CD138+ spectra and established expression-based risk scores

Rosalie Griffin Waller

27-OCT-2020

```
# Install and load required R packages
library(dplyr)
library(data.table)
library(ggplot2)
```

```
## Warning: package 'ggplot2' was built under R version 3.6.3
```

Define data directory

```
data_dir = "/path/to/data" # exclude ending "/"
```

Load data

```
# read in normalized and combat adjusted expression data on baseline samples
exp_cbat = read.csv(file = paste0(data_dir,
                                  "/baseline-expression-norm-combat.csv"))

# read in spectra and clinical data, remove clinical data
spectra = read.csv(file = paste0(data_dir,
                                  "/baseline-clinical-spectra-sd.csv")
                   ) %>% dplyr::select("SEQ_ID",starts_with("PC"))
```

1. University of Arkansas UAMS-70 gene panel to classify patients as low or high risk for relapse

DOI: 10.1182/blood-2006-07-038430

1.1. COMPUTE SCORE IN COMMPASS DATA

Genes not found in data

Down-regulated: “PNPLA4”,“KIAA1754”,“MCLC”,“AD-020”,“PARG1”,“RFP2”,“FLJ20489”;

Up-regulated: “FABP5”,“PDHA1”,“TRIP13”,“SELI”,“SLCI19A1”,“ASPM”,“STK6”, “FLJ13052”,“LAS1L”,“BIRC5”,“CKA
“MGC4308”,“DSG2”,“C6orf173”,“MGC15606”,“KIF14”,“DKFZP586L0724”,“WEE1”,“ROBO1”, “MPHOSPH1”

```
# working from ComBat adjusted expression estimates in baseline samples
DAT = exp_cbat

# LIST OF UP-REGULATED GENES IN UAMS-70 GENE SCORE
up = c("FABP5","PDHA1","TRIP13","AIM2","SELI","SLCI19A1","LARS2",
```

```
       "OPN3","ASPM","CCT2","UBE2I","STK6","FLJ13052",
        "LAS1L","BIRC5","RFC4","CKS1B","CKAP1","MGC57827",
       "DKFZp779O175","PFN1","ILF3","IFI16","TBRG4","PAPD1",
        "EIF2C2","MGC4308","ENO1","DSG2","C6orf173","EXOSC4",
       "TAGLN2","RUVBL1","ALDOA","CPSF3","MGC15606","LGALS1",
        "RAD18","SNX5","PSMD4","RAN","KIF14","CBX3","TMPO",
       "DKFZP586L0724","WEE1","ROBO1","TCOF1","YWHAZ",
        "MPHOSPH1")
print(paste0("UAMS up regulated genes: ",
             length(intersect(up,colnames(DAT))),
             " of ",
             length(up),
             " genes in dataset"))
```

```
## [1] "UAMS up regulated genes: 22 of 50 genes in dataset"
```

```
# select up regulated genes in data
anno_up = DAT %>% dplyr::select(intersect(up,colnames(DAT)))

# LIST OF DOWN-REGULATED GENES IN UAMS-70 GENE SCORE
down = c("GNG10","PNPLA4","KIAA1754","AHCYL1","MCLC","EVI5","AD-020",
         "PARG1","CTBS","UBE2R2","FUCA1","RFP2","FLJ20489","LTBP1","TRIM33")
print(paste0("UAMS down regulated genes: ",
             length(intersect(down,colnames(DAT))),
             " of ",length(down),
             " genes in dataset"))
```

```
## [1] "UAMS down regulated genes: 7 of 15 genes in dataset"
```

```
# select up regulated genes in data
anno_dw = DAT %>% dplyr::select(intersect(down,colnames(DAT)))

# COMPUTE GEOMETRIC MEANS
x = DAT[,"SEQ_ID"] %>% data.table
colnames(x) = "SEQ_ID"
x$up = anno_up %>% rowMeans()
x$dw = anno_dw %>% rowMeans()

# COMPUTE PROPORTION OF MEAN UP/DOWN AND PLOT
x$score = x$up - x$dw # Note: expression already in log2 scale
#hist(x$score,breaks = 200,main = "Risk Score",
#xlab = "log2(mean up reg) - log2(mean down reg)")

#mean(x$score)
rm(DAT,anno_up,up,down,anno_dw) # Cleanup variables
```

Clustering

```
# K-Means Cluster Analysis
fit <- kmeans(x$score, 3)

# get cluster max
aggregate(x$score,by=list(fit$cluster),FUN=max)
```

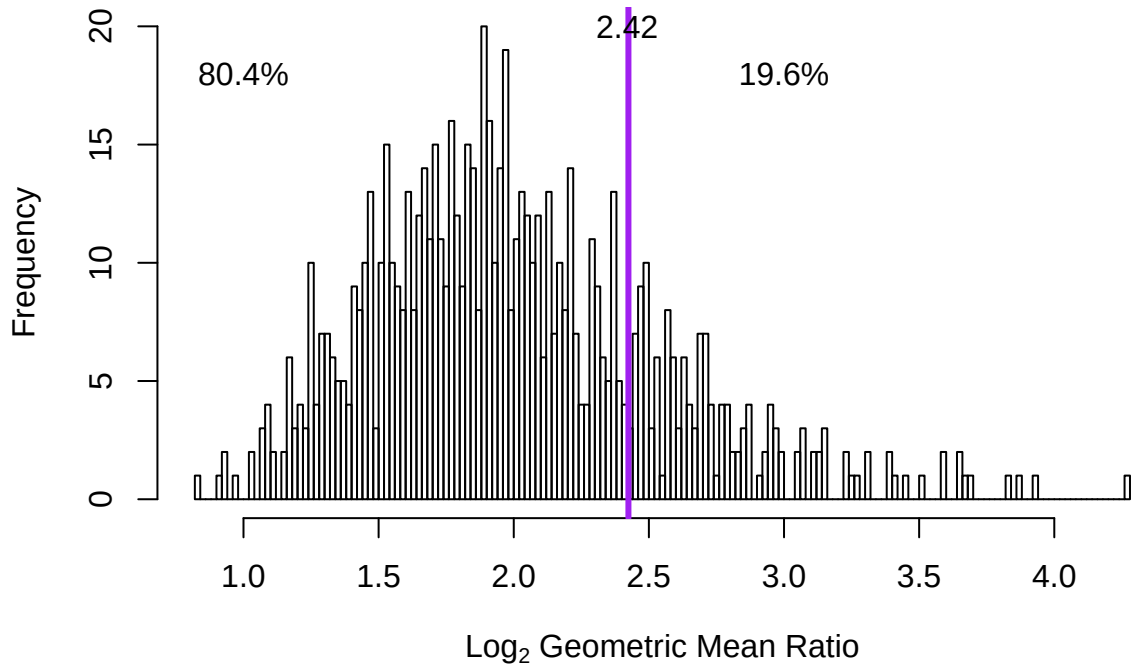
```
##   Group.1        x
## 1       1 4.274634
## 2       2 1.759113
## 3       3 2.424285
```

```
# append cluster assignment
x.score <- data.frame(x, fit$cluster)

# proportion in high/low risk group
#nrow(x.score[x.score$score > 2.424285,])/nrow(x.score)
#nrow(x.score[x.score$score < 2.424285,])/nrow(x.score)
```

Histogram

```
h <- hist(x.score$score,breaks = 150,main = "",
          xlab = expression("Log"[2]*" Geometric Mean Ratio"))#,ylim = c(0,18))
abline(v=2.424285, col="purple",lwd=3)
text(x=1,y=18,labels = "80.4%")
text(x=3,y=18,labels = "19.6%")
text(x=2.42,y=20,labels = "2.42")
```



1.2. LINEAR REGRESSION WITH SPECTRA

```
dt.70 = merge(x,spectra,by="SEQ_ID") # Merge PCs with computed UAMS risk score
# run linear regression with UAMS risk score as dependent variable
```

```r
lm.70 = lm(data = dt.70[,-c("SEQ_ID","up","dw")], formula = score ~ .)
summary(lm.70)
```

```
##
## Call:
## lm(formula = score ~ ., data = dt.70[, -c("SEQ_ID", "up", "dw")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -0.7090 -0.1258 -0.0016  0.1218  0.5628
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  1.996877   0.007260 275.040  < 2e-16 ***
## PC1_SD      -0.059253   0.007265  -8.156 1.52e-15 ***
## PC2_SD      -0.162142   0.007265 -22.318  < 2e-16 ***
## PC3_SD       0.208251   0.007265  28.665  < 2e-16 ***
## PC4_SD       0.181338   0.007265  24.960  < 2e-16 ***
## PC5_SD      -0.213664   0.007265 -29.410  < 2e-16 ***
## PC6_SD      -0.057146   0.007265  -7.866 1.33e-14 ***
## PC7_SD      -0.029545   0.007265  -4.067 5.29e-05 ***
## PC8_SD       0.029619   0.007265   4.077 5.07e-05 ***
## PC9_SD      -0.188253   0.007265 -25.912  < 2e-16 ***
## PC10_SD     -0.058484   0.007265  -8.050 3.38e-15 ***
## PC11_SD      0.029629   0.007265   4.078 5.04e-05 ***
## PC12_SD      0.007368   0.007265   1.014 0.310827
## PC13_SD     -0.060472   0.007265  -8.324 4.22e-16 ***
## PC14_SD     -0.064130   0.007265  -8.827  < 2e-16 ***
## PC15_SD      0.047910   0.007265   6.595 8.21e-11 ***
## PC16_SD     -0.033932   0.007265  -4.671 3.58e-06 ***
## PC17_SD      0.058003   0.007265   7.984 5.54e-15 ***
## PC18_SD      0.052007   0.007265   7.159 2.00e-12 ***
## PC19_SD      0.015787   0.007265   2.173 0.030106 *
## PC20_SD      0.009642   0.007265   1.327 0.184848
## PC21_SD     -0.001090   0.007265  -0.150 0.880814
## PC22_SD      0.004206   0.007265   0.579 0.562845
## PC23_SD      0.036224   0.007265   4.986 7.71e-07 ***
## PC24_SD      0.025605   0.007265   3.524 0.000451 ***
## PC25_SD     -0.081726   0.007265 -11.249  < 2e-16 ***
## PC26_SD      0.102514   0.007265  14.111  < 2e-16 ***
## PC27_SD     -0.024134   0.007265  -3.322 0.000939 ***
## PC28_SD     -0.028106   0.007265  -3.869 0.000119 ***
## PC29_SD      0.037324   0.007265   5.137 3.58e-07 ***
## PC30_SD      0.009218   0.007265   1.269 0.204927
## PC31_SD     -0.010259   0.007265  -1.412 0.158357
## PC32_SD     -0.020508   0.007265  -2.823 0.004890 **
## PC33_SD      0.043664   0.007265   6.010 2.93e-09 ***
## PC34_SD     -0.012859   0.007265  -1.770 0.077145 .
## PC35_SD      0.038937   0.007265   5.359 1.12e-07 ***
## PC36_SD      0.047091   0.007265   6.482 1.67e-10 ***
## PC37_SD     -0.009972   0.007265  -1.373 0.170303
## PC38_SD     -0.023609   0.007265  -3.250 0.001209 **
## PC39_SD     -0.003538   0.007265  -0.487 0.626439
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2011 on 727 degrees of freedom
## Multiple R-squared:  0.8637, Adjusted R-squared:  0.8564
## F-statistic: 118.1 on 39 and 727 DF,  p-value: < 2.2e-16
```

```
nsig = data.table(summary(lm.70)$coeff[-1,"Pr(>|t|)"]) %>%
  subset(V1<0.05) %>% nrow() # Count sig spectra in model
print(paste0(nsig," of the ",ncol(dt.70[,-c("SEQ_ID","up","dw")])-1,
             " spectra were significat at p < 0.05 in the combined model"))
```

```
## [1] "30 of the 39 spectra were significat at p < 0.05 in the combined model"
```

```
ajr2 = round(summary(lm.70)$adj.r.squared,digits = 2) # Print adjusted R^2
df1 = round(summary(lm.70)$fstatistic[2])
df2 = round(summary(lm.70)$fstatistic[3])
f = round(summary(lm.70)$fstatistic[1],digits = 1)
# Compute p-value from f-statistic
p = formatC(pf(f,df1,df2,lower.tail = F), format = "E", digits = 1)
print(paste0("adjusted R^2=",ajr2,", F(",df1,",",df2,")=",f,", p=",p))
```

```
## [1] "adjusted R^2=0.86, F(39,727)=118.1, p=2.4E-285"
```

```
# Overall p-value
df1 = summary(lm.70)$fstatistic[2]
df2 = summary(lm.70)$fstatistic[3]
f = summary(lm.70)$fstatistic[1]
lm.70$p = pf(f,df1,df2,lower.tail = F) # Compute p-value from f-statistic
```

1.3. Logistic regression, high v low risk score

```
dt.70$bin = as.factor(if_else(dt.70$score<2.424285,"low","high"))
# run logistic regression with UAMS risk score as dependent variable
glm.70 = glm(data = dt.70[,-c("SEQ_ID","up","dw","score")],
             formula = bin ~ ., family = "binomial")
summary(glm.70)
```

```
##
## Call:
## glm(formula = bin ~ ., family = "binomial", data = dt.70[, -c("SEQ_ID",
##     "up", "dw", "score")])
##
## Deviance Residuals:
##      Min        1Q    Median        3Q       Max
## -3.03959   0.00278   0.02741   0.13692   2.56439
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  4.714999   0.487091   9.680  < 2e-16 ***
## PC1_SD       0.576870   0.213378   2.704 0.006861 **
```

```
## PC2_SD       1.549356   0.230393   6.725 1.76e-11 ***
## PC3_SD      -2.376172   0.334821  -7.097 1.28e-12 ***
## PC4_SD      -2.009738   0.284549  -7.063 1.63e-12 ***
## PC5_SD       2.552088   0.322388   7.916 2.45e-15 ***
## PC6_SD       0.910069   0.210074   4.332 1.48e-05 ***
## PC7_SD       0.264035   0.178333   1.481 0.138720
## PC8_SD      -0.263619   0.199417  -1.322 0.186186
## PC9_SD       2.193338   0.295203   7.430 1.09e-13 ***
## PC10_SD      0.618844   0.204407   3.028 0.002466 **
## PC11_SD     -0.258329   0.179825  -1.437 0.150845
## PC12_SD     -0.258280   0.179032  -1.443 0.149119
## PC13_SD      0.434561   0.194266   2.237 0.025291 *
## PC14_SD      0.708697   0.212136   3.341 0.000835 ***
## PC15_SD     -0.569369   0.183282  -3.107 0.001893 **
## PC16_SD      0.366720   0.196485   1.866 0.061985 .
## PC17_SD     -0.477517   0.186908  -2.555 0.010624 *
## PC18_SD     -0.413935   0.194061  -2.133 0.032924 *
## PC19_SD     -0.047701   0.194652  -0.245 0.806412
## PC20_SD     -0.001356   0.208818  -0.006 0.994819
## PC21_SD     -0.231290   0.184791  -1.252 0.210704
## PC22_SD     -0.110904   0.197115  -0.563 0.573681
## PC23_SD     -0.579275   0.215540  -2.688 0.007198 **
## PC24_SD     -0.636399   0.214357  -2.969 0.002989 **
## PC25_SD      0.749075   0.228379   3.280 0.001038 **
## PC26_SD     -1.350406   0.234698  -5.754 8.73e-09 ***
## PC27_SD      0.518027   0.206062   2.514 0.011939 *
## PC28_SD      0.484236   0.195268   2.480 0.013144 *
## PC29_SD     -0.365076   0.192381  -1.898 0.057739 .
## PC30_SD     -0.349034   0.193245  -1.806 0.070892 .
## PC31_SD      0.407271   0.203307   2.003 0.045153 *
## PC32_SD      0.486281   0.199551   2.437 0.014815 *
## PC33_SD      0.014617   0.181881   0.080 0.935947
## PC34_SD      0.003407   0.193500   0.018 0.985953
## PC35_SD     -0.464092   0.192108  -2.416 0.015702 *
## PC36_SD     -0.348691   0.191198  -1.824 0.068195 .
## PC37_SD      0.222712   0.209220   1.064 0.287109
## PC38_SD      0.239257   0.192533   1.243 0.213985
## PC39_SD     -0.036171   0.202613  -0.179 0.858312
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 760.92  on 766  degrees of freedom
## Residual deviance: 206.25  on 727  degrees of freedom
## AIC: 286.25
##
## Number of Fisher Scoring iterations: 8
```

```
nsig = data.table(summary(glm.70)$coeff[-1,"Pr(>|z|)"]) %>%
  subset(V1<0.05) %>% nrow() # Count sig spectra in model
print(paste0(nsig," of the ",ncol(dt.70[,-c("SEQ_ID","up","dw","score")])-1,
             " spectra were significat at p < 0.05 in the combined model"))
```

```
## [1] "22 of the 39 spectra were significat at p < 0.05 in the combined model"
```

```
# Overall p-value
NLL = glm(data = dt.70[,-c("SEQ_ID","up","dw","score")], formula = bin ~ 1, family = "binomial")

pchisq(deviance(NLL)-deviance(glm.70),
                        df.residual(NLL)-df.residual(glm.70),
                        lower.tail=FALSE)
```

```
## [1] 2.183962e-92
```

1.4. **PLOT**

```
theme_set(theme_classic() +
            theme(legend.position="none",
                  legend.title = element_text(size=8),
                  legend.text = element_text(size=7),
                  axis.title=element_text(size=9),
                  axis.text=element_text(size=8)))
```

Actual v predicted values

```
fit = lm.70
actual_preds <- data.table(cbind(dt.70[,c("SEQ_ID","bin")],
                                 actual=fit$model$score,predicted=fit$fitted.values))
cor(actual_preds[,-c("SEQ_ID","bin")])
```

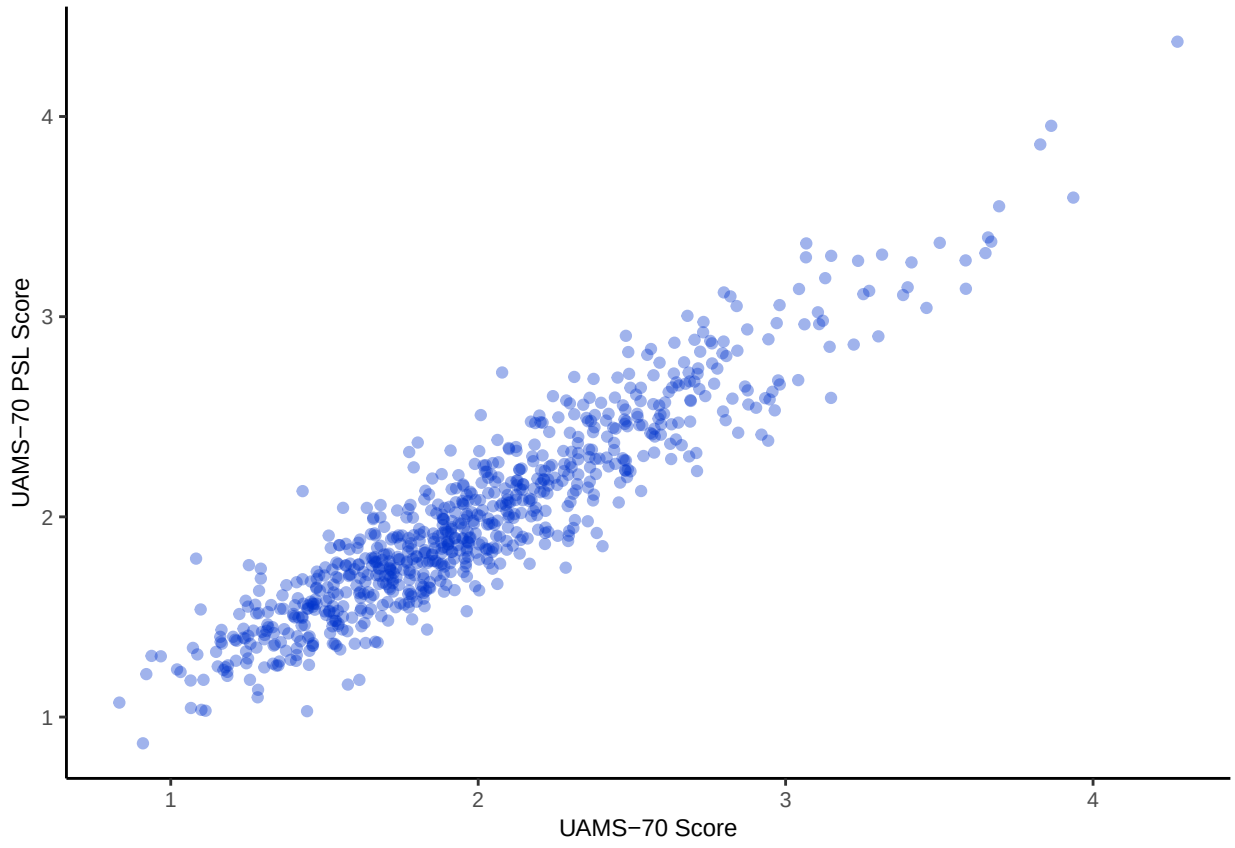
```
##              actual predicted
## actual    1.0000000 0.9293582
## predicted 0.9293582 1.0000000
```

```
# mean absolute percentage error
mean(abs((actual_preds$predicted - actual_preds$actual))/actual_preds$actual)
```

```
## [1] 0.08170039
```

Plot fitted values x actual values

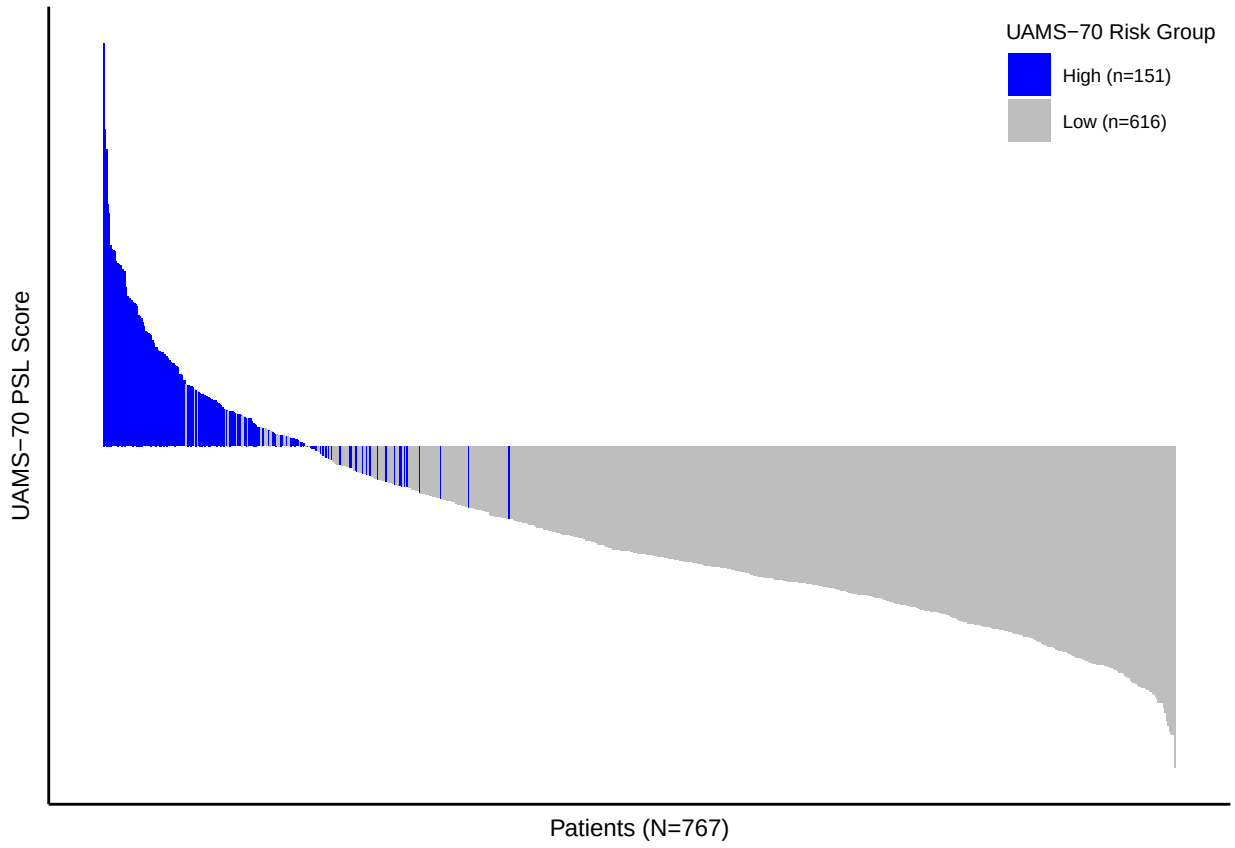
```
p1 = ggplot(actual_preds,aes(y = predicted, x = actual)) +
  geom_point(color = "#0033CC60") +
  xlab("UAMS-70 Score") + ylab("UAMS-70 PSL Score")
p1
```




```
ggsave(filename = "plots/UAMS-70_dot.pdf",plot = p1,width=3.75,height=3.75)
```

Waterfall High/Low risk score

```
setorder(actual_preds,-predicted)
# predicted score - high risk cutoff from clustering
w <- ggplot(actual_preds,aes(y=predicted-2.424285,x=1:nrow(actual_preds),fill=bin)) +
  geom_bar(stat="identity") +
  scale_fill_manual(values = c("blue","gray"),
                    name = "UAMS-70 Risk Group",
                    labels = c("High (n=151)", "Low (n=616)")) +
  xlab("Patients (N=767)") + ylab("UAMS-70 PSL Score") +
  scale_x_continuous(limits = c(0,nrow(actual_preds)+1),
                     breaks=seq(1,nrow(actual_preds),1)) +
  theme(legend.position = c(1,1), legend.justification = c(1,1),
        axis.text = element_blank(), axis.ticks = element_blank())
w
```




```
ggsave(filename = "plots/UAMS-70_waterfall.pdf",plot = w,width=3.75,height=2)
```

2. Br J Haematol. 2020 May 15. Development of a RNA sequencing-based prognostic gene signature in multiple myeloma

DOI: 10.1111/bjh.16744

Developed at the Shahid Bahonar University of Kerman, Kerman, Iran. I will refer to this prognostic score as SBUK-17.

Genes and regression coefficients from Table III.

2.1. COMPUTE SCORE IN COMMPASS DATA

```
DAT = exp_cbat # working from ComBat adjusted expression estimates in baseline samples

genes_17 = c("ADSS","BIRC5","CACYBP","CCT2","CCT3","CKS1B","CTPS1","ENO1",
             "GAPDH","KIAA0101","MSH6","NONO","PRKDC","RAN","SF3B4","TFB2M","UBE2A")
print(paste0("17 genes: ",length(intersect(genes_17,colnames(DAT))),
             " of ",length(genes_17)," genes in dataset"))
```

```
## [1] "17 genes: 10 of 17 genes in dataset"
```

```
coeff_17 = c(0.67,0.39,0.56,1.15,0.61,0.59,0.59,0.84,0.49,
             0.30,0.91,1.03,1.04,0.65,1.01,0.83,0.66)
```

```r
bjh17 = data.table(genes=genes_17,coeff=coeff_17)
# select genes in data
anno_17 = DAT %>% dplyr::select(intersect(genes_17,colnames(DAT)))

DAT$score = data.matrix(anno_17) %*%
  diag(bjh17[genes%in%intersect(genes_17,colnames(DAT))]$coeff) %>%
  data.table() %>% rowSums()

# Check algorithm
#coe = bjh17[genes%in%intersect(genes_17,colnames(DAT))][,coeff]
#tmp = DAT[SEQ_ID=="MMRF_1024_1_BM"] %>% select(intersect(genes_17,colnames(DAT)))
#sum(tmp*coe) == DAT[SEQ_ID=="MMRF_1024_1_BM"]$score
#tmp = DAT[50,] %>% select(intersect(genes_17,colnames(DAT)))
#sum(tmp*coe) == DAT[50,]$score
```

2.2. LINEAR REGRESSION WITH SPECTRA

```r
dt.17 = data.table(merge(DAT[,c("SEQ_ID","score")],spectra))

# run linear regression with risk score as dependent variable
lm.17 = lm(data = dt.17[,-c("SEQ_ID")], formula = score ~ .)
summary(lm.17)
```

```
##
## Call:
## lm(formula = score ~ ., data = dt.17[, -c("SEQ_ID")])
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.6458 -0.4568 -0.0132  0.4487  2.7058
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)  9.119144   0.026864 339.452  < 2e-16 ***
## PC1_SD      -0.976764   0.026882 -36.335  < 2e-16 ***
## PC2_SD      -1.281865   0.026882 -47.685  < 2e-16 ***
## PC3_SD       0.976906   0.026882  36.341  < 2e-16 ***
## PC4_SD       1.028264   0.026882  38.251  < 2e-16 ***
## PC5_SD      -1.151373   0.026882 -42.831  < 2e-16 ***
## PC6_SD      -0.119457   0.026882  -4.444 1.02e-05 ***
## PC7_SD      -0.090625   0.026882  -3.371 0.000788 ***
## PC8_SD      -0.047880   0.026882  -1.781 0.075309 .
## PC9_SD      -0.320212   0.026882 -11.912  < 2e-16 ***
## PC10_SD     -0.342421   0.026882 -12.738  < 2e-16 ***
## PC11_SD      0.311174   0.026882  11.576  < 2e-16 ***
## PC12_SD      0.101102   0.026882   3.761 0.000183 ***
## PC13_SD      0.001177   0.026882   0.044 0.965081
## PC14_SD     -0.403288   0.026882 -15.002  < 2e-16 ***
## PC15_SD     -0.319003   0.026882 -11.867  < 2e-16 ***
## PC16_SD      0.297694   0.026882  11.074  < 2e-16 ***
## PC17_SD     -0.088224   0.026882  -3.282 0.001080 **
## PC18_SD      0.373637   0.026882  13.899  < 2e-16 ***
## PC19_SD     -0.262150   0.026882  -9.752  < 2e-16 ***
```

```
## PC20_SD      0.034308   0.026882   1.276 0.202276
## PC21_SD     -0.068960   0.026882  -2.565 0.010508 *
## PC22_SD     -0.021063   0.026882  -0.784 0.433557
## PC23_SD     -0.020595   0.026882  -0.766 0.443853
## PC24_SD      0.214803   0.026882   7.991 5.26e-15 ***
## PC25_SD      0.037578   0.026882   1.398 0.162571
## PC26_SD     -0.016926   0.026882  -0.630 0.529137
## PC27_SD     -0.224481   0.026882  -8.351 3.42e-16 ***
## PC28_SD      0.069229   0.026882   2.575 0.010212 *
## PC29_SD      0.099440   0.026882   3.699 0.000233 ***
## PC30_SD      0.041684   0.026882   1.551 0.121425
## PC31_SD     -0.022349   0.026882  -0.831 0.406041
## PC32_SD      0.325325   0.026882  12.102  < 2e-16 ***
## PC33_SD      0.039846   0.026882   1.482 0.138705
## PC34_SD     -0.042696   0.026882  -1.588 0.112657
## PC35_SD     -0.061672   0.026882  -2.294 0.022063 *
## PC36_SD     -0.022596   0.026882  -0.841 0.400868
## PC37_SD      0.043611   0.026882   1.622 0.105166
## PC38_SD     -0.044375   0.026882  -1.651 0.099226 .
## PC39_SD      0.182827   0.026882   6.801 2.17e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.744 on 727 degrees of freedom
## Multiple R-squared:  0.9314, Adjusted R-squared:  0.9277
## F-statistic: 252.9 on 39 and 727 DF,  p-value: < 2.2e-16
```

```
nsig = data.table(summary(lm.17)$coeff[-1,"Pr(>|t|)"]) %>%
  subset(V1<0.05) %>% nrow() # Count sig spectra in model
print(paste0(nsig," of the ",ncol(dt.17[,-c("SEQ_ID","score")]),
             " spectra were significat at p < 0.05 in the combined model"))
```

```
## [1] "25 of the 39 spectra were significat at p < 0.05 in the combined model"
```

```
ajr2 = round(summary(lm.17)$adj.r.squared,digits = 2) # Print adjusted R^2
df1 = round(summary(lm.17)$fstatistic[2])
df2 = round(summary(lm.17)$fstatistic[3])
f = round(summary(lm.17)$fstatistic[1],digits = 1)
# Compute p-value from f-statistic
p = formatC(pf(f,df1,df2,lower.tail = F), format = "E", digits = 1)
print(paste0("adjusted R^2=",ajr2,", F(",df1,",",df2,")=",f,", p=",p))
```

```
## [1] "adjusted R^2=0.93, F(39,727)=252.9, p=0.0E+00"
```

```
# Overall p-value
df1 = summary(lm.17)$fstatistic[2]
df2 = summary(lm.17)$fstatistic[3]
f = summary(lm.17)$fstatistic[1]
lm.17$p = pf(f,df1,df2,lower.tail = F) # Compute p-value from f-statistic
```

2.3. PLOT

Actual v predicted values

```
dt = dt.17
fit = lm.17
actual_preds <- data.table(cbind(dt[,"SEQ_ID"],
                                 actual=fit$model$score,
                                 predicted=fit$fitted.values))
cor(actual_preds[,-"SEQ_ID"])
```

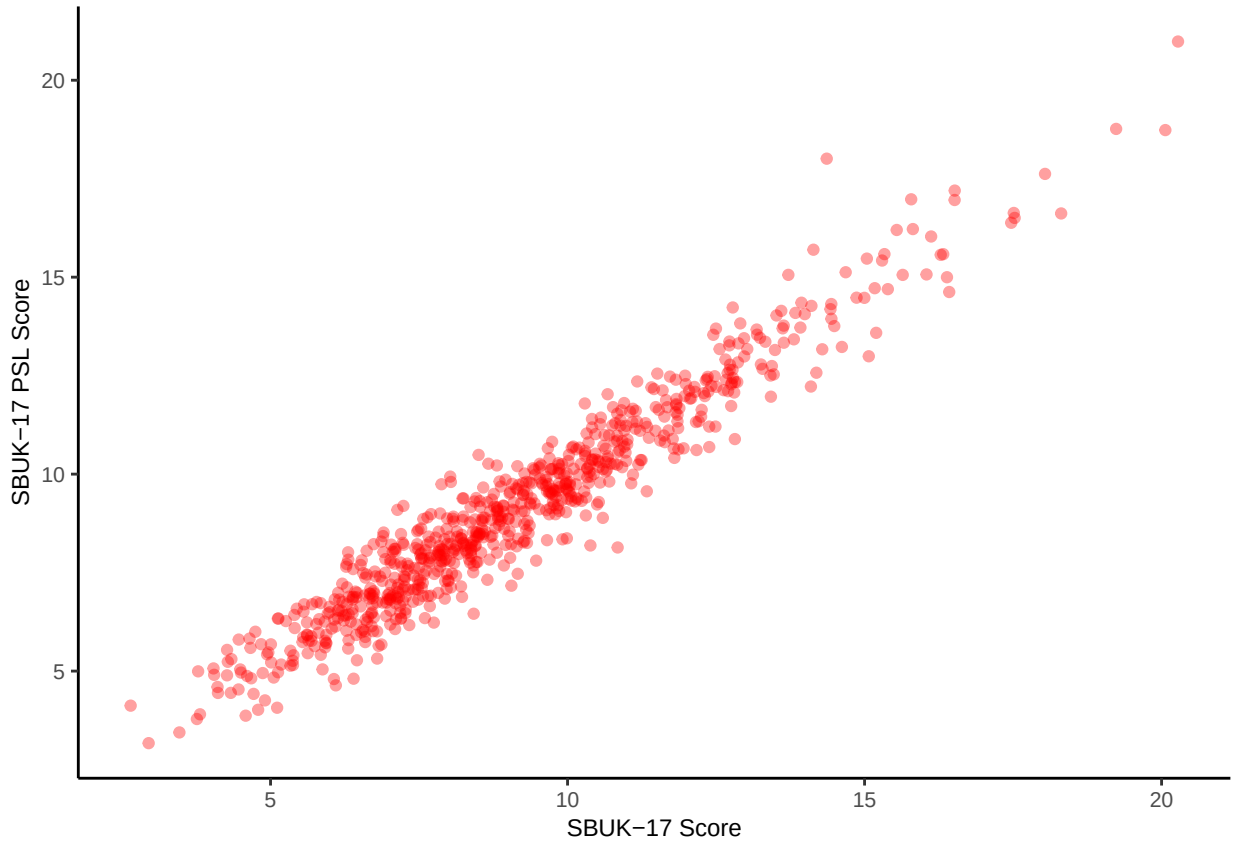
```
##               actual predicted
## actual    1.0000000 0.9650687
## predicted 0.9650687 1.0000000
```

```
# mean absolute percentage error
mean(abs((actual_preds$predicted - actual_preds$actual))/actual_preds$actual)
```

```
## [1] 0.06692651
```

Plot fitted values x actual values

```
p1 = ggplot(actual_preds,aes(y = predicted, x = actual)) +
  geom_point(color = "#FF000060") +
  xlab("SBUK-17 Score") + ylab("SBUK-17 PSL Score")
p1
```



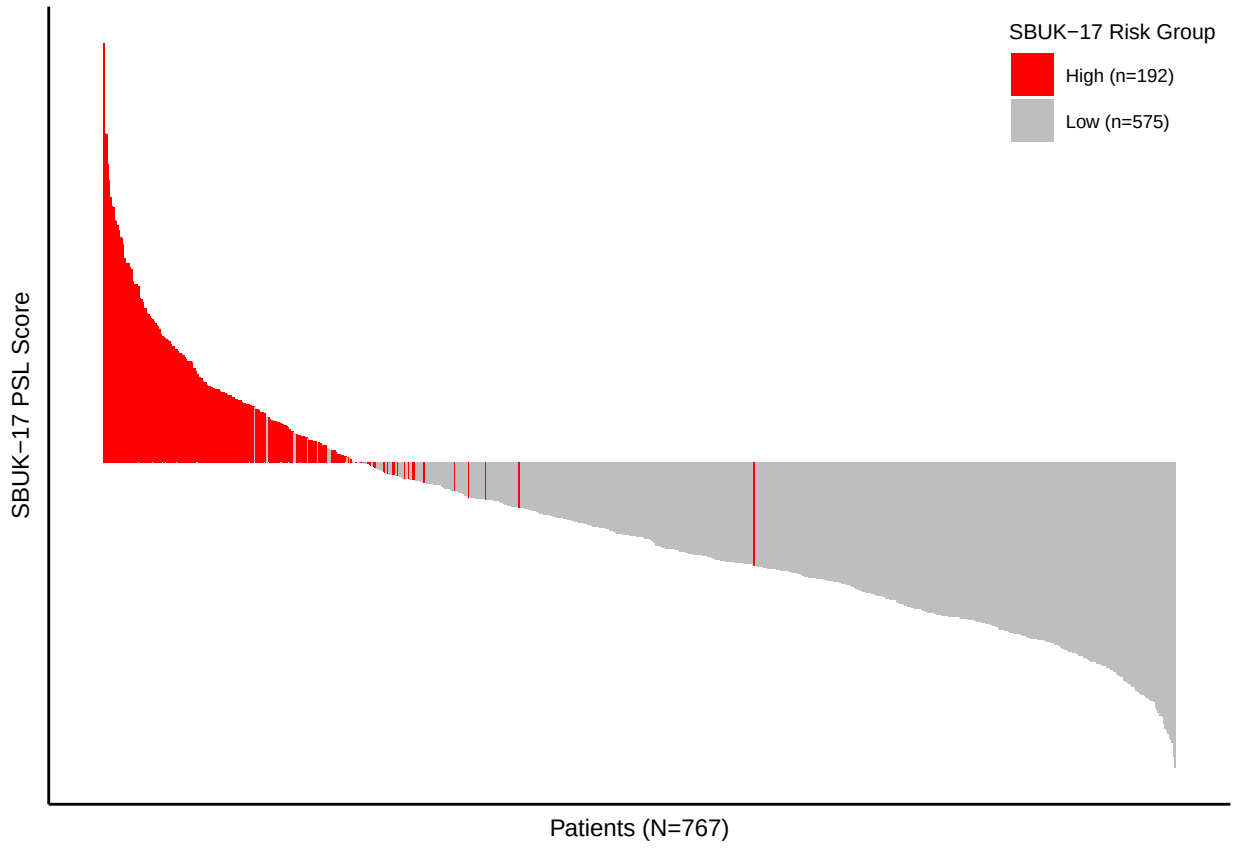
```
ggsave(filename = "plots/SBUK-17_dot.pdf",plot = p1,width=3.75,height=3.75)
```

Waterfall High/Low risk score "At the next step, the patients were dichotomized into a high-risk and a low-risk category based on the 75th percentile of the risk scores and survival time was compared between the two groups using the Kaplan–Meier analysis and the log-rank test at a P value of less than 0.01. SPSS 16.0 and Survival (http://cran.r-project.org/package=survival) packages were used to execute the survival analysis. The 75th percentile cut-off was based on the proportion of patients in the training set who had a survival time of more than two years." (DOI: 10.1111/bjh.16744)

```
# find 75th percentile of SBUK-17 score
high.cut = quantile(actual_preds$actual,.75)
# bin patients in high- or low-risk groups
actual_preds$bin = as.factor(if_else(actual_preds$actual<high.cut,"low","high"))
summary(actual_preds)
```

```
##             SEQ_ID        actual         predicted        bin
##  MMRF_1021_1_BM:  1   Min.   : 2.645   Min.   : 3.170   high:192
##  MMRF_1024_1_BM:  1   1st Qu.: 7.119   1st Qu.: 7.135   low :575
##  MMRF_1029_1_BM:  1   Median : 8.759   Median : 8.838
##  MMRF_1030_1_BM:  1   Mean   : 9.119   Mean   : 9.119
##  MMRF_1031_1_BM:  1   3rd Qu.:10.675   3rd Qu.:10.595
##  MMRF_1032_1_BM:  1   Max.   :20.277   Max.   :20.983
##  (Other)       :761
```

```
setorder(actual_preds,-predicted)
# predicted score - high risk cutoff from clustering
w <- ggplot(actual_preds,aes(y=predicted-high.cut,x=1:nrow(actual_preds),fill=bin)) +
  geom_bar(stat="identity") +
  scale_fill_manual(values = c("red","gray"), name = "SBUK-17 Risk Group",
                    labels = c("High (n=192)", "Low (n=575)")) +
  xlab("Patients (N=767)") + ylab("SBUK-17 PSL Score") +
  scale_x_continuous(limits = c(0,nrow(actual_preds)+1),
                     breaks=seq(1,nrow(actual_preds),1)) +
  theme(legend.position = c(1,1), legend.justification = c(1,1),
        axis.text = element_blank(), axis.ticks = element_blank())
w
```




```
ggsave(filename = "plots/SBUK-17_waterfall.pdf",plot = w,width=3.75,height=2)
```

Save model results

```
save(lm.17,lm.70,file = "rdata/lm.expression-scores.rdata")
```