A Brief Guide for Constructing Survival Plots in R

The data to create survival analysis plots can be retrieved at www.data.oncomx.org

Data Configuration

OncoMX gathers data from TCGA on differential expression of genes in cancer in combination with survival clinical data. The differential expression data with clinical information needs to be modified slightly for the survival R libraries to be functional.

For the R libraries to read the survival data correctly, there needs to be at least three columns:

Differential expression group:

- For every patient, categorize as either high expression or low expression. We bifurcate these groups based on the median expression data in the group we are analyzing.
- Example: In a group of 41 patients for Colon Adenocarcinoma from TCGA, we examined differential expression of the gene CCL2. In this group the median log2fold change is -0.477.
- In a new column named "exp_group", all patients that are below the median log2fold are given the label "low" while all patients that are above the median log2fold change are given the label "high".

Outcome:

- For every patient, there is a category for "alive" or "dead".
- Make a new column that has a "0" if the patient has "alive" listed and a "1" if the patient has "dead' listed. We named this column "outcome_log".

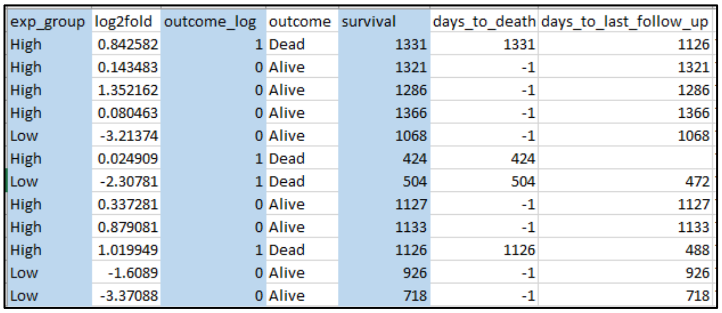
Survival time:

- For every patient, there needs to be a time listed for how many days they spent in the study. This number is listed in TCGA as "days_to_death" for patients listed as "dead" and "days_to_last_follow_up" for patients listed as "alive".
- Make a column that includes these numbers for each patient. We labeled this column "survival"

Here's an example of the properly formatted data file.

Columns in blue have been generated per the instructions above for use in R.

The original data columns that informed the newly formatted columns are white.



exp_group	log2fold	outcome_log	outcome	survival	days_to_death	days_to_last_follow_up
High	0.842582	1	Dead	1331	1331	1126
High	0.143483	0	Alive	1321	-1	1321
High	1.352162	0	Alive	1286	-1	1286
High	0.080463	0	Alive	1366	-1	1366
Low	-3.21374	0	Alive	1068	-1	1068
High	0.024909	1	Dead	424	424	
Low	-2.30781	1	Dead	504	504	472
High	0.337281	0	Alive	1127	-1	1127
High	0.879081	0	Alive	1133	-1	1133
High	1.019949	1	Dead	1126	1126	488
Low	-1.6089	0	Alive	926	-1	926
Low	-3.37088	0	Alive	718	-1	718

We performed these modifications in Excel to convert the columns:

- To create a column that translates the Alive/Dead column to 0/1 respectively
    - In Excel: =IF(G1=”Dead”,1,0)
        - In this case G1 is the column “outcome”
- To create a column that translates the Alive/Dead column to 0/1 respectively
    - In Excel: =IF(G1=”Dead”,1,0)
        - In this case G1 is the column “outcome”

- To create a column that has the overall time until either an incident or patient dropout:
    - In Excel: =IF(G1=”Dead”,M1,O1)
        - In this case G1 is the column “outcome”, M1 is the column days_to_death, and O1 is the column “days_to_last_follow_up”

Here is an R markdown file for running the survival analysis and generating the plot:

Survival_COAD_CCL2

Quick tutorial

Load libraries to make survival plots and comparisons

```
library(survival)
library(survminer)

## Loading required package: ggplot2

## Loading required package: ggpubr
```

Load csv file with survival and differential expression data

```
CCL2_COAD_JC <- read.csv(file = 'TCGA_COAD_JC/CCL2_COAD_JC.csv', header = TRU
E, stringsAsFactors = FALSE)
```

Make an object with survival data and indicate time of survival and alive/dead columns.

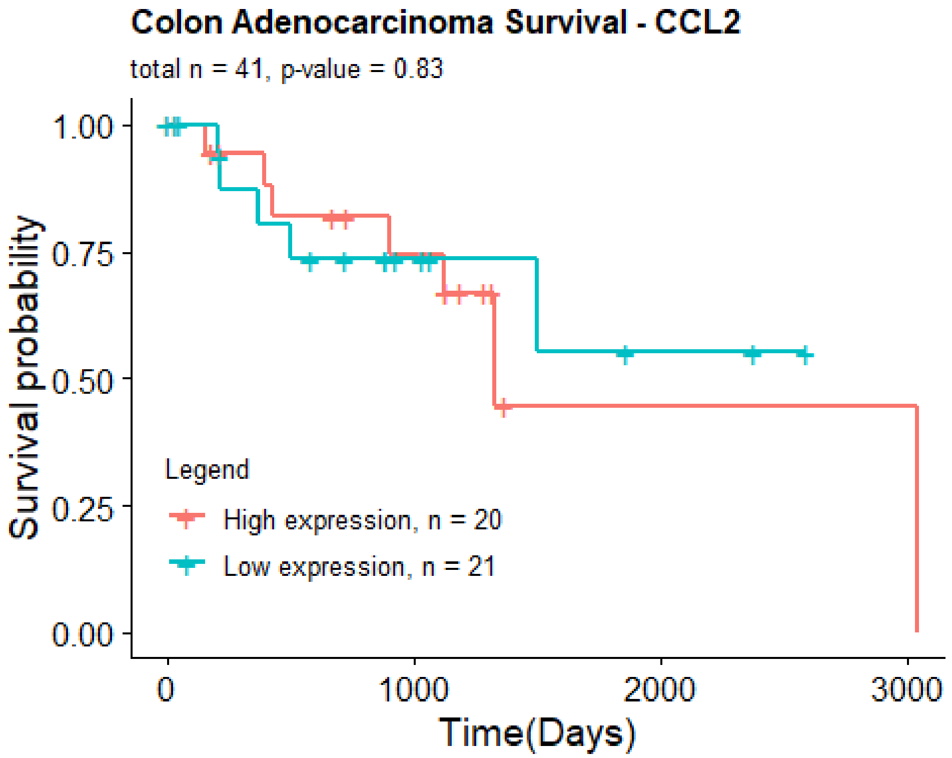
```
surv_object <- Surv(time = CCL2_COAD_JC$survival, event = CCL2_COAD_JC$outcom
e_log)
```

Make the Kaplan-Meier curves

```
fit1 <- survfit(surv_object ~ exp_group, data = CCL2_COAD_JC)
```

Make plots from the Kaplan-Meier curves

```
ggsurvplot(fit1, data = CCL2_COAD_JC,
           pval = FALSE,
           title = "Colon Adenocarcinoma Survival - CCL2",
           subtitle = "total n = 41, p-value = 0.83",
           font.title = c(12, "bold", "black"),
           font.subtitle = c(10, "plain", "black"),
           xlab = "Time(Days)",
           legend.labs = c("High expression, n = 20", "Low expression, n = 21
"),
           legend = c(0.25,0.25),
           legend.title = c("Legend")
           )
```



A note on the p-value in the graph above. If you would like to see the p-value for the Kaplan-Meier Analysis, set pval = TRUE for the ggsurvplot function. In this case, we chose to see the p-value using pval = TRUE, then we set pval = FALSE and wrote the value in the subtitle section for appearance's sake. We also took the total n and n for each group from excel and manually entered them, ggsurvplot does not automatically generate these numbers for you on the plot.