Urinary Biomarkers for Pancreatic Cancer

Log Theme09 - Introduction Machine Learning

Lisa Hu
414264
Bio-Informatics
Hanzehogeschool Groningen, ILST
Dave Langers (LADR) & Bart Barnard (BABA)

September 24, 2022

Contents

1 Data description

The data can be found on kaggle.com: Urinary biomarkers for pancreatic cancer

The following packages were used:

- ggplot2
- tidyr
- dplyr
- readr

1.1 Reading the data

Running the following command:

```
dataset <- read_excel("data/Data.xls")
```

Gave the following error:

```
filepath: <..>/Data.xls
libxls error: Unable to open file
```

Even though the filepath was correct, trying different filenames and changing the method, it still did not work. After scouring the internet, there was no clear solution to this. To avoid this error, open the xls-file in a program that can edit spreadsheets and export it to a functional file format (e.g. CSV).

From here, we can read the data and codebook:

```
dataset <- read.csv("data/Data.csv")
codebook <- read_delim("data/codebook.txt", delim = "|")
pander(codebook[1:4], booktabs = T, caption = "Data values", split.tables = 100)
```

Table 1: Data values

Name	Full Name	Type	Unit
sample_id	Sample ID	chr	-
patient_cohort	Patient's Cohort	chr	-
sample_origin	Sample Origin	chr	-
age	Age of subject	dbl	-
sex	Sex of subject	chr	-
diagnosis	Diagnosis	dbl	-
stage	Stage	chr	-
benign_sample_diagnosis	Benign Sample's Diagnosis	chr	-
plasma_CA19_9	Blood plasma CA19-9	dbl	U/ml
creatinine	Creatinine	dbl	mg/ml
LYVE1	LYVE1	dbl	ng/ml
REG1B	REG1B	dbl	ng/ml
TFF1	TFF1	dbl	ng/ml
REG1A	REG1A	dbl	ng/ml

```
pander(codebook[c(1,5)], booktabs = T, caption = "Description",
       justify = c("right", "left"), split.tables = 100)
```

Table 2: Description

Name	Description
sample_id	Unique string identifying each subject
patient_cohort	Cohort 1 = previously used samples; Cohort 2 = newly added samples
sample_origin	BPTB: Barts Pancreas Tissue Bank, London, UK; ESP: Spanish National Cancer Research Centre, Madrid, Spain; LIV: Liverpool University, UK; UCL: University College London, UK
age	Age in years
sex	M = male; F = female
diagnosis	1 = control (no cancer); 2 = benign hepatobiliary disease; 3 = PDA (pancreatic cancer)
stage	The stage of the disease (IA, IB, IIA, IIB, III, IV)
benign_sample_diagnosis	The diagnosis for those with a benign diagnosis
plasma_CA19_9	Blood plasma levels of CA19-9 monoclonal antibody, usually elevated when pancreatic cancer
creatinine	Urinary biomarker of kidney function
LYVE1	Urinary levels of Lymphatic Vessel Endothelial Hyaluronan receptor 1
REG1B	Urinary levels of Regenerating Family Member 1 Beta
TFF1	Urinary levels of Trefoil Factor 1
REG1A	Urinary levels of Regenerating Family Member 1 Alpha

The information given in the codebook originates from the `Documentation.xls`. This file was given with the data file and can be found on the website.

1.2 Manipulate the data

A lot of the rows contain empty strings instead of NA, so set them to NA first. Besides that, the columns `sample_id` , `patient_cohort`, `sample_origin`, and `benign_sample_diagnosis` in the dataset significant value for the analysis and are therefor dropped.

```
# Change the empty strings to NA
dataset[dataset == ""] <- NA

# Remove unnecessary columns
drop <- c("sample_id", "patient_cohort", "sample_origin", "benign_sample_diagnosis")
dataset <- dataset[,!(names(dataset) %in% drop)]

pander(summary(dataset), split.table = 100)
```

Table 3: Table continues below

age	sex	diagnosis	stage	plasma_CA19_9
Min. :26.00	Length:590	Min. :1.000	Length:590	Min. : 0.0
1st Qu.:50.00	Class :character	1st Qu.:1.000	Class :character	1st Qu.: 8.0
Median :60.00	Mode :character	Median :2.000	Mode :character	Median : 26.5
Mean :59.08	NA	Mean :2.027	NA	Mean : 654.0
3rd Qu.:69.00	NA	3rd Qu.:3.000	NA	3rd Qu.: 294.0
Max. :89.00	NA	Max. :3.000	NA	Max. :31000.0
NA	NA	NA	NA	NA's :240

creatinine	LYVE1	REG1B	TFF1	REG1A
Min. :0.05655	Min. : 0.000129	Min. : 0.0011	Min. : 0.005	Min. : 0.00
1st Qu.:0.37323	1st Qu.: 0.167179	1st Qu.: 10.7572	1st Qu.: 43.961	1st Qu.: 80.69
Median :0.72384	Median : 1.649862	Median : 34.3034	Median : 259.874	Median : 208.54
Mean :0.85538	Mean : 3.063530	Mean : 111.7741	Mean : 597.869	Mean : 735.28
3rd Qu.:1.13948	3rd Qu.: 5.205037	3rd Qu.: 122.7410	3rd Qu.: 742.736	3rd Qu.: 649.00
Max. :4.11684	Max. :23.890323	Max. :1403.8976	Max. :13344.300	Max. :13200.00
NA	NA	NA	NA	NA's :284

A summary of the data shows very high maximum values, but rather low medians. A log-transformation is applied to correct this. The missing values in the REG1A column will not be imputed since there is a lot of them and the imputation would only

```
log.data <- log(dataset[5:10] +1)
colnames(log.data) <- paste("log", names(dataset[5:10]), sep = ".")
dataset <- cbind(dataset, log.data)
```

The samples are then grouped by diagnosis for quicker access of the different samples. Table 3 shows the different amounts of samples per diagnosis and the amount of which are also blood samples. After the blood samples are seperated, they can be deleted from the original dataset and the column can be dropped.

```
# Different diagnosis and blood groups
control <- subset(dataset, diagnosis == 1)
benign <- subset(dataset, diagnosis == 2)
pdac <- subset(dataset, diagnosis == 3)
blood <- subset(dataset, plasma_CA19_9 >= 0)
```

```r
# Delete all the entries with blood sample and drop "plasma" columns
dataset <- dataset[-which(dataset$plasma_CA19_9 >= 0),]
dataset <- dataset[,-c(5, 11)]

# Demographics
demograph <- data.frame(c(sum(control$sex == "F"), sum(control$sex == "M")),
                        c(sum(benign$sex == "F"), sum(benign$sex == "M")),
                        c(sum(pdac$sex == "F"), sum(pdac$sex == "M")))

blood.demo <- data.frame(c(sum(blood$sex == "F" & blood$diagnosis == 1),
                           sum(blood$sex == "M" & blood$diagnosis == 1)),
                         c(sum(blood$sex == "F" & blood$diagnosis == 2),
                           sum(blood$sex == "M" & blood$diagnosis == 2)),
                         c(sum(blood$sex == "F" & blood$diagnosis == 3),
                           sum(blood$sex == "M" & blood$diagnosis == 3)))

colnames(blood.demo) <- c("Control", "Benign", "PDAC")
colnames(demograph) <- c("Control", "Benign", "PDAC")
demograph <- rbind(demograph, blood.demo)
rownames(demograph) <- c("Female total", "Male total", "Female blood", "Male blood")

pander(demograph, booktabs = T, caption = "Demographic of the samples",
       justify = c("left", "center", "center", "center"))
```

Table 5: Demographic of the samples

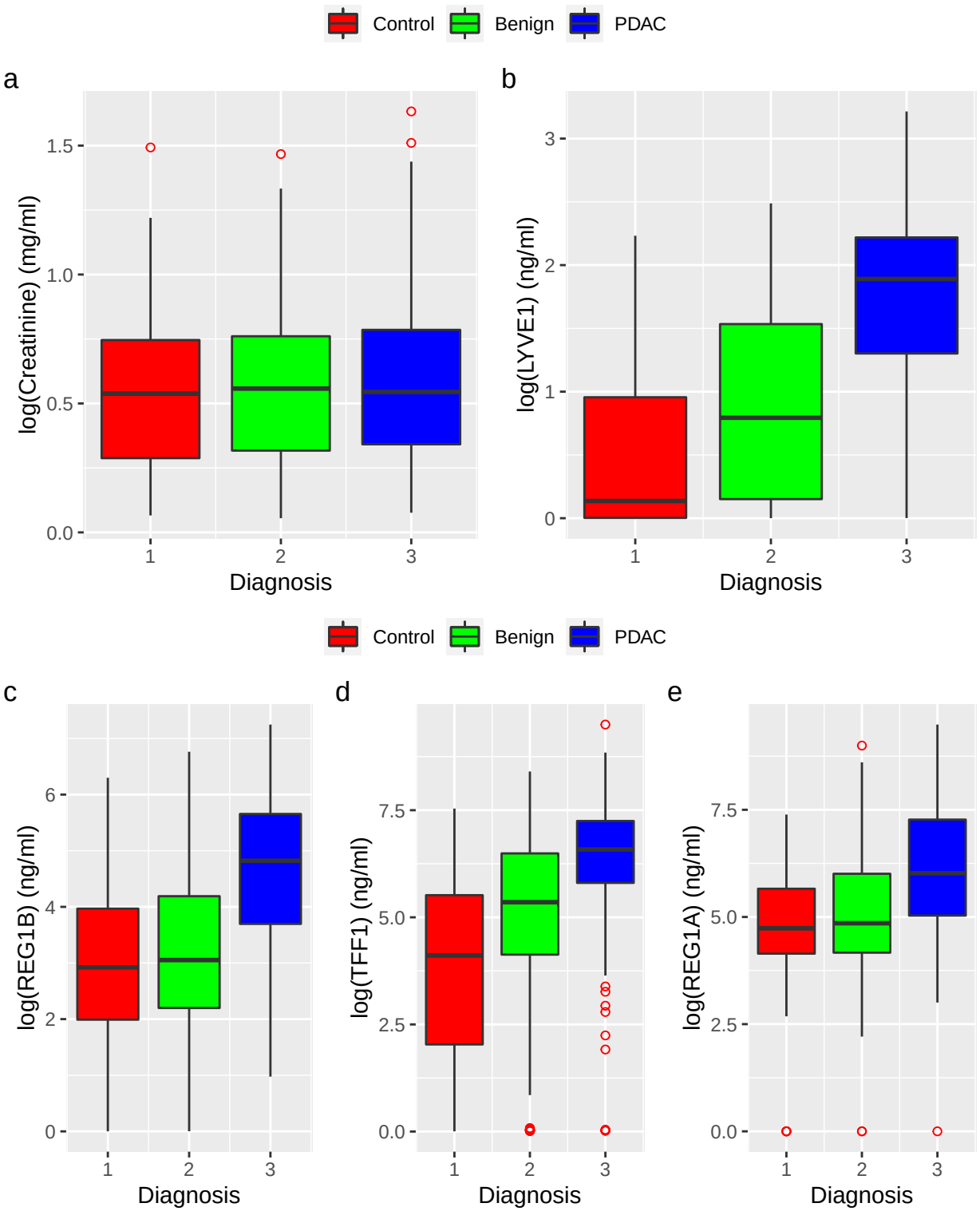
	Control	Benign	PDAC
Female total	115	101	83
Male total	68	107	116
Female blood	58	57	64
Male blood	34	51	86

1.3 Analyse the data

```
# Boxplot function
create.plots <- function(y.values, y.label, plt.tag) {
  list(ggplot(data = control, aes(x = diagnosis, y = !!sym(y.values))) +
    geom_boxplot(outlier.color = "red", outlier.shape = 1, aes(fill = "Control")) +
    geom_boxplot(data = benign, outlier.color = "red", outlier.shape = 1,
                 aes(fill = "Benign")) +
    geom_boxplot(data = pdac, outlier.color = "red", outlier.shape = 1,
                 aes(fill = "PDAC")) +
    labs(x = "Diagnosis", y = y.label, tag = plt.tag) +
    scale_fill_manual(values = c("red", "green", "blue"),
                      limits = c("Control", "Benign", "PDAC"),
                      name = ""))
}

# Create the boxplots for the different columns
y.values <- names(dataset[10:14])
y.labs <- c("log(Creatinine) (mg/ml)", "log(LYVE1) (ng/ml)", "log(REG1B) (ng/ml)",
            "log(TFF1) (ng/ml)", "log(REG1A) (ng/ml)")
plt.tag <- c("a", "b", "c", "d", "e")
plts <- mapply(create.plots, y.values, y.labs, plt.tag)

# Grid and print the plots
p1 <- ggarrange(plotlist = plts[1:2], ncol = 2,
                common.legend = TRUE, legend = "top")
p2 <- ggarrange(plotlist = plts[3:5], ncol = 3,
                common.legend = TRUE, legend = "top")
my.grid <- ggarrange(p1, p2, nrow = 2)
print(annotate_figure(my.grid))
```



The outliers are not localized in a specific diagnosis group, but rather spread around everywhere.

1.3.1 Correlation matrix