs(title = "A. GBM PDX PCA Plot") +
  theme_prism() +
  theme(legend.position = c(0.9, 0.5))
##ggsave(path = "./plots/", filename = "prefilterpca.png", plot = pca.plot, height = 5, width = 7)
```

Supplemental Figure 1

```
sf1 <- ggarrange(pca.plot, den, ncol = 2, nrow = 1)
sf1
```




```
#ggexport(sf1, filename = "./plots/supplementalfigure1.png", width = 1000, height = 500)
```

Dimensionality reduction analysis

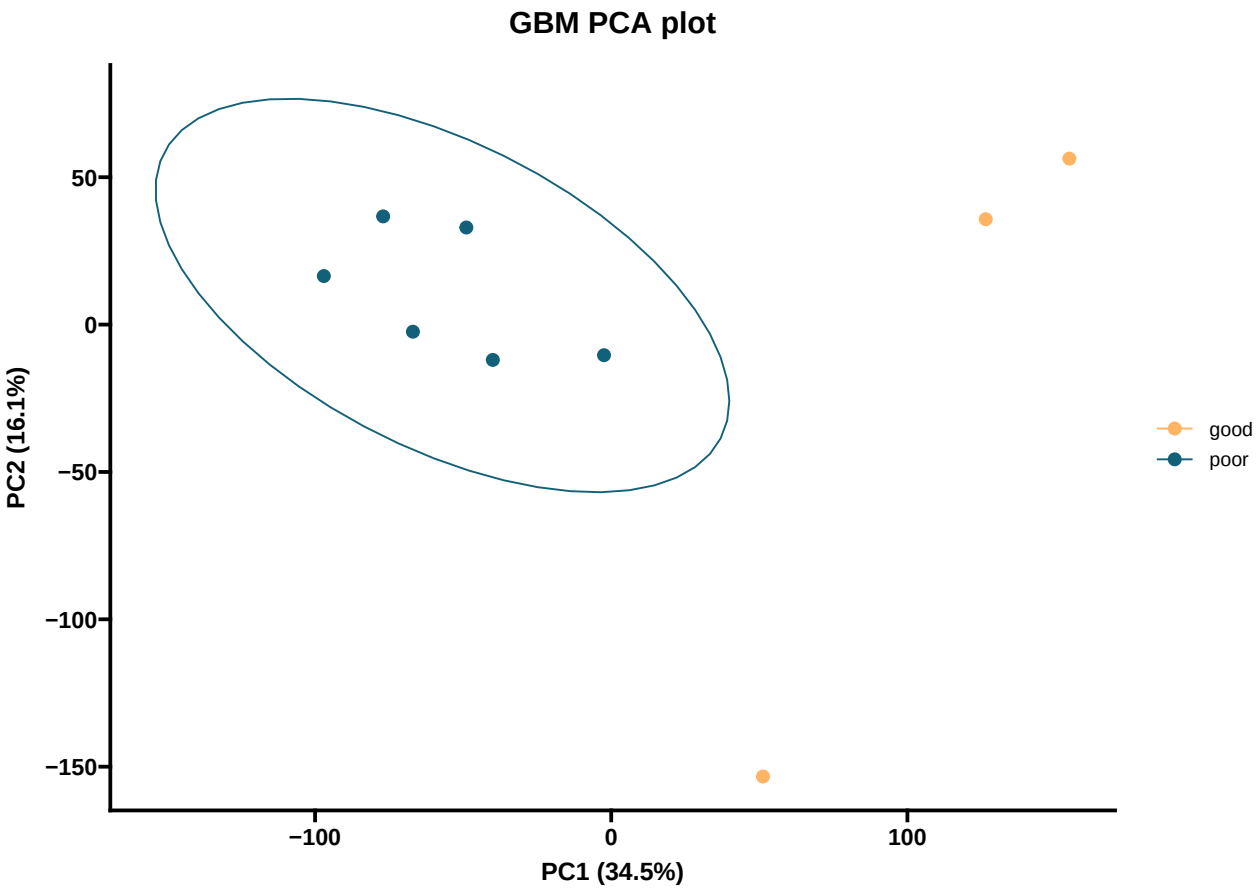
Create counts matrix

```
design <- read_tsv("./input/studydesign.txt")
sampleLabels <- design$sample
group <- factor(design$group)
mm <- model.matrix(~0 + group)

counts <- read.table(file = "./input/counts.tabular", header = TRUE, sep = "\t")
colnames(counts) <- c("geneID", sampleLabels)
counts$gene <- getSYMBOL(as.character(counts$geneID), data = 'org.Hs.eg')
counts <- counts %>% select(gene, everything())
counts <- counts[rowSums(counts <= 0) <= 3, ] %>% drop_na() #filter: at most 3 zeros
counts.matrix <- as.matrix(counts[3: 11])
rownames(counts.matrix) <- counts$gene
```

PCA

```
pca.res <- prcomp(t(log(counts.matrix + 1)), scale. = F, retx = T)
pc.var <- pca.res$sdev ^ 2
pc.per <- round(pc.var / sum(pc.var) * 100, 1)
pca.res.df <- as_tibble(pca.res$x)
pca.plot <- ggplot(pca.res.df) +
  aes(x = PC1, y = PC2, label = sampleLabels, color = group) +
  geom_point(size = 3) +
  #geom_text(aes(label = sampleLabels), hjust = 0, vjust = 0) +
  #stat_ellipse() +
  scale_color_manual(values = c("#ffb464", "#126079")) +
  xlab(paste0("PC1 (", pc.per[1], "%", ")")) +
  ylab(paste0("PC2 (", pc.per[2], "%", ")")) +
  labs(title = "GBM PCA plot") +
  # caption = paste0("produced on ", Sys.time())) +
  stat_ellipse() +
  theme_prism()
# #ggsave(path = "./plots/", filename = "postfilterpca.png", plot = pca.plot, height = 5, width = 7)
pca.plot
```



We choose not to perform UMAP due to limited sample size and use of PDX tumors.

Differential gene expression analysis

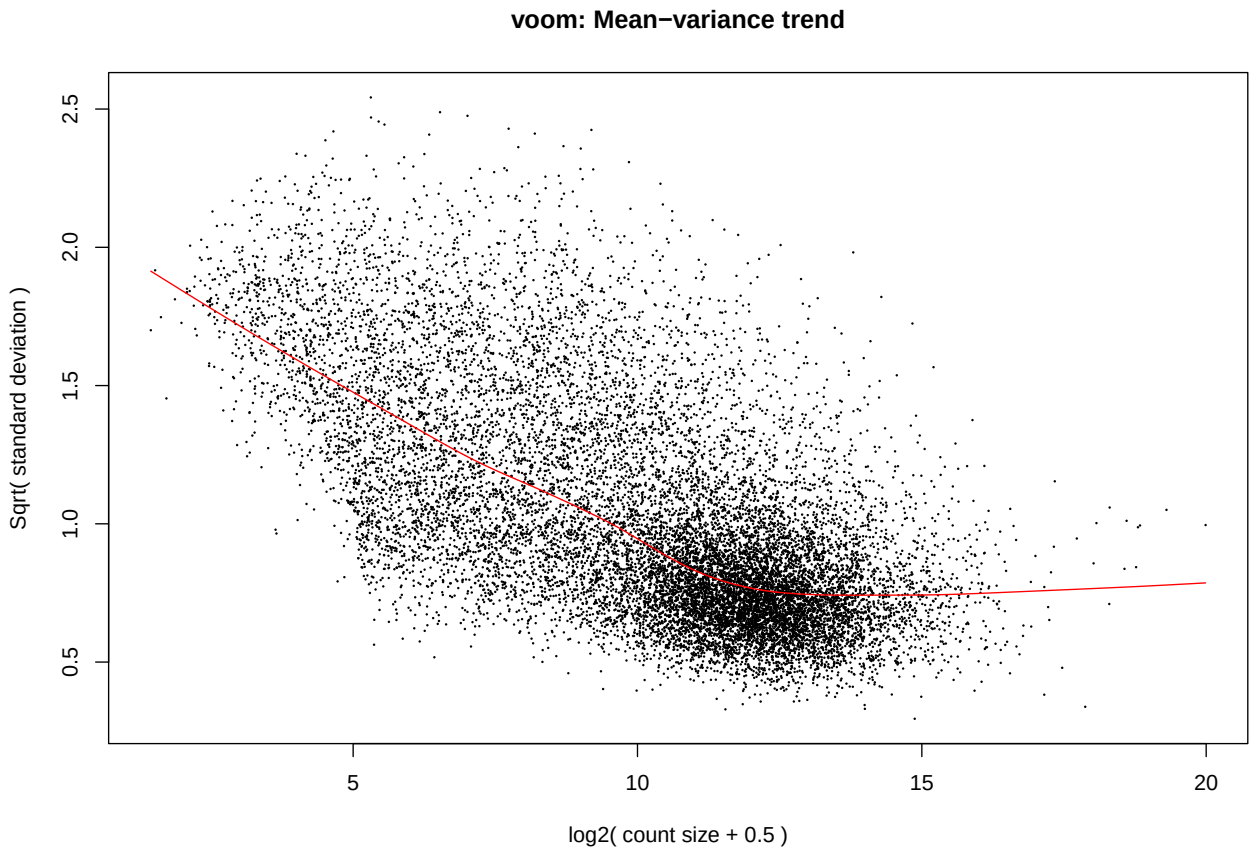
Processing counts data

```
mart = useEnsembl(biomart='ensembl', dataset = "hsapiens_gene_ensembl", mirror = "useast")
#mart <- useMart("ENSEMBL_MART_ENSEMBL") #coding genes for cleaning
#mart <- useDataset("hsapiens_gene_ensembl", mart)

coding_genes <- getBM(attributes = c("hgnc_symbol"),
  filters = c("biotype"),
  values = list(biotype = "protein_coding"),
  mart = mart)$hgnc_symbol

rownames(counts.matrix) <- counts$geneID
dds <- DESeqDataSetFromMatrix(countData = round(counts.matrix),
  colData = design,
  design = ~group)

keep <- rowSums(counts(dds)) >= 350 #determined via hyperparameter exploration
dds <- dds[keep, ]
dds <- DESeq(dds)
dds <- estimateSizeFactors(dds)
deseqvoom <- voom(counts(dds, normalized = TRUE), mm, plot = T)
```



Differential gene expression analysis

```
res <- results(dds)
res <- lfcShrink(dds, coef = "group_poor_vs_good", type = "ashr")
```

```
## using 'ashr' for LFC shrinkage. If used in published research, please cite:
##     Stephens, M. (2016) False discovery rates: a new deal. Biostatistics, 18:2.
##     https://doi.org/10.1093/biostatistics/kxw041
```

```
deseq <- as.data.frame(res) %>% drop_na() %>% arrange(padj) %>% arrange(desc(abs(log2FoldChange)))
deseq$gene <- getSYMBOL(rownames(deseq), data = 'org.Hs.eg')
deseq <- dplyr::filter(deseq, gene %in% coding_genes)
deseq <- deseq %>%
  mutate(enrichment = case_when(
    (padj < 0.05) & (log2FoldChange > 1) ~"poor",
    (padj < 0.05) & (log2FoldChange <- 1) ~"good"))
deseq$enrichment[is.na(deseq$enrichment)] <- "none"
rownames(deseq) <- deseq$gene
#write_csv(deseq, "./tables/dge.csv")
head(deseq, 10)
```

```
##          baseMean log2FoldChange     lfcSE       pvalue         padj   gene
## MXRA5   1759.7796      10.873968 2.0639850 1.378951e-10 1.282932e-08  MXRA5
## FIGNL2   436.4693      10.618612 1.3501858 8.036464e-18 2.708289e-15 FIGNL2
## DPP10   2903.6767      10.550579 2.3641414 8.834061e-09 5.113303e-07  DPP10
## SHD     3326.0768      10.543546 1.0904000 1.100572e-24 1.192155e-21    SHD
## IGLON5  2932.8203      10.406657 1.5991788 1.909022e-13 3.272259e-11 IGLON5
## SYT13    925.2397      10.358431 1.9234684 1.523300e-10 1.408589e-08  SYT13
## NCAN   38401.3357      10.214752 0.6536549 2.376496e-57 3.603955e-53   NCAN
## SIX6     646.6990      10.041625 1.8530796 1.971098e-10 1.800705e-08   SIX6
## SCN3B   1313.9016       9.852600 1.3113978 3.510798e-16 9.859491e-14  SCN3B
## VGF    27367.6704       9.781803 1.4479717 8.682615e-14 1.586408e-11    VGF
##        enrichment
## MXRA5        poor
## FIGNL2       poor
## DPP10        poor
## SHD          poor
## IGLON5       poor
## SYT13        poor
## NCAN         poor
## SIX6         poor
## SCN3B        poor
## VGF          poor
```

```
dgeplot <- ggplot(deseq) +
  aes(y = -log10(padj), x = log2FoldChange, colour = enrichment) +
  scale_color_manual(values = c("good" = "#126079", "none" = "grey", "poor" = "#ffb464")) +
  geom_point(size = 1.5, alpha = 0.25) +
  #facet_zoom(xlim = c(9, 11), zoom.size = 1) +
  geom_rect(mapping = aes(xmin = 9, xmax = 11, ymin = 2, ymax = 54), alpha = 0, color = 'black') +
  geom_text_repel(size = 4, data = head(deseq, 20), aes(label = gene), max.overlaps = Inf, colour = "bla
```

```
  #geom_text_repel(size = 3, data = subset(deseq, gene == 'EGR1'), aes(label = gene)) +
  geom_hline(yintercept = -log10(0.05), linetype = "longdash", colour = "black", size = .5) +
  geom_text(aes(-8, -log10(0.05), label = "p = 0.05", vjust = 1), colour = "black") +
  labs(title = "A. Differential RNA Expression") +
  #subtitle = "Positive logFC indicates upregulation in poor Bevacizumab responders") +
  xlab("Fold Change (log2)") +
  ylab("Significance (log10)") +
  theme_prism() +
  theme(legend.position = c(0.15, 0.8))
##ggsave(filename = "./plots/deseq.png", plot = dgeplot, height = 6, width = 6)
#dgeplot
```

Collagen Gene Set Analysis

```
collagen_genes <- c("COL1A1", "COL1A2", "COL2A1", "COL3A1", "COL4A1", "COL4A2", "COL4A3", "COL4A4", "COL

collagen_genes <- deseq %>% filter(gene %in% collagen_genes) %>% filter(enrichment != "none")
collagen_genes <- collagen_genes$gene

collagen <- counts %>%
  filter(gene %in% collagen_genes) %>%
  dplyr::select(-geneID)
rownames(collagen) <- collagen$gene
collagen <- collagen %>%
  dplyr::select(-gene)
collagen <- as.data.frame(t(collagen))
collagen$sample <- rownames(collagen)
rownames(collagen) <- NULL
collagen$group <- gsub("GBM[0-9]*_", "", collagen$sample)
collagen <- collagen %>% pivot_longer(cols = -c("sample", "group"),
  names_to = "gene",
  values_to = "expression")

collagen$log.expr <- log(collagen$expression + 1)
collagen$sample <- substr(collagen$sample,
  1,
  nchar(collagen$sample) - 5)

collagen_plot <- ggplot(data = collagen, mapping = aes(x = sample, y = gene, fill = log.expr)) +
  geom_tile() +
  scale_fill_gradient(high = "#ffb464", low = "#126079") +
  scale_colour_prism(palette = "colors") +
  xlab(label = "Patient Derived Xenograft") + # Add a nicer x-axis title
  ggtitle("D. Collagen Genes RNA Expression") +
  facet_grid(~group,
    switch = "x", scales = "free_x", space = "free_x") +
  #labs(color = "Your title here") +
  theme_prism() +
  theme(axis.title.y = element_blank(),
    axis.text.x = element_text(angle = 60, vjust = 0.7),
    legend.position = "bottom")
```

```
##ggsave(filename = "./plots/collagen.png", plot = collagen_plot, height = 6, width = 6)
#collagen_plot
```

Angiogenesis Gene Set Analysis

```
blood_vessel <- scan("./input/blood_vessel_geneset.txt", character(), quote = "")
blood_vessel <- deseq %>% filter((gene %in% blood_vessel) &
  (enrichment != "none") &
  (abs(log2FoldChange) > 4))

# angiogenesis heatmap
goi <- blood_vessel$gene
b_v_heatmap <- counts %>% filter(gene %in% goi) %>% dplyr::select(-geneID)
rownames(b_v_heatmap) <- b_v_heatmap$gene
b_v_heatmap <- b_v_heatmap %>% dplyr::select(-gene)
b_v_heatmap <- as.data.frame(t(b_v_heatmap))
b_v_heatmap$sample <- rownames(b_v_heatmap)
rownames(b_v_heatmap) <- NULL
b_v_heatmap$group <- gsub("GBM[0-9]*_", "", b_v_heatmap$sample)
b_v_heatmap <- b_v_heatmap %>% pivot_longer(cols = -c("sample", "group"),
  names_to = "gene",
  values_to = "expression")

b_v_heatmap$log.expr <- log(b_v_heatmap$expression + 1)
b_v_heatmap$sample <- substr(b_v_heatmap$sample, 1, nchar(b_v_heatmap$sample) - 5)

b_v_heatmap.plot <- ggplot(data = b_v_heatmap, mapping = aes(x = sample, y = gene, fill = log.expr)) +
  geom_tile() +
  scale_fill_gradient(high = "#ffb464", low = "#126079") +
  scale_colour_prism(palette = "colors") +
  xlab(label = "Patient Derived Xenograft") + # Add a nicer x - axis title
  ggtitle("C. Angiogenic Gene RNA Expression") +
  facet_grid(~group,
    switch = "x", scales = "free_x", space = "free_x") +
  #labs(color = "Your title here") +
  theme_prism() +
  theme(axis.title.y = element_blank(),
    axis.text.x = element_text(angle = 60, vjust = 0.7),
    legend.position = "bottom")
#b_v_heatmap.plot
##ggsave(filename = "./plots/angiogenesis.png",
```

Gene set enrichment anlaysis (GSEA)

GSEA data cleaning and KEGG pathway analysis

```
hs_gsea <- msigdbr(species = "Homo sapiens")
hs_gsea %>% dplyr::distinct(gs_cat, gs_subcat) %>% dplyr::arrange(gs_cat, gs_subcat)
```

```r
hs_gsea_h <- msigdbr(species = "Homo sapiens",
    category = "H") %>%
  dplyr::select(gs_name, gene_symbol)
hs_gsea_kegg <- msigdbr(species = "Homo sapiens",
    category = "C2",
    subcategory = "CP:KEGG") %>%
  dplyr::select(gs_name, gene_symbol)

deseq.GSEA.select <- dplyr::select(deseq, gene, log2FoldChange, padj)
deseq.gsea <- abs(deseq.GSEA.select$log2FoldChange) / deseq.GSEA.select$log2FoldChange * -log10(deseq.GS
names(deseq.gsea) <- as.character(deseq.GSEA.select$gene)
deseq.gsea <- sort(deseq.gsea, decreasing = TRUE)
deseq.gsea.res <- GSEA(deseq.gsea, pvalueCutoff = 1, TERM2GENE = hs_gsea_kegg, verbose = FALSE)
deseq.GSEA.df <- as_tibble(deseq.gsea.res @result)
deseq.GSEA.df <- deseq.GSEA.df %>%
  mutate(phenotype = case_when(
    (NES > 0) & (p.adjust < 0.05) ~"poor",
    (NES < 0) & (p.adjust < 0.05) ~"good"))
deseq.GSEA.df$phenotype[is.na(deseq.GSEA.df$phenotype)] <- "none"
deseq.GSEA.df$Description <- gsub("_", " ", deseq.GSEA.df$Description)
#write_csv(deseq.GSEA.df, "./tables/gsea_kegg.csv")
```

```
## # A tibble: 23 x 2
##    gs_cat gs_subcat
##    <chr>  <chr>
##  1 C1     ""
##  2 C2     "CGP"
##  3 C2     "CP"
##  4 C2     "CP:BIOCARTA"
##  5 C2     "CP:KEGG"
##  6 C2     "CP:PID"
##  7 C2     "CP:REACTOME"
##  8 C2     "CP:WIKIPATHWAYS"
##  9 C3     "MIR:MIRDB"
## 10 C3     "MIR:MIR_Legacy"
## # ... with 13 more rows
```

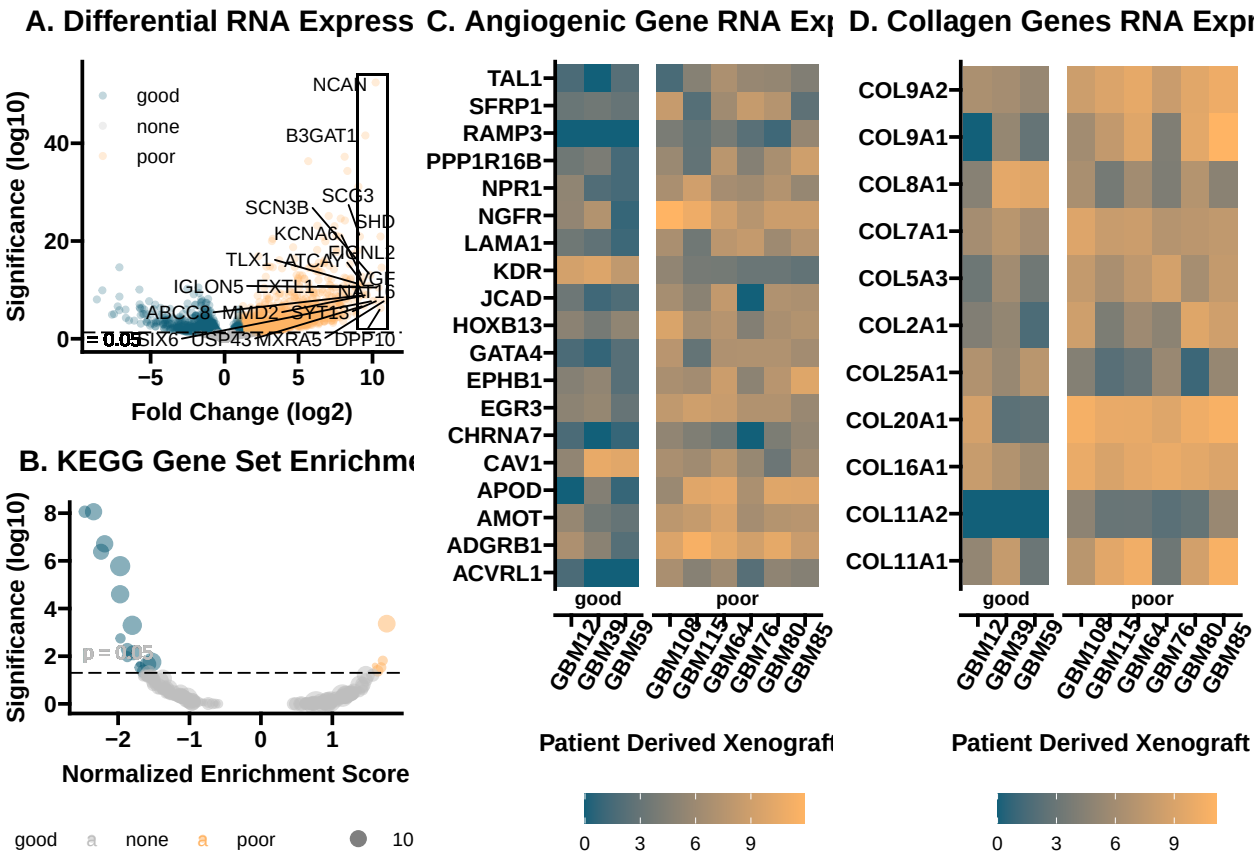
GSEA plot

```r
kegg_gsea <- ggplot(deseq.GSEA.df, aes(x = NES, y = -log10(p.adjust), color = phenotype)) +
  geom_point(aes(size = setSize), alpha = 0.5) +
  scale_color_manual(values = c("good" = "#126079", "none" = "grey", "poor" = "#ffb464")) +
  geom_text(aes(-2, -log10(0.05), label = "p = 0.05", vjust = -1)) +
  geom_hline(yintercept = -log10(0.05), linetype = "longdash", size = .5) +
  #geom_text_repel(size = 4, data = (deseq.GSEA.df %>% dplyr::filter((NES > 0) & (p.adjust < 0.05)))[1,
  #  aes(label = Description)) +
  labs(title = "B. KEGG Gene Set Enrichment") +
  # subtitle = "Positive NES indicates upregulation of gene set in poor Bevacizumab responders") +
  ylab("Significance (log10)") +
  xlab("Normalized Enrichment Score") +
  theme_prism() +
```

```
  theme(legend.position = "bottom")
##ggsave(path = "./plots/", filename = "gsea_kegg.png", plot = kegg_gsea, width = 6, height = 6)
#kegg_gsea
```

Figure 1

```
f1 <- ggarrange(ggarrange(dgeplot, kegg_gsea, ncol = 1, nrow = 2),
                b_v_heatmap.plot, collagen_plot, ncol = 3, nrow = 1)
f1
```



Immunohistochemical staining analysis

IHC-quantified protein experssion of EGR1 (good vs poor)

```
egr1.poor.good <- read.csv("./input/ihc/egr1poorgood.csv")
egr1.poor.good$fullname <- paste(egr1.poor.good$pdx_id, egr1.poor.good$animal_id)
egr1.ihc.plot <- ggplot(egr1.poor.good, aes(x = fct_rev(bev_resp), y = h_score)) +
  geom_boxplot(alpha = 0.5, fill = c("#ffb464", "#126079")) +
  geom_dotplot(aes(fill = factor(fullname)),
    binaxis = "y",
```

```
    stackdir = "center",
    dotsize = 0.5,
    binpositions = "all",
    stackgroups = TRUE) +
  theme_prism() +
  theme(
    legend.key.height = unit(10, "pt"),
    legend.title = element_text()) +
  guides(fill = guide_legend(title = "Animal ID")) +
  #stat_compare_means(label.x.npc = "left", label.y.npc = "bottom") +
  ggtitle("EGR1 Expression (Nuclear)") +
  xlab("Bevacizumab Response Group") + ylab("Expression (H Score)")
wilcox.test((egr1.poor.good %>% dplyr::filter(bev_resp == "good")) $h_score,
  (egr1.poor.good %>% dplyr::filter(bev_resp == "poor")) $h_score)
```

IHC-quantified protein experssion of EGR1 (poor vs placebo)

```
egr1.poor.placebo <- read.csv("./input/ihc/egr1poorplacebo.csv")
egr1.poor.placebo$fullname <- paste(egr1.poor.placebo$pdx_id, egr1.poor.placebo$animal_id)
poor.placebo.nuc <- ggplot(egr1.poor.placebo %>% dplyr::filter(staining_type == "nuclear"),
    aes(x = treatment, y = h_score)) +
  geom_boxplot(alpha = 0.5, fill = c("#ffb464", "#126079")) +
  geom_dotplot(aes(fill = factor(fullname)),
    binaxis = "y",
    stackdir = "center",
    dotsize = 0.5,
    binpositions = "all",
    stackgroups = TRUE) +
  theme_prism() +
  theme(
    legend.key.height = unit(10, "pt"),
    legend.title = element_text()) +
  guides(fill = guide_legend(title = "Animal ID")) +
  ggtitle("EGR1 Nuclear Expression") +
  xlab("Treatment Group") + ylab("Expression (H Score)")

poor.placebo.cyt <- ggplot(egr1.poor.placebo %>% dplyr::filter(staining_type == "cytoplasmic"),
    aes(x = treatment, y = h_score)) +
  geom_boxplot(alpha = 0.5, fill = c("#ffb464", "#126079")) +
  geom_dotplot(aes(fill = factor(fullname)),
    binaxis = "y",
    stackdir = "center",
    dotsize = 0.5,
    binpositions = "all",
    stackgroups = TRUE) +
  theme_prism() +
  theme(
    legend.key.height = unit(10, "pt"),
    legend.title = element_text()) +
  guides(fill = guide_legend(title = "Animal ID")) +
  #stat_compare_means(label.x.npc = "left", label.y.npc = "bottom") +
```

```
  ggtitle("EGR1 Cytoplasmic Expression") +
  xlab("Treatment Group") + ylab("Expression (H Score)")
```

Supplemental Figure 2

```
sf2 <- ggarrange(poor.placebo.nuc, poor.placebo.cyt, nrow = 1, ncol = 2)
sf2
```

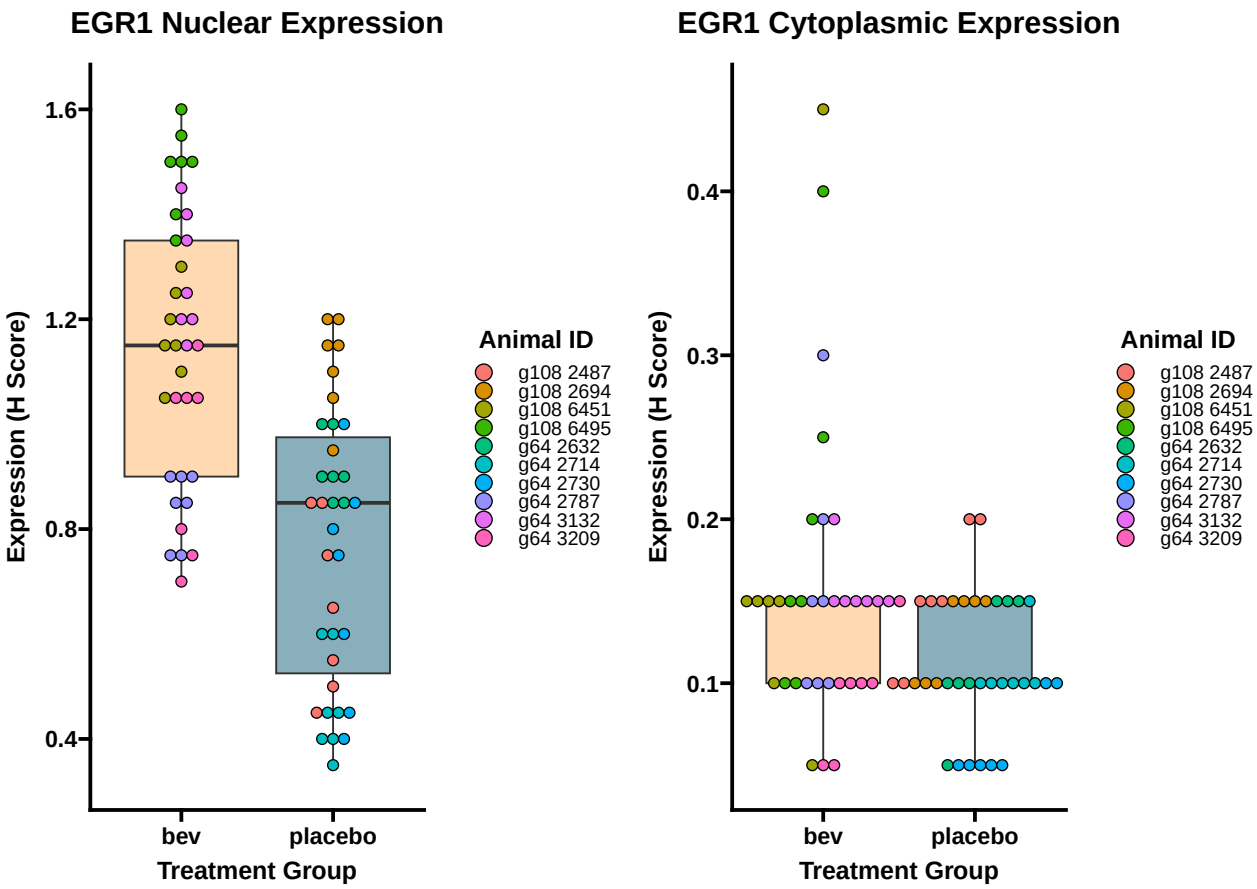
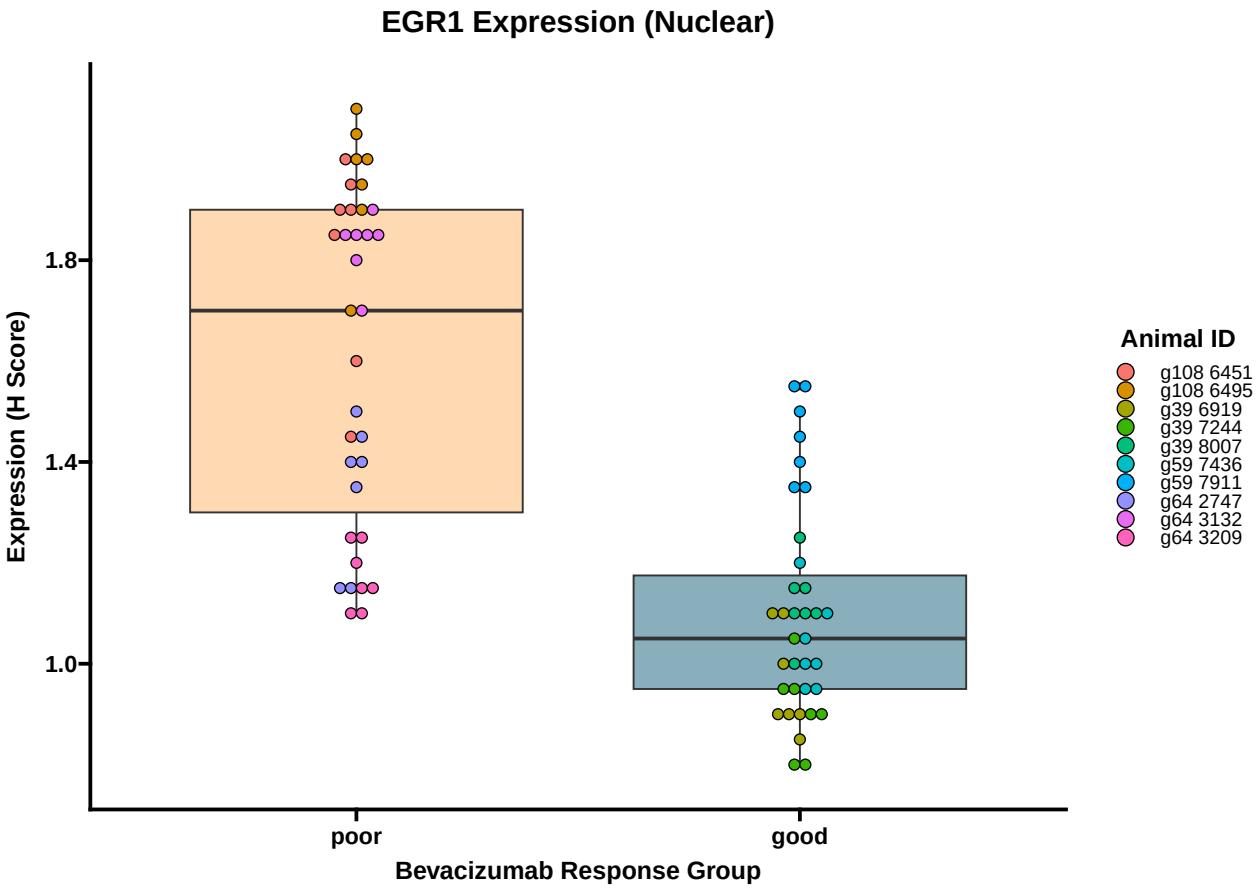


Figure 2

```
egr1.ihc.plot
```



Survival Analysis

Load packages

```
pkgs <- c("UCSCXenaTools", "dplyr", "survival", "survminer", "ggbreak", "ggprism", "svglite")
#BiocManager::install(pkgs)
invisible(lapply(pkgs, function (x) suppressMessages(library(x, character.only = T))))
```

Download TCGA glioblastoma dataset

```
gbm_cohort = XenaData %>%
  filter(XenaHostNames == "tcgaHub") %>%
  XenaScan("TCGA Glioblastoma") # microarray dataset, CNVA dataset DNA level

#download clinical data-- --
cli_query = gbm_cohort %>%
  filter(DataSubtype == "phenotype") %>% # select clinical dataset
  XenaGenerate() %>% # generate a XenaHub object
  XenaQuery() %>%
  XenaDownload()
```

```
cli = XenaPrepare(cli_query)

ge = gbm_cohort %>%
  filter(DataSubtype == "protein expression RPPA", Label == "RPPA (replicate-base normalization)")
# TODO: try AFFYmetrix

# download gene expression data
ge = gbm_cohort %>%
  filter(DataSubtype == "gene expression RNAseq", Label == "IlluminaHiSeq")
EGR1 = fetch_dense_values(host = ge$XenaHosts,
  dataset = ge$XenaDatasets,
  identifiers = "EGR1",
  use_probeMap = TRUE) %>% .[1, ]
EGR3 = fetch_dense_values(host = ge$XenaHosts,
  dataset = ge$XenaDatasets,
  identifiers = "EGR3",
  use_probeMap = TRUE) %>% .[1, ]
SOX10 = fetch_dense_values(host = ge$XenaHosts,
  dataset = ge$XenaDatasets,
  identifiers = "SOX10",
  use_probeMap = TRUE) %>% .[1, ]
RAMP3 = fetch_dense_values(host = ge$XenaHosts,
  dataset = ge$XenaDatasets,
  identifiers = "RAMP3",
  use_probeMap = TRUE) %>% .[1, ]
CHRNA7 = fetch_dense_values(host = ge$XenaHosts,
  dataset = ge$XenaDatasets,
  identifiers = "CHRNA7",
  use_probeMap = TRUE) %>% .[1, ]
```

Survival stratified by SOX10 expression

Survival stratified by EGR3 expression

Survival stratified by EGR1 expression

```
#merge-- --
merged_EGR1 = tibble(sample = names(EGR1),
    EGR1_expression = as.numeric(EGR1)) %>%
  left_join(cli$GBM_survival.txt, by = "sample") %>%
  #filter(sample_type == "Primary Tumor") %>% # Keep only 'Primary Tumor'
  select(sample, EGR1_expression, OS.time, OS) %>%
  rename(time = OS.time,
    status = OS)

fit_EGR1 = coxph(Surv(time, status) ~EGR1_expression, data = merged_EGR1)
fit_EGR1

merged_EGR1 = merged_EGR1 %>%
  mutate(group = case_when(
    EGR1_expression > quantile(EGR1_expression, 0.9) ~'High',
```

```
    (EGR1_expression < quantile(EGR1_expression, 0.9) &
      EGR1_expression > quantile(EGR1_expression, 0.1)) ~'Normal',
    EGR1_expression < quantile(EGR1_expression, 0.1) ~'Low',
    TRUE~NA_character_
  )) %>%
  mutate(z = (EGR1_expression - mean(EGR1_expression)) / sd(EGR1_expression)) %>%
  mutate(group = case_when(
    z > 1.5~'High',
    z < -1.5~'Low',
    (z < 1.5) & (z > -1.5) ~'Normal',
    TRUE~NA_character_
  ))
fit_EGR1 = survfit(Surv(time, status) ~group,
  data = merged_EGR1 %>% dplyr::filter(group != "Low"))
EGR1_plot <- ggsurvplot(fit_EGR1,
  pval = TRUE,
  pval.coord = c(600, 0.5),
  #xlim = c(0, 1000),
  palette = c("#ffb464", "#126079"),
  #conf.int = TRUE,
  #pval = TRUE,
  risk.table = TRUE,
  risk.table.col = "strata",
  legend.labs = c("High Expression", "Normal Expression"),
  surv.median.line = "hv",
  break.time.by = 250,
    ggtheme = theme_prism(),
    legend = c(0.7, 0.8),
    title = "EGR1-expression Stratisfied Survival Plot")
##ggsave("../plots/survival/EGR1survival.svg", plot = print(EGR1_plot), height = 6, width = 6)
```

Survival stratified by RAMP3 expression

Survival stratified by CHRNA7 expression

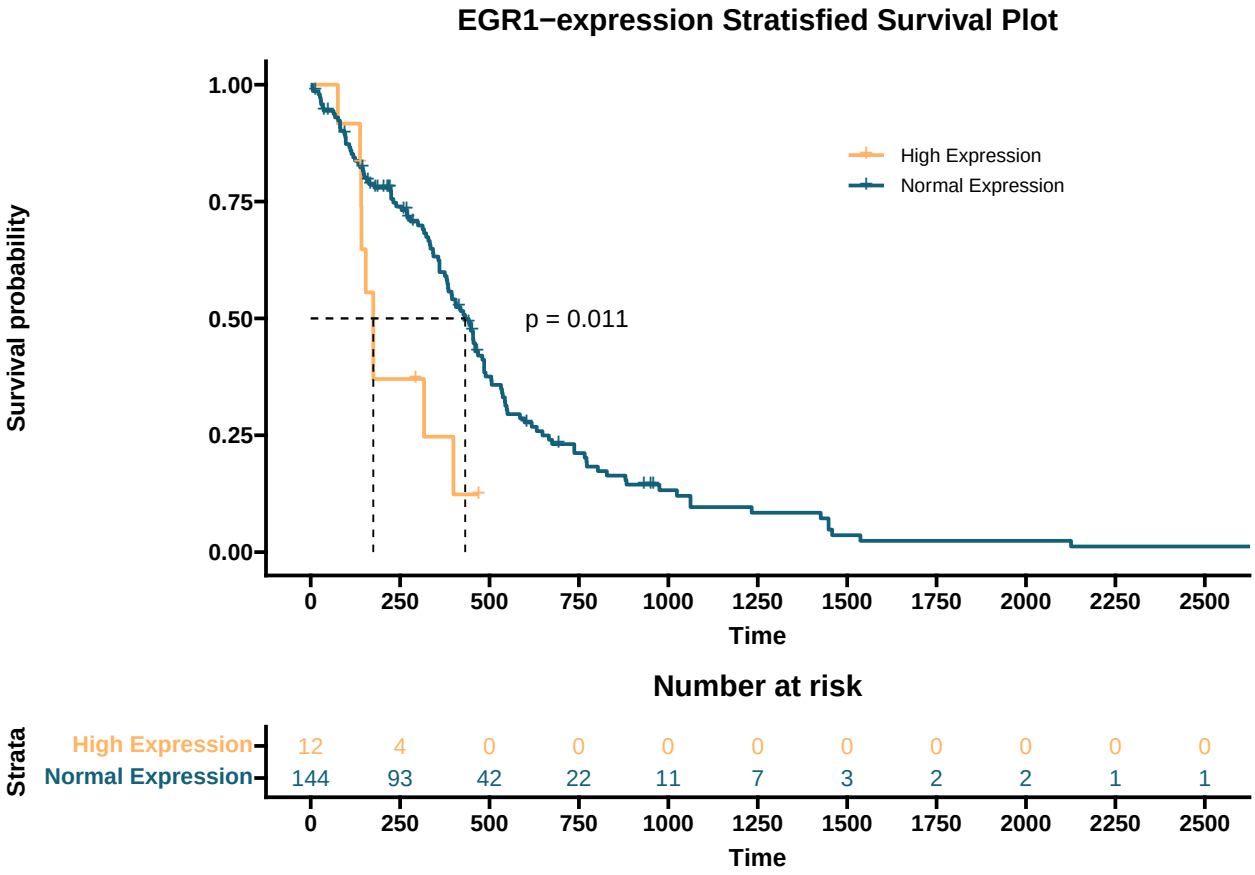
Survival stratified by EGR1 and SOX10 expression

Survival stratified by EGR1 and CHRNA7 expression

Survival stratified by EGR1 and RAMP3 expression

Figure 3

```
EGR1_plot
```


EGR1−expression Stratisfied Survival Plot
Survival probability
1.00
0.75
0.50
0.25
0.00
High Expression
Normal Expression
p = 0.011
0
250
500
750
1000
1250
1500
1750
2000
2250
2500
Time
Number at risk
Strata
High Expression 12 4 0 0 0 0 0 0 0 0 0
Normal Expression 144 93 42 22 11 7 3 2 2 1 1
Time