Useful Plots for Bioinformatics

By Xiangming Huang

SURVIVAL CURVE

- **Basic definitions**
  1. Survival time: the time from "response to treatment" to the occurrence of the event of interest.
  2. Censored observation: An observation where the event of interest is not observed within the study period
  3. Survival probability: the probability that an individual survives from the time origin to a specific future time.
  4. Hazard: the probability that an individual who is under observation at a time t has an event at that time.
  5. Kaplan-Meier survival estimate: a non-parametric method used to estimate the survival probability from observed survival times.
- **Library setup**

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

- **Analyze data with survfit()**

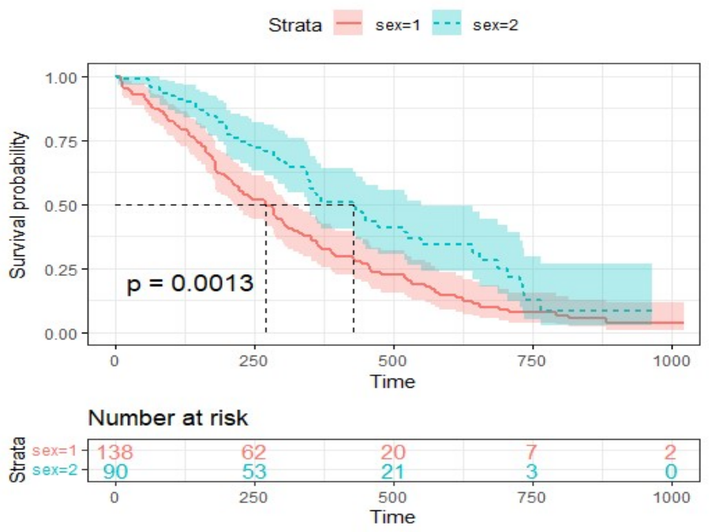
```
data("lung")
fit <- survfit(Surv(time, status) ~ sex, data = lung)
```

- **Commonly used statistical functions**

```
fun = "log" # log transformation of the survivor function
fun = "event" # cumulative events
fun = "cumhaz" # cumulative hazard function
```

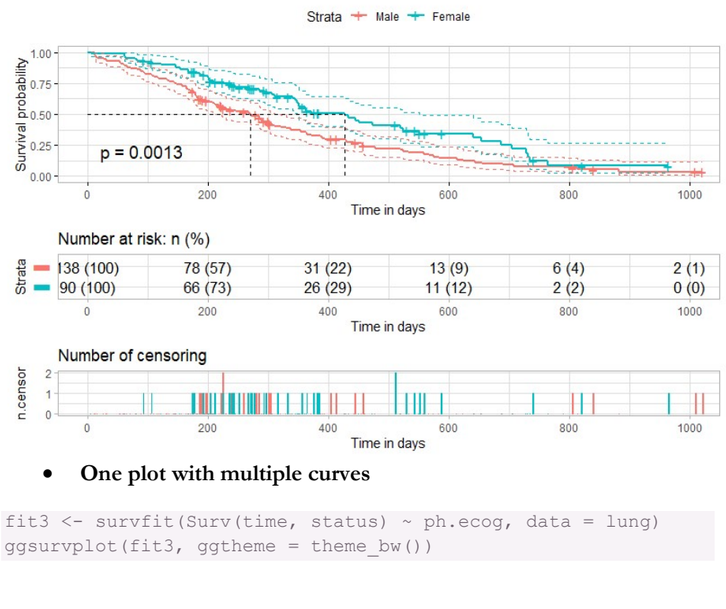
- **Survival curve with risk table and p-value**

```
 # Change color, linetype by strata, risk.table color by
strata
ggsurvplot(fit, # survfit object with calculated statistics.
           pval = TRUE,# show p-value of log-rank test.
           conf.int = TRUE,# show confidence intervals.
           censor = FALSE,# Turn of censor tick
           risk.table = TRUE,# Add risk table
           risk.table.col = "strata",# Change color by groups
           linetype = "strata",# Change line type by groups
           surv.median.line = "hv",# Specify median survival
           ggtheme = theme_bw())# Change ggplot2 theme
```



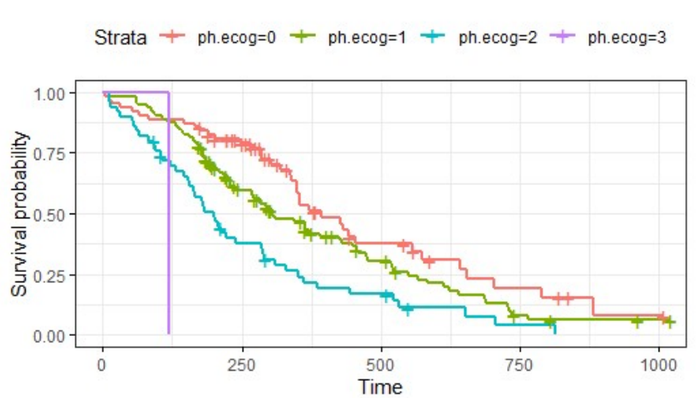
- **Survival Curve with risk table, p-value, and censor plot**

```
ggsurvplot(
  fit,
  pval = TRUE,
  conf.int = TRUE,
  conf.int.style = "step",
  xlab = "Time in days",
  break.time.by = 200,
  ggtheme = theme_light(),
  risk.table = "abs_pct",
  risk.table.y.text.col = T,
  risk.table.y.text = FALSE,
  ncensor.plot = TRUE,
  surv.median.line = "hv",
  legend.labs = c("Male","Female")
)
```



- **One plot with multiple curves**

```
fit3 <- survfit(Surv(time, status) ~ ph.ecog, data = lung)
ggsurvplot(fit3, ggtheme = theme_bw())
```



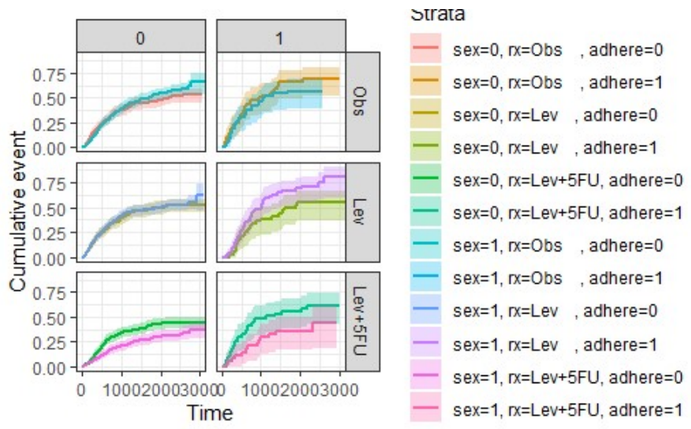
- **Facet multiple curves**

```
fit4 <- survfit(Surv(time,status) ~ sex + rx + adhere, data =
colon)
ggsurvplot(fit4,
           fun = "event",
           conf.int = TRUE,
           censor = FALSE, # Turn off censor
           ggtheme = theme_bw())$plot +
```

Useful Plots for Bioinformatics

By Xiangming Huang

```
theme_bw() + theme(legend.position = "right") +
facet_grid(rx ~ adhere)
```



HEATMAP

- **Basics**

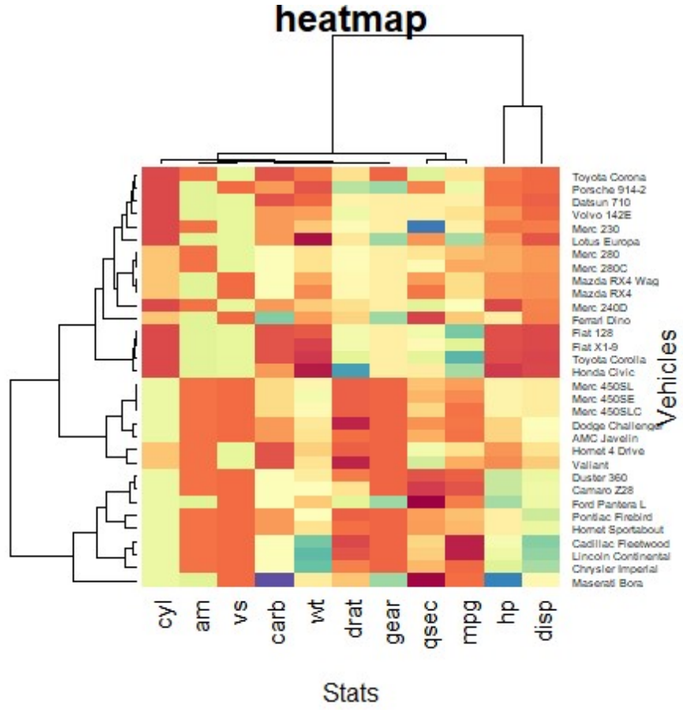
Input	matrix
DataFrame to matrix	`as.matrix()`
Normalization	`scale()`

- **Coloring**

```
library(RColorBrewer)
col1 <- colorRampPalette(brewer.pal(11,"PuOr"))(256)
col2 <- colorRampPalette(brewer.pal(11, "Spectral"))(256)
```

- **Standard heatmap**

```
heatmap(df, #matrix with numeric value only.
        Colv = NULL, #no dendrogram and reordering along the
column.
        Rowv = NULL, #no dendrogram and reordering along the
row.
        scale = "column", #normalize each column
        col = col2,#customize color palette
        xlab = "Stats",#lab of x-axis
        ylab = "Vehicles", #lab of y-axis
        main = "heatmap", #title of the heatmap
        cexRow = 0.5) #adjust the size of row labels
```

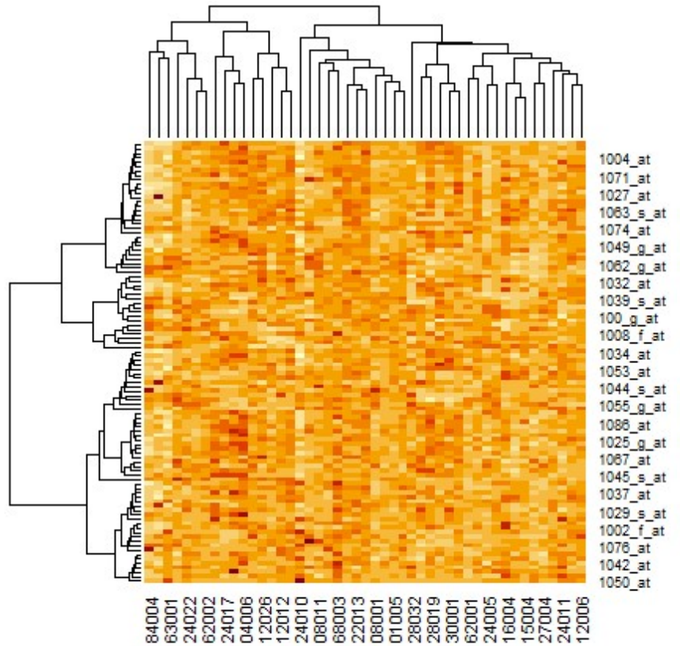


- **Heatmap from gene expression dataset**

```
#download data
if (!require("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install(version = "3.16")
BiocManager::install("ALL")
```

- **Obtain microarray data**

```
# Load gene expression dataset
# Select a dataset from two subgroups, BCR/ABL, and ALL1/AF4
# Use the first 100 rows to create a heatmap
library("ALL")
data("ALL")
eset <- ALL[, ALL$mol.biol %in% c("BCR/ABL", "ALL1/AF4")]
heatmap(exprs(eset[1:100,]))
```



- **Preprocessing data**

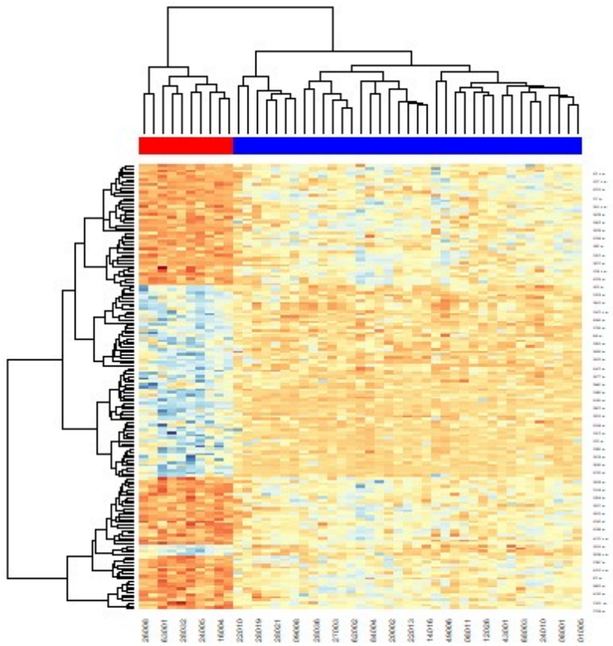
```
#Use the lmFit function to find genes differentially expressed
#Convert data frame to a matrix
library(limma)
f <- factor(as.character(eset$mol.biol))
design <- model.matrix(~f)
fit <- eBayes(lmFit(eset,design))
# Select genes with adjusted p-values below 0.05
selected <- p.adjust(fit$p.value[,2]) < 0.05
esetSel <- eset [selected, ]
```

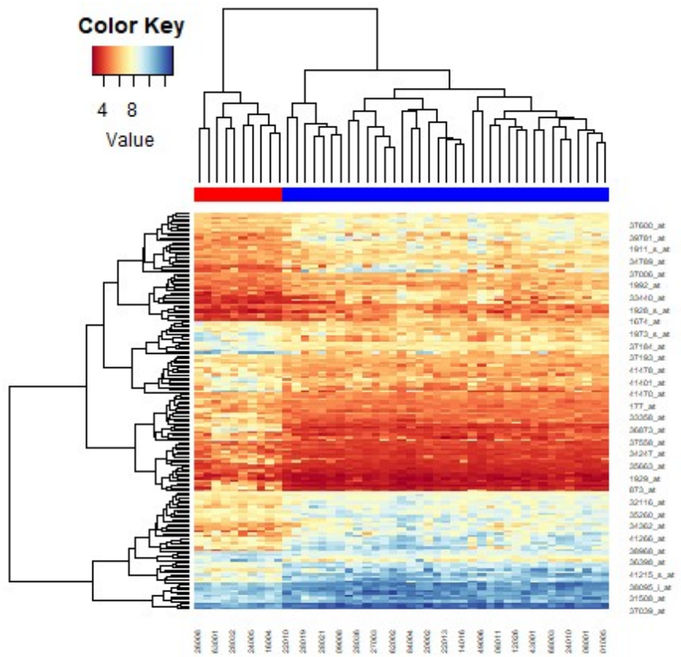
- **Plot a heatmap with sidebar**

```
color.map <- function(mol.biol) { if (mol.biol=="ALL1/AF4")
"red" else "blue"}
patientcolors <- unlist(lapply(esetSel$mol.bio, color.map))
heatmap(exprs(esetSel), col=col1, ColSideColors=patientcolors,
cexRow = 0.1, cexCol = 0.5)
```

Useful Plots for Bioinformatics

By Xiangming Huang





- **Use heatmap.2 to plot**

```
library("gplots")
heatmap.2(exprs(esetSel), #numeric matrix
          col=col1, #color scheme
          scale="none", #normalization
          ColSideColors=patientcolors, #add side bar
          key=TRUE, # color-key
          trace="none", #trace line along column or row
          symkey=FALSE, #symmetric color key
          density.info="none", #superimpose a plot on key
          cexRow = 0.5, #row label size
          cexCol = 0.5 #column label size
          )
```

VOLCANO PLOT

- **Library setup**

```
library(tidyverse)
library(ggrepel)
library(RColorBrewer)
```

- **Input format**

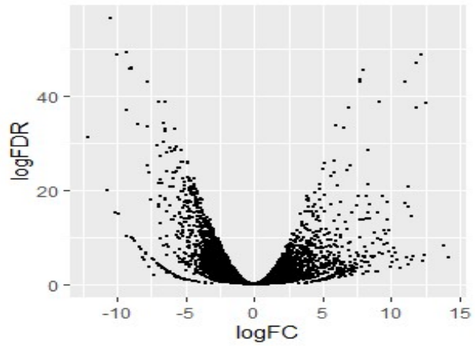
Genes	Log (fold change)	P-value	FDR (adjusted p-value)
String	Numeric value	Numeric value	Numeric value

- **Loading data (for demonstration)**

```
data3 <-
read_tsv("https://raw.githubusercontent.com/sdgamboa/misc_datasets/master/L0_vs_L20.tsv")
```

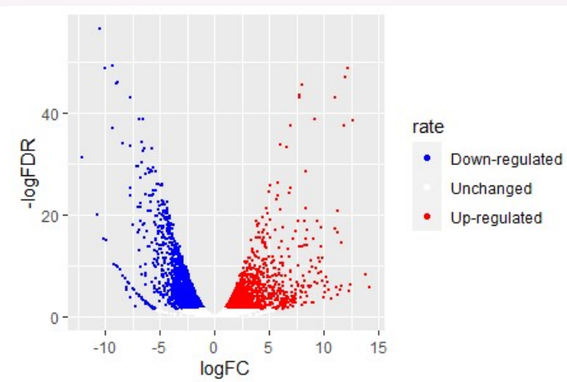
- **Basic volcano plot**

```
ggplot(data3,aes(logFC, -log(FDR,10))) +
  geom_point(size = 0.5) +
  xlab("logFC") +
  ylab("logFDR")
```



- **Coloring genes by fold change**

```
data3 <- data3 %>%
  mutate(
    rate = case_when (logFC >= 1 & FDR <= 0.05 ~ "Up-
regulated", logFC <= -1 & FDR <= 0.05 ~ "Down-regulated", TRUE
~ "Unchanged"))
ggplot(data3, aes(logFC, -log(FDR,10), color = rate)) +
  geom_point(size = 0.3) +
  xlab("logFC") +
  ylab("-logFDR") +
  scale_color_manual(values = c("blue", "white", "red")) +
  guides(colour = guide_legend(override.aes =
list(size=1.5)))
```

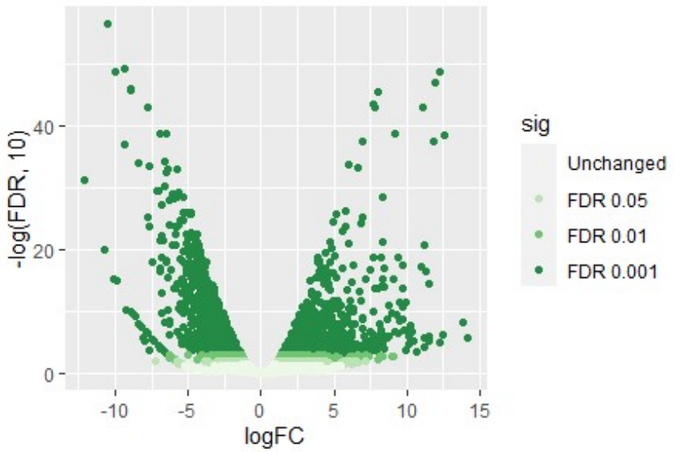


Useful Plots for Bioinformatics

By Xiangming Huang

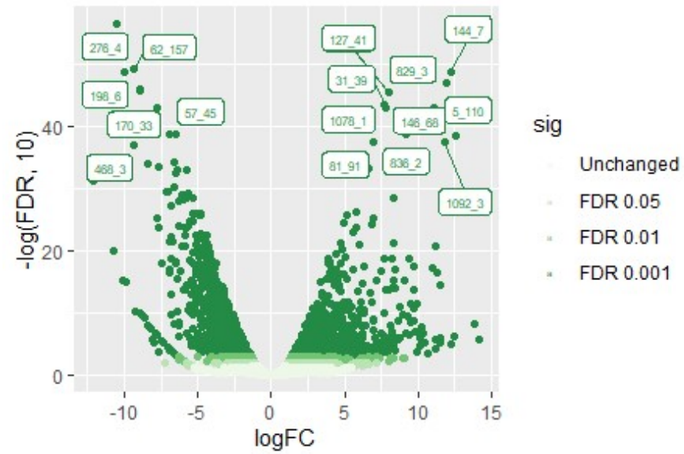
- **Coloring gene by significance**

```
data4 <- data3 %>%
  mutate(
    sig = case_when(
      abs(logFC) >= 1 & FDR <= 0.05 & FDR > 0.01 ~ "FDR 0.05",
      abs(logFC) >= 1 & FDR <= 0.01 & FDR > 0.001 ~ "FDR
0.01",
      abs(logFC) >= 1 & FDR <= 0.001 ~ "FDR 0.001",
      TRUE ~ "Unchanged")
  )
data4 <- within(data4, sig <- factor(sig, levels =
c("Unchanged", "FDR 0.05", "FDR 0.01", "FDR 0.001")))
p3 <- ggplot(data4, aes(logFC, -log(FDR,10), colour = sig)) +


  geom_point() +
  scale_colour_brewer(palette = "Greens") +
  guides(colour = guide_legend(override.aes = list(size=1.5)))
p3
```



- **Label significant genes**

```
top <- 10
top_genes <- bind_rows(
  data4 %>%
    filter(rate == 'Up-regulated') %>%
    arrange(FDR, desc(abs(logFC))) %>%
    head(top),
  data4 %>%
    filter(rate == 'Down-regulated') %>%
    arrange(FDR, desc(abs(logFC))) %>%
    head(top)
)

p3 <- p3 +
  geom_label_repel(data = top_genes,
                   mapping = aes(logFC, -log(FDR,10), label =
Genes),
                   size = 2)
p3
```



SOURCE

http://www.sthda.com/english/wiki/survival-analysis-basics

https://r-graph-gallery.com/215-the-heatmap-function.html

https://samdsblog.netlify.app/post/visualizing-volcano-plots-in-r/

https://warwick.ac.uk/fac/sci/moac/people/students/peter_cock/r/heatmap/