Contents

1 Loading of libraries **1**

2 Loading/Splitting training and testing sets **1**

2.1 Data visualization 1

3 Signature calculation **4**

4 Train the model **4**

4.1 Preparation of the data needed for the training of the model 4

4.2 Training results 5

4.3 Plots of results from training 5

4.4 Plot results from testing 5

---

title: “Notebook for the SBC on the GBM dataset from the Firebrowse Database(TCGA)” author: “Camila Duitama (Based on Ashar Ahmad’s code for the SBC model)” date: “May 2019” output: html_document: df_print: paged toc: yes toc_depth: ‘2’ pdf_document: highlight: zenburn df_print: kable fig_caption: yes fig_height: 6 fig_width: 10 number_sections: yes toc: yes toc_depth: ‘2’

---

1 Loading of libraries

Libraries needed to run the coded must be loaded first

2 Loading/Splitting training and testing sets

Data is separated into training and testing

2.1 Data visualization

Initial visualization of censoring, survival times and mRNAArray Data of the training and testing set

```
table(censoring) %>%
  kable(caption = "Censoring frequency for the training set") %>%
  kable_styling(bootstrap_options = c("striped", "hover","responsive"),full_width = T, position = "cente
```

Censoring frequency for the training set

censoring

Freq

1

77

```
table(censoring.new) %>%
  kable(caption = "Censoring frequency for the testing set") %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed","responsive"),full_width = T, posi
```

Censoring frequency for the testing set

censoring.new

Freq

1

77

```
  Y.pre.train[1:5,1:5] %>%
  kable(caption = "mRNA MicroArray data for the Training set " ) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed","responsive"),full_width = T, posi
```

mRNA MicroArray data for the Training set

AACS

FSTL1

ELMO2

CREB3L1

RPS11

6.539245

9.794400

6.213981

4.836276

10.81124

7.186891

4.945053

5.230444

5.818606

10.47730

7.675038

10.840095

6.620676

5.333213

10.63727

7.996010

8.931571

7.552416

6.087341

11.00153

8.355122

4.240622

6.707334

4.865492

10.68588

```
  Y.pre.test[1:5,1:5] %>%
  kable(caption = "mRNA MicroArray data for the Testing set " ) %>%
  kable_styling(bootstrap_options = c("striped", "hover", "condensed","responsive"),full_width = F, pos
```

mRNA MicroArray data for the Testing set

AACS

FSTL1

ELMO2

CREB3L1

RPS11

5.936346

10.623493

6.996271

4.883836

10.51334

5.665690

10.109494

6.795128

4.862345

10.37863

5.714858

9.320218

6.893214

4.227045

10.29740

6.586214

10.458615

7.077229

5.274177

11.35884

8.967289

7.380626

7.504833

4.365866

11.23741

3 Signature calculation

This chunk calculates SBC gene signature It's based on the idea of Univariate testing of Survival Data features NOTE: For now the rows where there is no censoring data will be removed from ALL datasets, until we find an imputation method.

The SBC signature on the dataset looks like this

```
 # to_plot<-data.frame(Genes=signature.sbc)
 #  to_plot%>%
 #    kable(caption = "SBC Signature") %>%
 #    kable_styling(bootstrap_options = c("striped", "hover", "condensed","responsive"),full_width = F,
 #    scroll_box(width = "500px", height = "200px")
```

```
Label<- c()
for (i in 1:length(Clinical_TrainingSet$days_to_death)){
  if (i<=n) {
  Label<-append(Label,"Training")
  }else{
  Label<-append(Label,"Testing")
  }}
timedf<-data.frame(logtime=log(Clinical_TrainingSet$days_to_death),set=Label)
mu <- ddply(timedf, "Label", summarise, grp.mean=mean(logtime))
ggplot(data =timedf,aes(x=logtime,color=Label,fill=Label)) +
  geom_histogram(bins = 15, alpha=0.5, position="dodge")+
  labs(y="Counts", x = "log(time)")+ ggtitle("Log(time) for training and testing data")+
  geom_vline(data=mu, aes(xintercept=grp.mean, color=Label),linetype="dashed")
```

plot of chunk Histogram of log(time) for both training and testing set

4 Train the model

4.1 Preparation of the data needed for the training of the model

```
##############################################################################
###### We prepare the Data Structures Needed for the Running of the SBC ####
##############################################################################

######## Verhaak Signature ###############
Verhaak_gene_signature <-read_csv("/Volumes/GoogleDrive/My Drive/Documents/Life Science Informatics Mas
                                  col_names = FALSE, col_types = cols(X1 = col_skip()))
Verhaak_gene_signature<-Verhaak_gene_signature[-indexes,]
signature.vk<-Verhaak_gene_signature$X2
```

```
######## Verhaak Labels #################
labels.vk<-Clinical_TrainingSet$Subtype

#######Getting signature matrix on training and testing######
Y <- data.matrix(Y.pre.train[,as.character(signature.sbc)])
Y.new <- data.matrix(Y.pre.test[,as.character(signature.sbc)])
smod.new <- Surv(time.new, censoring.new)
c.true <- as.factor(labels.vk[1:n])
c.true <- unclass(c.true)
c.true.new<-as.factor(labels.vk[-(1:n)])
c.true.new<-unclass(c.true.new)
```

4.2 Training results

4.3 Plots of results from training

```
################################
### Some plots and analysis ####
################################
#### Generating some plots and results ON Training data ###
logrank <- survdiff(smod ~ c.final)
pval<-1 - pchisq(unlist(logrank)$chisq,df =3)
surv.fit <- survfit(smod ~ c.final)
p5 <- ggsurv(surv.fit, plot.cens=FALSE,main = " DPMM \n Kaplan Meier Estimators \n Verhaak Cancer Data S
p5
```

plot of chunk Survival curves from training

```
############ Generating some Plots ##########################
pc <- prcomp(Y)
pc.pred <- predict(pc,newdata = Y)
p1 <- ggplot(as.data.frame(pc.pred), aes(x=pc.pred[,1], y= pc.pred[,2], colour= as.factor(c.final))) + g
p1
```

plot of chunk PCA from training #Testing

4.4 Plot results from testing

```
logrank.new <- survdiff(smod.new ~ c.sbc.new)
#df= Degrees of freedom should be number of clusters-1
pval.new<-1 - pchisq(unlist(logrank.new)$chisq,df =length(logrank.new$n)-1)
surv.fit <- survfit(smod.new ~ c.sbc.new)
p5.new <- ggsurv(surv.fit, plot.cens=FALSE,main = " DPMM \n Kaplan Meier Estimators \n Verhaak Cancer Da
p5.new
```

plot of chunk Survival curves from testing

```
pc <- prcomp(Y.new)
pc.pred <- predict(pc,newdata = Y.new)
p3.new <- ggplot(as.data.frame(pc.pred), aes(x=pc.pred[,1], y= pc.pred[,2], colour= as.factor(c.sbc.new)
p3.new
```

plot of chunk PCA from testing

```
#############
```