logbook

Rob Meulenkamp

9/19/2021

Topic Research

For this research the breast cancer remains the primary focus. Somatic mutations play an import role in the development of breast cancer. The effect of the mutations remains poorly understood. The data provide 105 annotated breast cancer, which 77 provides high-quality data. Machine learning provides calculated patterns, which could help to interpret the effects of somatic mutations with breast cancer. [1]

Research question

Could you predict the stage of breast cancer based on the protein expression values?

Exploratory Data Analysis

The proteomic data was created using reverse phase protein arrays (RPPA). Performed by the The Cancer Genome Atlas (TCGA) breast cancer study; but this is limited with the availability of antibody. The NCI Clinical Proteomic Tumor Analysis Consortium (CPTAC) was using mass spectrometry to provide in depth anlyses about the annotated proteomes TCGA tumor samples.[2] [3]

77_cancer_proteomes_CPTAC_itraq.csv attributes:
1. `RefSeq_accession_number` : Provides accession number protein.
2. `gene_symbol` : Contains gene symbol if present.
3. `gene_name` : Excess to the gene name.
4. Other columns: Identifier from every patient with the expression value (log2 iTRAQ ratios).

clinical_data_breast_cancer.csv attributes:
1. `Complete.TCGA.ID` : Identifier from the patients.
2. `Tumor` : Stages tumor based on the size of the tumor.
Other columns aren't used for this project.

Loading the data sets in R and get a nice overview.

```
p_data <- read.csv("data/77_cancer_proteomes_CPTAC_itraq.csv")

clinic <- read.csv("data/clinical_data_breast_cancer.csv")


table_view <- hux(Dataset = c("cancer_proteomes", "clinical_data"),
                  Records = c(86, 30),
                  Measurements = c(12553 ,105))
table_view %>%
      set_all_padding(4) %>%
      set_outer_padding(0) %>%
```

```
      set_number_format(0) %>%
      set_bold(row = 1, col = everywhere) %>%
      set_bottom_border(row = 1, col = everywhere) %>%
      set_width(0.8) %>%
      set_position("left")
```

Dataset	Records	Measurements
cancer_proteomes	86	12553
clinical_data	30	105

```
# Select few columns for quick view
table_proteome <- p_data %>%
            select(1:4)

table_clinic <- clinic %>%
                  select(1:7)

pander(head(table1, 5))
```

country	year	cases	population
Afghanistan	1999	745	19987071
Afghanistan	2000	2666	20595360
Brazil	1999	37737	1.72e+08
Brazil	2000	80488	174504898
China	1999	212258	1.273e+09

```
pander(head(table_clinic, 5))
```

Table 2: Table continues below

Complete.TCGA.ID	Gender	Age.at.Initial.Pathologic.Diagnosis	ER.Status
TCGA-A2-A0T2	FEMALE	66	Negative
TCGA-A2-A0CM	FEMALE	40	Negative
TCGA-BH-A18V	FEMALE	48	Negative
TCGA-BH-A18Q	FEMALE	56	Negative
TCGA-BH-A0E0	FEMALE	38	Negative

PR.Status	HER2.Final.Status	Tumor
Negative	Negative	T3
Negative	Negative	T2
Negative	Negative	T2
Negative	Negative	T2
Negative	Negative	T3

Unfortunately the ID of the two data sets doesn't match. This is about the "Complete.TCGA.ID" column. Also, the data set cancer proteomes contain three duplicates and three healthy patients at the end of file. This need to filter out before the start of the research. The code below changes the ID and allow to join the

two data sets on this variable.

```
n <- p_data$RefSeq_accession_number

#Get all but first 3 columns
proteomes <- as.data.frame(t(p_data[,4:83]))
colnames(proteomes) <- n

proteomes <- cbind(rownames(proteomes), data.frame(proteomes, row.names=NULL))
colnames(proteomes)[1] <- "Complete.TCGA.ID"

#Function string manipulation
get.clinical.id <- function(proteome.id) {
  x = substr(proteome.id, 4, 7)
  y = substr(proteome.id, 0, 2)
  paste("TCGA",y,x,sep="-")
}

proteomes$Complete.TCGA.ID <- sapply(proteomes$Complete.TCGA.ID, get.clinical.id)
proteomes_all <- proteomes


#Remove the duplicates
proteomes_all <- proteomes_all[!duplicated(proteomes_all$Complete.TCGA.ID),]
```

```
#Merge the Tumor column from clinic data
test <- merge(clinic, proteomes_all, BY = "Complete.TCGA.ID")
test <- test[-c(2:6, 8:30)]
```

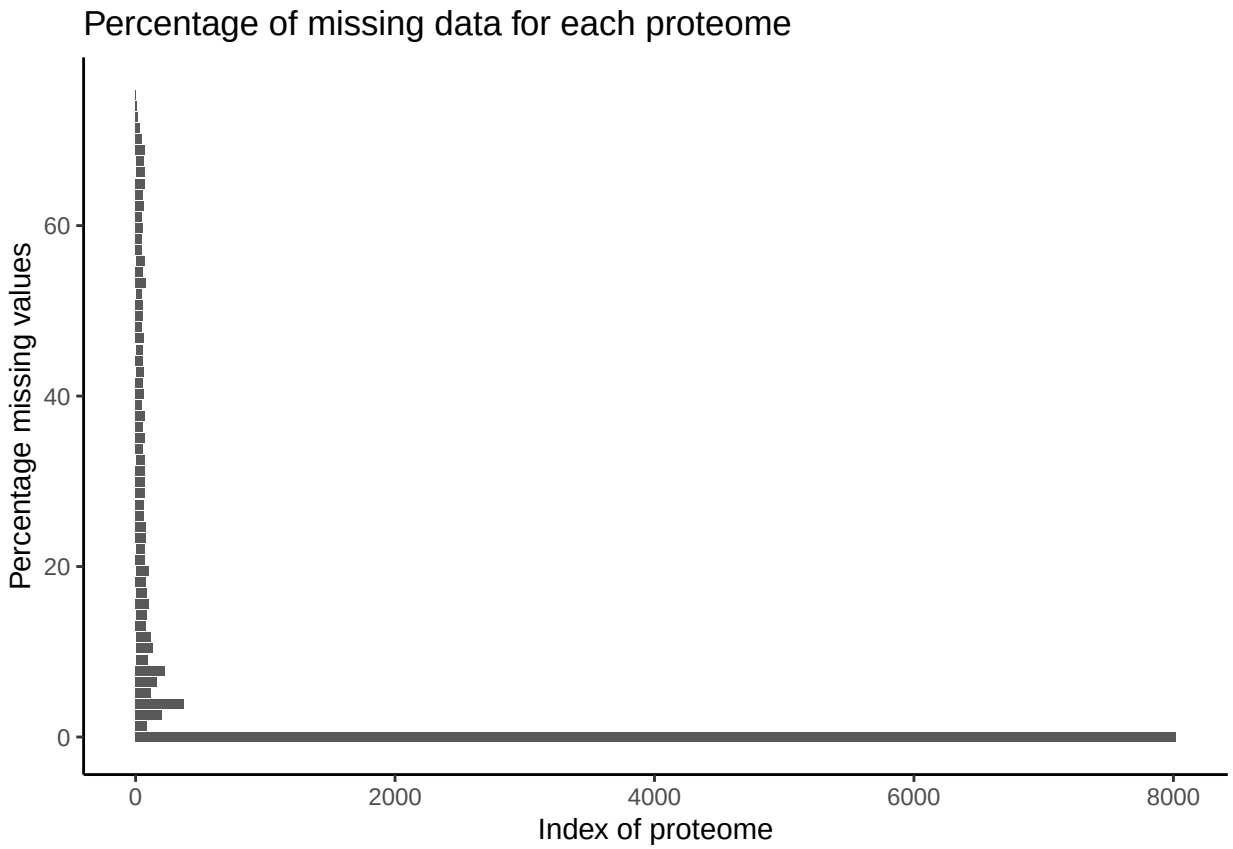
Now, the two data sets are merged together, there is room to investigate the missing values in file. It is necessary to inspect for each proteome how many NA values it contains. Proteomes with too many NA-values get filtered out.

```
na_count <- colSums(is.na(proteomes_all)) / nrow(proteomes_all) * 100

na_count <- as.data.frame(na_count)



ggplot(na_count, aes(y=na_count)) +
  geom_bar() +
  ylab("Percentage missing values") +
  xlab("Index of proteome") +
  theme_classic() +
  ggtitle("Percentage of missing data for each proteome")
```



A small proportion of the proteomes contains NA-values. For this project proteomes who hold more than 20% missing values get filtered out.
Proteomes with less than 20% missing values and still have NA-values will be replaced with the computed median value.

```
# remove variable with >20% missing data
proteomes_all <- test[ , colSums(is.na(test))  / nrow(proteomes_all) < 0.20]




for (i in which(sapply(proteomes_all, is.numeric))) {
    proteomes_all[is.na(proteomes_all[, i]), i] <- median(proteomes_all[, i], na.rm = TRUE)
}
```

```
## Amount of proteomes with more than 20% missing data : 2499
```

From the 12554 proteomes 2499 proteomes with more than 20% missing values are filtered out the data set. Leaving a data set ready for usage.

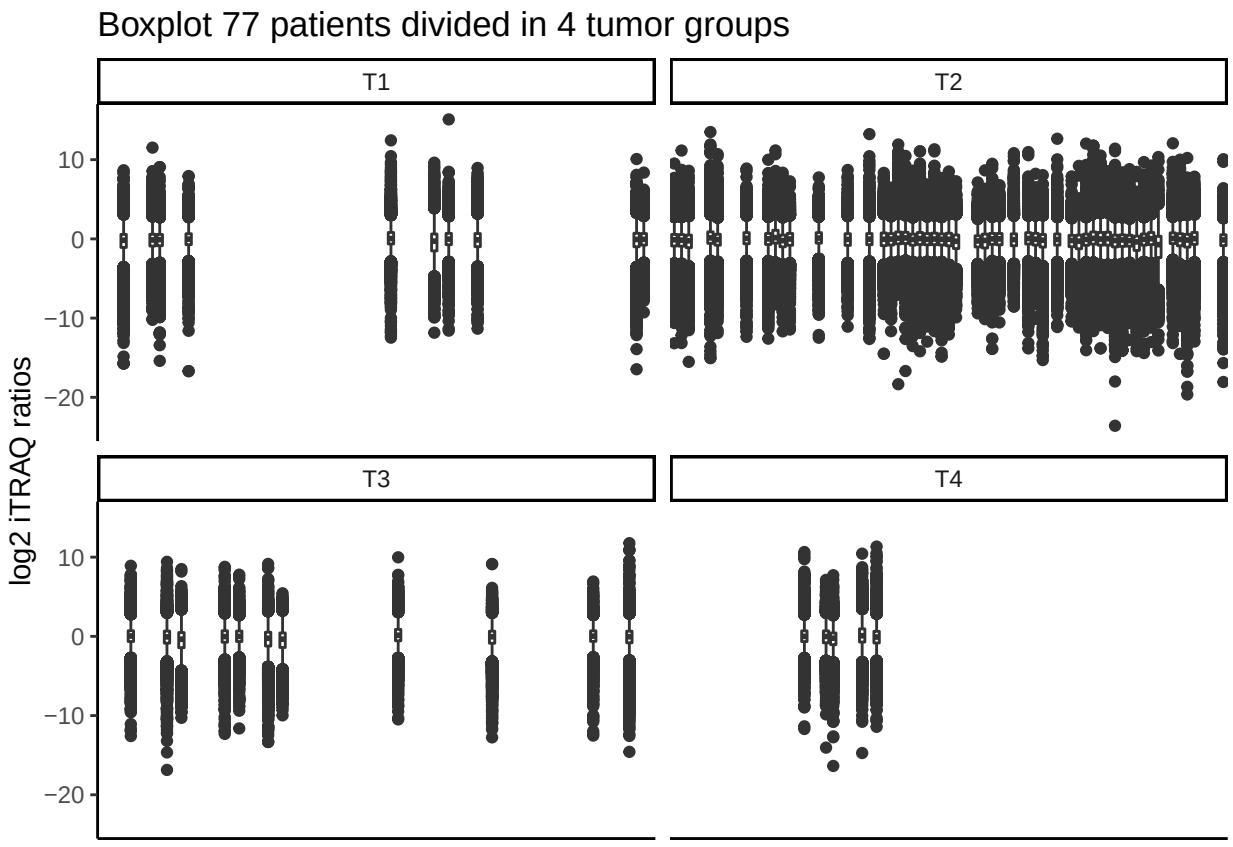
Boxplots provides clear information about outliers in the data set. It is important to inspect all the patients expression values to detect outliers. Outliers create noise and could effect the result in the wrong way.

```
# change the format from wide to long
long <- melt(proteomes_all)
```

```
## Using Complete.TCGA.ID, Tumor as id variables
```

```
ggplot(long, aes(x=Complete.TCGA.ID, y = value)) +
  geom_boxplot() +
```

```
  facet_wrap(~Tumor) +
  theme_classic() +
  theme(axis.title.x = element_blank(),
        axis.text.x=element_blank(),
        axis.ticks.x=element_blank()) +
  ylab("log2 iTRAQ ratios") +
  ggtitle("Boxplot 77 patients divided in 4 tumor groups")
```



The first thing what the eye catch, is the number of patients in tumor group 2 in comparison with tumor group 4. The difference between the number of patients is tremendous. This could have a big impact with the accuracy of predicting the tumor group for every patient. The classes are unevenly represented. There aren't any outliers in the data set.

A pie chart helps to get a better view with the class distribution. The proportional representation of patients in every tumor group is displayed in the frame.

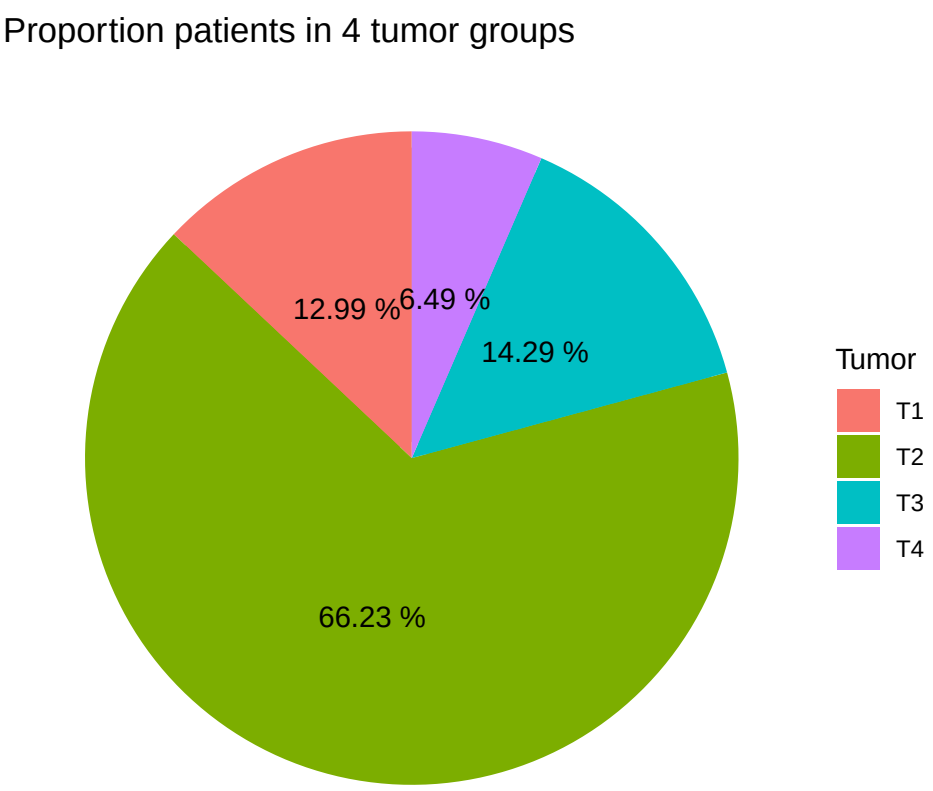
```
blank_theme <- theme_minimal()+
  theme(
  axis.title.x = element_blank(),
  axis.title.y = element_blank(),
  panel.border = element_blank(),
  panel.grid=element_blank(),
  axis.ticks = element_blank(),
  plot.title=element_text(size=14, face="bold")
  )


count_t <- proteomes_all %>%
```

```
          filter(!is.na(Tumor)) %>%
          group_by(Tumor) %>%
          count()

pct <- paste(round(count_t$n / sum(count_t$n) * 100, digits = 2), "%")

ggplot(count_t, aes(x="", y= n, fill= Tumor )) +
        geom_bar(stat="identity", width=1) +
        coord_polar("y", start=0) +
        geom_col() +
        blank_theme +
        theme_void() +
        geom_text(aes(label = pct),
              position = position_stack(vjust = 0.5)) +
        ggtitle("Proportion patients in 4 tumor groups")
```



First, the group T2 exists of 66.23 % of the patients. The group T1, T3 and T4 have combined less patients in comparison with group T2. This could have effect on the accuracy of predicting the tumor group for the minority representative tumor groups.

References

[1] Mertins, Philipp, et al. *Proteogenomics connects somatic mutations to signalling in breast cancer.* Nature 534.7605 (2016): 55-62.

[2] Ellis, M. J. et al. *Connecting genomic alterations to cancer biology with proteomics: the NCI Clinical Proteomic Tumor Analysis Consortium* Cancer Discov. 3, 1108–1112 (2013).

[3] Zhang, B. et al. *Proteogenomic characterization of human colon and rectal cancer* Nature 513, 382–387 (2014).