FRET Classifier and Validation

P Barber

27 September 2019

Load patient data

```
load("COIN_Final.Rdata")
```

Encode, Impute Covariates

Isolate those covariates known to be needed for Risk Signatures etc. Impute by replacing NA by the average of other patients.

```
encoded_data <- with(patient_data, data.frame(SUMLES, WBC, NEUT, SHB0, FRET, FRET.cohort))

encoded_data$RAS.Mutation <- ifelse(patient_data$RAS=="Mutation", 1, 0)
encoded_data$mlivonly <- ifelse(patient_data$mlivonly=="Yes", 1, 0)
encoded_data$PIK3CA.Mutation <- ifelse(patient_data$PIK3CA=="Mutation", 1, 0)

# Get means from the 398
df <- encoded_data[encoded_data$FRET.cohort==1,]
df <- subset(df, select = -c(FRET.cohort))
trained_means <- colMeans(df, na.rm = T)

# remove FREt.cohort col
encoded_data <- subset(encoded_data, select = -c(FRET.cohort))

# Apply to all data
#encoded_data[] <- lapply(encoded_data, NA2mean)
encoded_data[] <- replaceNA(encoded_data, trained_means)
```

Load risk signature classifier

```
Class_Predictor <-
  read.table("SP_Classifier_Score/COIN_withFRET_July16/RiskScore_formula.txt",
                        sep = '*', col.names = c("Weight", "Covariate"),
                        stringsAsFactors = F, skip = 2, fill = T)

kable(Class_Predictor)
```

Weight	Covariate
S=(0.006338408651)	SUMLES
+ (-0.142238099638)	WBC
+ (0.211925854316)	NEUT
+ (0.769846490411)	SHB0
+ (9.895611265279)	FRET
+ (-0.728162098653)	mlivonly
+ (-1.151128618935)	PIK3CA.Mutation
- (0.697856603639)	

```r
# Load output from Saddle Point Signature and tidy up
Class_Predictor_Betas <-
  read.table("SP_Classifier_Score/COIN_withFRET_July16/SETCV_L2/betas_with_errors_optimised.txt",
             sep = ' ', col.names = c("Covariate", "beta", "pm", "error",
                                      "rank", "z.score", "p.value"),
             stringsAsFactors = F)

Class_Predictor_Betas <- subset(Class_Predictor_Betas, select=-pm)
Class_Predictor_Betas$Covariate <- sub(":$", "", Class_Predictor_Betas$Covariate)
Class_Predictor_Betas$beta <- as.numeric(
  sub("^beta\\[\\d+]=", "", Class_Predictor_Betas$beta))
Class_Predictor_Betas$rank <- last_number(Class_Predictor_Betas$rank,
                                           decimals = F, negs = F)
Class_Predictor_Betas$z.score <- last_number(Class_Predictor_Betas$z.score,
                                           decimals = T, negs = F)
Class_Predictor_Betas$p.value <- last_number(Class_Predictor_Betas$p.value,
                                           decimals = T, negs = F)

# Create new cols as required
Class_Predictor_Betas$description <- Class_Predictor_Betas$Covariate
Class_Predictor_Betas$description <-
  sub("SUMLES", "Sum of longest diameter", Class_Predictor_Betas$description)
Class_Predictor_Betas$description <-
  sub("WBC", "White blood cell count", Class_Predictor_Betas$description)
Class_Predictor_Betas$description <-
  sub("NEUT", "Neutrophil count", Class_Predictor_Betas$description)
Class_Predictor_Betas$description <-
  sub("SHBO", "Haemaglobin (CTC grade)", Class_Predictor_Betas$description)
#Class_Predictor_Betas$description <-
#  sub("FRET", "HER2-HER3 FRET efficiency", Class_Predictor_Betas$description)
Class_Predictor_Betas$description <-
  sub("mlivonly", "Liver-only metastases", Class_Predictor_Betas$description)
Class_Predictor_Betas$description <-
  sub("PIK3CA.Mutation", "PIK3CA mutation", Class_Predictor_Betas$description)

Class_Predictor_Betas$HR <- exp(Class_Predictor_Betas$beta)
Class_Predictor_Betas$HR.U95.CI <- exp(Class_Predictor_Betas$beta+Class_Predictor_Betas$error)
Class_Predictor_Betas$HR.L95.CI <- exp(Class_Predictor_Betas$beta-Class_Predictor_Betas$error)

# Add the constant as new row
Class_Predictor_Betas <- rbind(Class_Predictor_Betas,
                               list("Constant", NA, NA, NA, NA, 0, "Constant", 1, 1, 1))

Class_Predictor_Betas$Weight <- Class_Predictor$Weight
Class_Predictor_Betas$Weight <- str_remove(Class_Predictor_Betas$Weight, "S=\\(")
Class_Predictor_Betas$Weight <- str_remove(Class_Predictor_Betas$Weight, "\\+ \\(")
Class_Predictor_Betas$Weight <- sub("\\- \\(", "-", Class_Predictor_Betas$Weight)
Class_Predictor_Betas$Weight <- str_remove(Class_Predictor_Betas$Weight, "\\)")
Class_Predictor_Betas$Weight <- as.numeric(Class_Predictor_Betas$Weight)

mydf <- data.frame(
    Covariate = Class_Predictor_Betas$description,
    #HazardRatio = signif(Class_Predictor_Betas$HR, digits = 3),
    #HazardLower = signif(Class_Predictor_Betas$HR.L95.CI, digits = 3),
    #HazardUpper = signif(Class_Predictor_Betas$HR.U95.CI, digits = 3),
#    Pvalue = signif(Class_Predictor_Betas$p.value, digits = 2),
    Pvalue = sapply(Class_Predictor_Betas$p.value, JNCI_pvals),
    Rank = Class_Predictor_Betas$rank,
    Weight = round(Class_Predictor_Betas$Weight, digits = 5),
    stringsAsFactors=FALSE
)

mydf <- mydf[order(mydf$Rank),]

#plotHRTable(mydf, useClass = F, useWeight = T, useRank = T, xmin = -4, xmax = 5,
#            cex = 1.0, rowspacing = 0.5, col = 1,
#            hrTicks = c(0.5, 1, 2),
#            classpos = -0.8, hrpos = 1.6, pvaluepos = 3.4,
#            weightpos = 4.47, rankpos = -0.7)

# Put blank entries for the constant
mydf$Pvalue[8] = ""
mydf$Rank[8] = ""

# Reorder cols
mydf <- subset(mydf, select = c(Covariate, Rank, Pvalue, Weight))

kable(mydf, row.names=F, col.names = c("Covariate", "Rank", "p-value", "Weight"), booktabs = T) %>%
  kable_styling(latex_options = "striped")
```

Covariate	Rank	p-value	Weight
Sum of longest diameter	1	<0.001***	0.00634
Neutrophil count	2	0.01*	0.21193
White blood cell count	3	0.04*	-0.14224
Haemaglobin (CTC grade)	4	<0.001***	0.76985
PIK3CA mutation	5	<0.001***	-1.15113
Liver-only metastases	6	<0.001***	-0.72816
FRET	7	0.003**	9.89561
Constant			-0.69786

Calculate Risk Score

```
patient_data$Class_Pred_Score <- calculateRiskScore(Class_Predictor, encoded_data)
```

ROC

ROC curve for the 398 training data, these are the ones that we have ground truth FRET OS Class for.

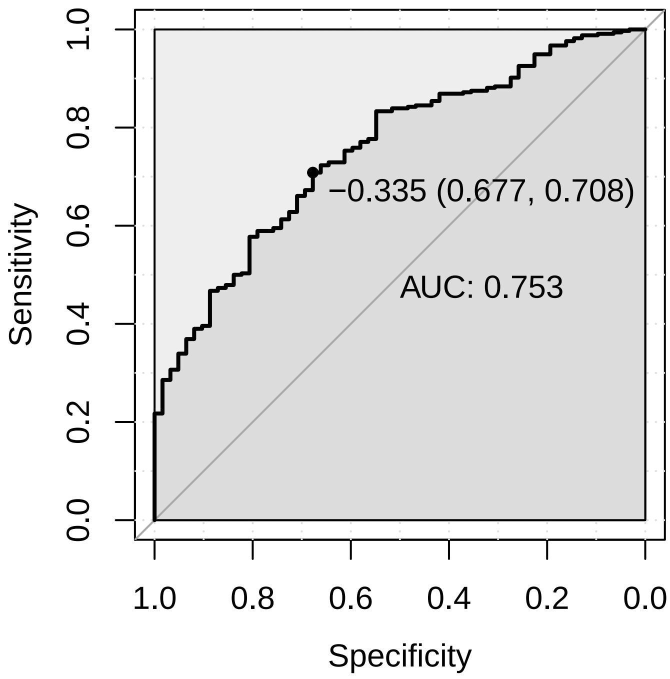
```
library(pROC)
```

```
## Warning: package 'pROC' was built under R version 3.5.3

## Type 'citation("pROC")' for a citation.
```

```
##
## Attaching package: 'pROC'

## The following objects are masked from 'package:stats':
##
##     cov, smooth, var
```

```
pROC_obj <- roc(patient_data$Class.FRET.OS, patient_data$Class_Pred_Score,
            smoothed = TRUE,
            # arguments for ci
            ci=FALSE, ci.alpha=0.9, stratified=FALSE,
            # arguments for plot
            plot=TRUE, auc.polygon=TRUE, max.auc.polygon=TRUE, grid=TRUE,
            print.auc=TRUE, print.thres=TRUE)
```

```
## Setting levels: control = 1, case = 2

## Setting direction: controls < cases
```



Choose an optimal threshold

```
optimal <- coords(pROC_obj, x="best", input="threshold", best.method="youden")
```

```
## Warning in coords.roc(pROC_obj, x = "best", input = "threshold",
## best.method = "youden"): An upcoming version of pROC will set the
## 'transpose' argument to FALSE by default. Set transpose = TRUE explicitly
## to keep the current behavior, or transpose = FALSE to adopt the new one
## and silence this warning. Type help(coords_transpose) for additional
## information.
```

```
optimal
```

```
##   threshold specificity sensitivity
##  -0.3347168   0.6774194   0.7083333
```

```
threshold = optimal[1]
```

Classify all patients

```
patient_data$Pred.Class <- ifelse(patient_data$Class_Pred_Score>threshold, 2, 1)
```

Confusion table for the 398

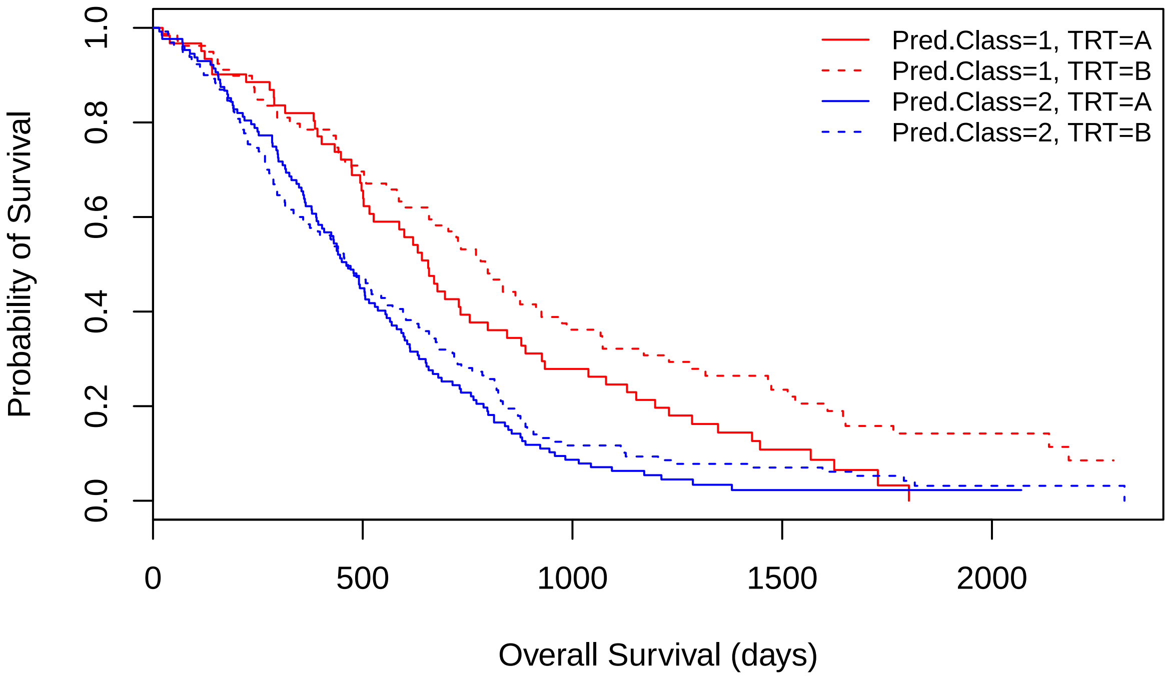
```
GT <- patient_data$Class.FRET.OS
Pred <- patient_data$Pred.Class

table(GT, Pred)
```

```
##    Pred
## GT    1   2
##   1  42  20
##   2  98 238
```

Analyse 398 training set

```
data <- patient_data[patient_data$FRET.cohort==1,]
```

```
library(survival)

data$SurvObj.pfs <- with(data, Surv(pfstime, pfsevent))
data$SurvObj.os <- with(data, Surv(ostime, osevent))
col=c("red", "red", "blue", "blue")
lty=c(1,2,1,2)

km <- survfit(SurvObj.os ~ Pred.Class + TRT, data=data)
plot(km, col=col, xlab="Overall Survival (days)", ylab="Probability of Survival", lty = lty)
legend("topright", col=col, legend = names(km$strata), lty = lty, bty="n", cex=0.8)
```



```r
print(km)
```

```
## Call: survfit(formula = SurvObj.os ~ Pred.Class + TRT, data = data)
##
##                         n events median 0.95LCL 0.95UCL
## Pred.Class=1, TRT=A  61     58    656     516     844
## Pred.Class=1, TRT=B  79     67    792     658     975
## Pred.Class=2, TRT=A 128    123    461     403     536
## Pred.Class=2, TRT=B 130    125    461     374     571
```

```r
survdiff(SurvObj.os ~ Pred.Class, data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os ~ Pred.Class, data = data)
##
##                  N Observed Expected (O-E)^2/E (O-E)^2/V
## Pred.Class=1 140      125      175      14.5      28.6
## Pred.Class=2 258      248      198      12.9      28.6
##
##  Chisq= 28.6  on 1 degrees of freedom, p= 9e-08
```

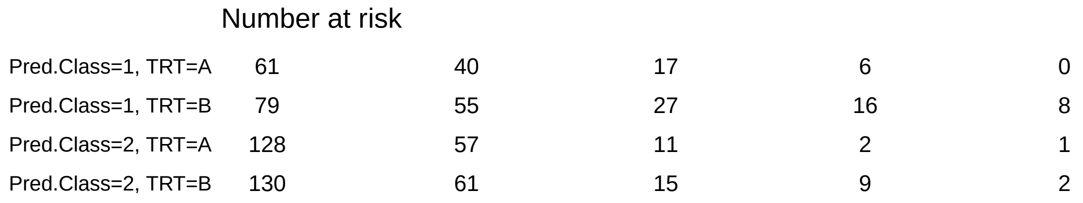
```r
survdiff(SurvObj.os[Pred.Class==1] ~ TRT[Pred.Class==1], data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 1] ~ TRT[Pred.Class ==
##     1], data = data)
##
##                         N Observed Expected (O-E)^2/E (O-E)^2/V
## TRT[Pred.Class == 1]=A 61       58     47.2      2.46      4.07
## TRT[Pred.Class == 1]=B 79       67     77.8      1.49      4.07
##
##  Chisq= 4.1  on 1 degrees of freedom, p= 0.04
```

```r
survdiff(SurvObj.os[Pred.Class==2] ~ TRT[Pred.Class==2], data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 2] ~ TRT[Pred.Class ==
##     2], data = data)
##
##                          N Observed Expected (O-E)^2/E (O-E)^2/V
## TRT[Pred.Class == 2]=A 128      123      116     0.439     0.842
## TRT[Pred.Class == 2]=B 130      125      132     0.385     0.842
##
##  Chisq= 0.8  on 1 degrees of freedom, p= 0.4
```

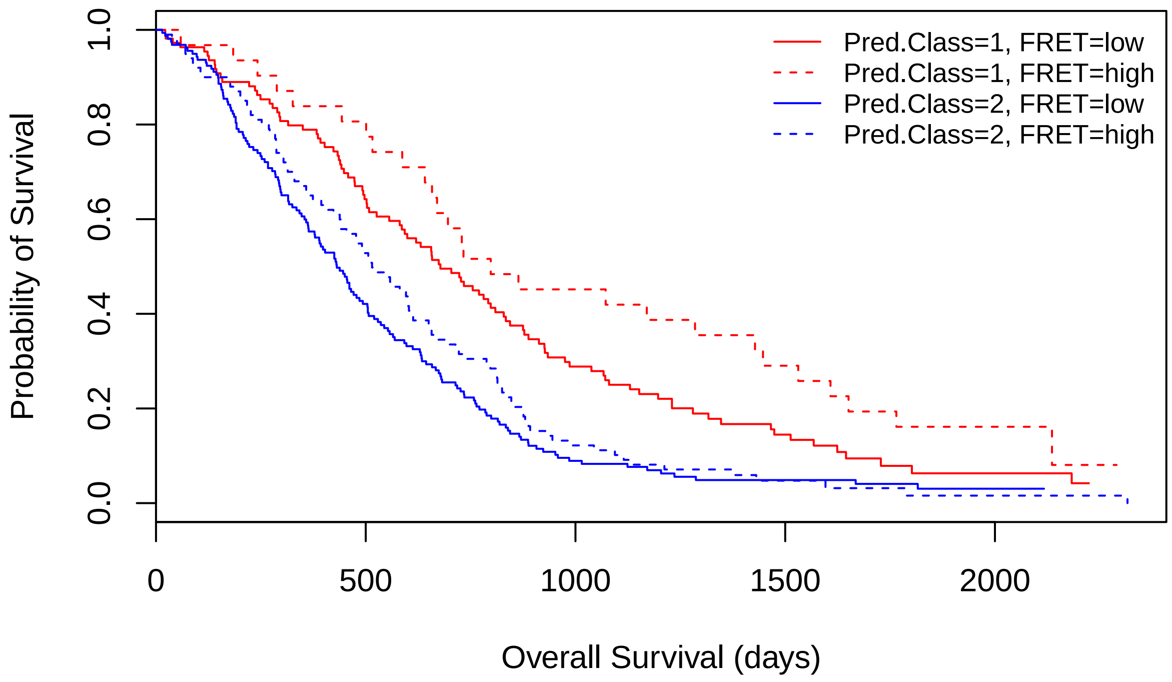
```r
ggrisktable(km, data=data) + theme_cleantable()
```



Number at risk

Pred.Class=1, TRT=A	61	40	17	6	0
Pred.Class=1, TRT=B	79	55	27	16	8
Pred.Class=2, TRT=A	128	57	11	2	1
Pred.Class=2, TRT=B	130	61	15	9	2

```r
#Split FRET by Tertiles
breaks <- quantile(data$FRET, probs = c(0.0, 0.67, 1.0), na.rm = T)
data$FRETeff <- cut(data$FRET, breaks = breaks, labels = c("low", "high"),
                    include.lowest = T)
data$FRET_raw <- data$FRET
data$FRET <- data$FRETeff

km <- survfit(SurvObj.os ~ Pred.Class + FRET, data=data)
plot(km, col=col, lty=lty, xlab="Overall Survival (days)", ylab="Probability of Survival")
legend("topright", col=col, legend = names(km$strata), lty=lty, bty="n", cex=0.8)
```



```
print(km)
```

```
## Call: survfit(formula = SurvObj.os ~ Pred.Class + FRET, data = data)
##
##                            n events median 0.95LCL 0.95UCL
## Pred.Class=1, FRET=low  109     98    678     586     829
## Pred.Class=1, FRET=high  31     27    798     658    1531
## Pred.Class=2, FRET=low  158    151    431     378     504
## Pred.Class=2, FRET=high 100     97    515     441     613
```

```
survdiff(SurvObj.os ~ Pred.Class, data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os ~ Pred.Class, data = data)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Pred.Class=1 140      125      175      14.5      28.6
## Pred.Class=2 258      248      198      12.9      28.6
##
##  Chisq= 28.6  on 1 degrees of freedom, p= 9e-08
```

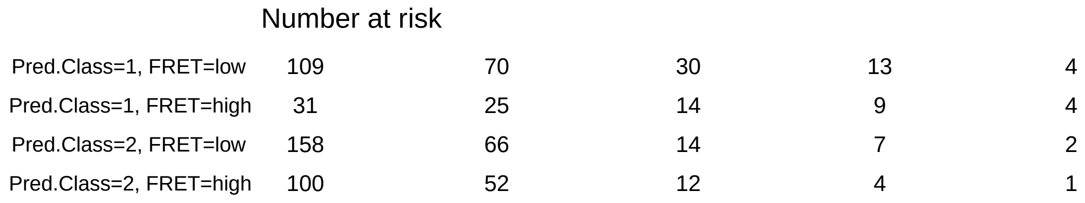
```
survdiff(SurvObj.os[Pred.Class==1] ~ FRET[Pred.Class==1], data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 1] ~ FRET[Pred.Class ==
##     1], data = data)
##
##                              N Observed Expected (O-E)^2/E (O-E)^2/V
## FRET[Pred.Class == 1]=low  109       98     89.1      0.88      3.15
## FRET[Pred.Class == 1]=high  31       27     35.9      2.19      3.15
##
##  Chisq= 3.2  on 1 degrees of freedom, p= 0.08
```

```
survdiff(SurvObj.os[Pred.Class==2] ~ FRET[Pred.Class==2], data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 2] ~ FRET[Pred.Class ==
##     2], data = data)
##
##                               N Observed Expected (O-E)^2/E (O-E)^2/V
## FRET[Pred.Class == 2]=low  158      151      141     0.647      1.52
## FRET[Pred.Class == 2]=high 100       97      107     0.858      1.52
##
##  Chisq= 1.5  on 1 degrees of freedom, p= 0.2
```

```
ggrisktable(km, data=data) + theme_cleantable()
```



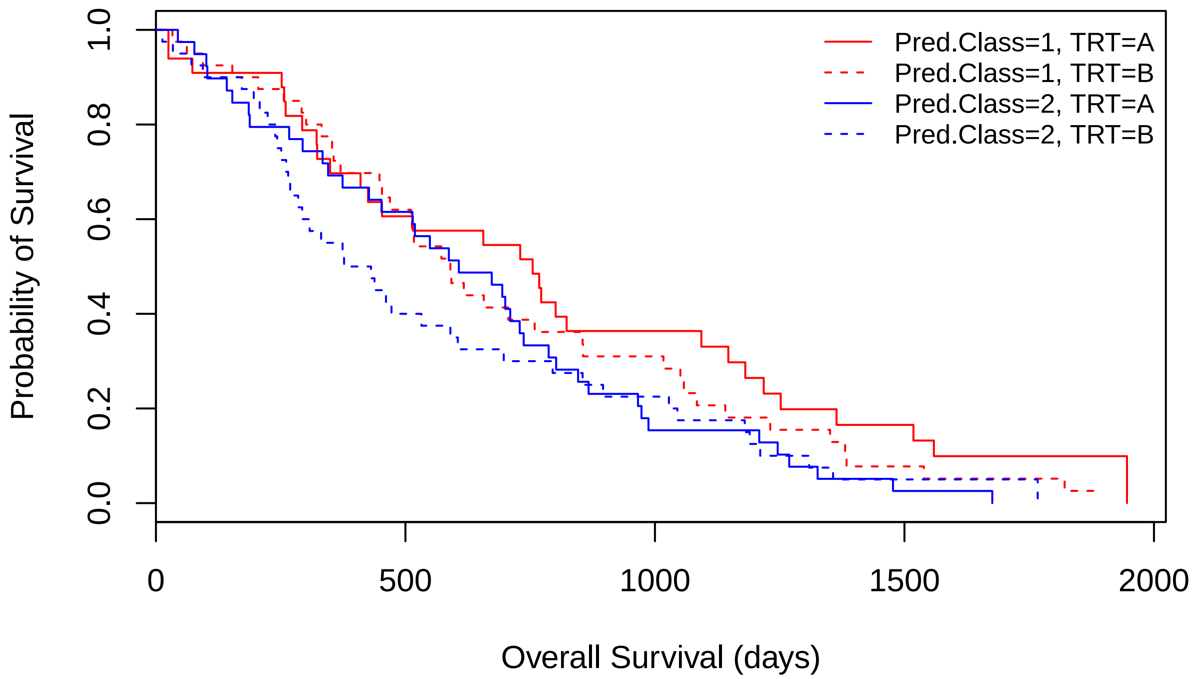
Number at risk

Pred.Class=1, FRET=low	109	70	30	13	4
Pred.Class=1, FRET=high	31	25	14	9	4
Pred.Class=2, FRET=low	158	66	14	7	2
Pred.Class=2, FRET=high	100	52	12	4	1

Analyse COIN III 152 validation set

```
data <- patient_data[patient_data$FRET.cohort==2,]
```

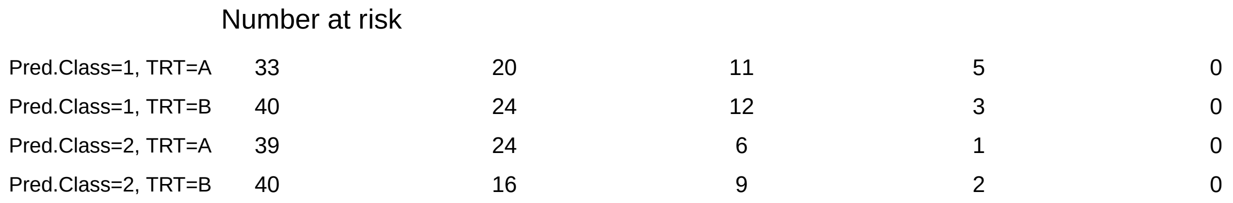
```
library(survival)

data$SurvObj.pfs <- with(data, Surv(pfstime, pfsevent))
data$SurvObj.os <- with(data, Surv(ostime, osevent))
col=c("red", "red", "blue", "blue")
lty=c(1,2,1,2)

km <- survfit(SurvObj.os ~ Pred.Class + TRT, data=data)
plot(km, col=col, xlab="Overall Survival (days)", ylab="Probability of Survival", lty = lty)
legend("topright", col=col, legend = names(km$strata), lty = lty, bty="n", cex=0.8)
```



```
print(km)
```

```
## Call: survfit(formula = SurvObj.os ~ Pred.Class + TRT, data = data)
##
##                          n events median 0.95LCL 0.95UCL
## Pred.Class=1, TRT=A 33     30    755     425    1181
## Pred.Class=1, TRT=B 40     38    590     469     856
## Pred.Class=2, TRT=A 39     39    607     452     787
## Pred.Class=2, TRT=B 40     39    404     285     697
```

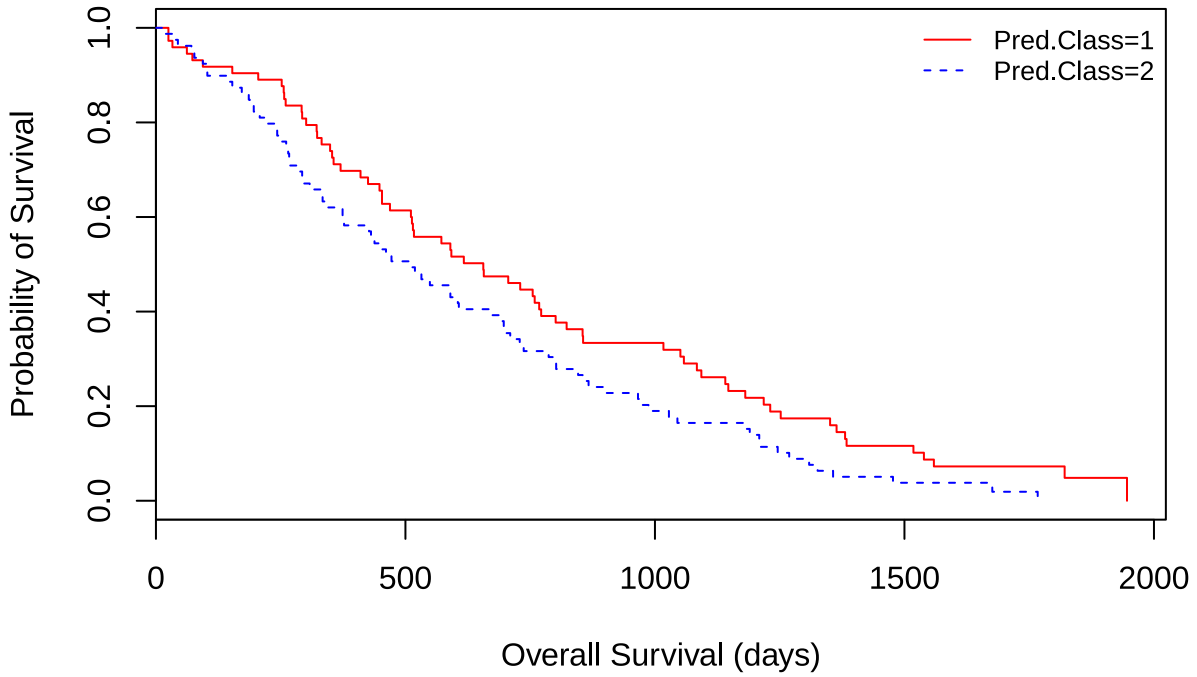
```
ggrisktable(km, data=data) + theme_cleantable()
```



Number at risk

Pred.Class=1, TRT=A	33	20	11	5	0
Pred.Class=1, TRT=B	40	24	12	3	0
Pred.Class=2, TRT=A	39	24	6	1	0
Pred.Class=2, TRT=B	40	16	9	2	0

```
col=c("red", "blue")
lty=c(1,2)
km <- survfit(SurvObj.os ~ Pred.Class, data=data)
plot(km, col=col, xlab="Overall Survival (days)", ylab="Probability of Survival", lty = lty)
legend("topright", col=col, legend = names(km$strata), lty = lty, bty="n", cex=0.8)
```



```
print(km)
```

```
## Call: survfit(formula = SurvObj.os ~ Pred.Class, data = data)
##
##                 n events median 0.95LCL 0.95UCL
## Pred.Class=1 73     68    656     511     823
## Pred.Class=2 79     78    514     374     697
```

```
survdiff(SurvObj.os ~ Pred.Class, data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os ~ Pred.Class, data = data)
##
##                 N Observed Expected (O-E)^2/E (O-E)^2/V
## Pred.Class=1 73       68     80.4      1.90      4.35
## Pred.Class=2 79       78     65.6      2.33      4.35
##
##  Chisq= 4.4  on 1 degrees of freedom, p= 0.04
```

```
survdiff(SurvObj.os[Pred.Class==1] ~ TRT[Pred.Class==1], data=data)
```

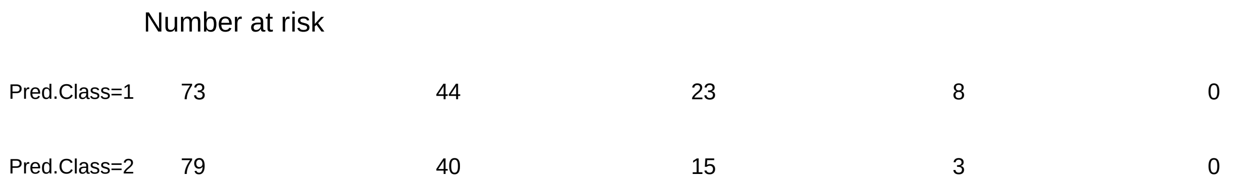
```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 1] ~ TRT[Pred.Class ==
##     1], data = data)
##
##                           N Observed Expected (O-E)^2/E (O-E)^2/V
## TRT[Pred.Class == 1]=A 33       30     33.6     0.395     0.804
## TRT[Pred.Class == 1]=B 40       38     34.4     0.387     0.804
```

```
##
##  Chisq= 0.8  on 1 degrees of freedom, p= 0.4
```

```r
survdiff(SurvObj.os[Pred.Class==2] ~ TRT[Pred.Class==2], data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 2] ~ TRT[Pred.Class ==
##     2], data = data)
##
##                          N Observed Expected (O-E)^2/E (O-E)^2/V
## TRT[Pred.Class == 2]=A 39       39     40.2    0.0385    0.0814
## TRT[Pred.Class == 2]=B 40       39     37.8    0.0410    0.0814
##
##  Chisq= 0.1  on 1 degrees of freedom, p= 0.8
```

```r
ggrisktable(km, data=data) + theme_cleantable()
```



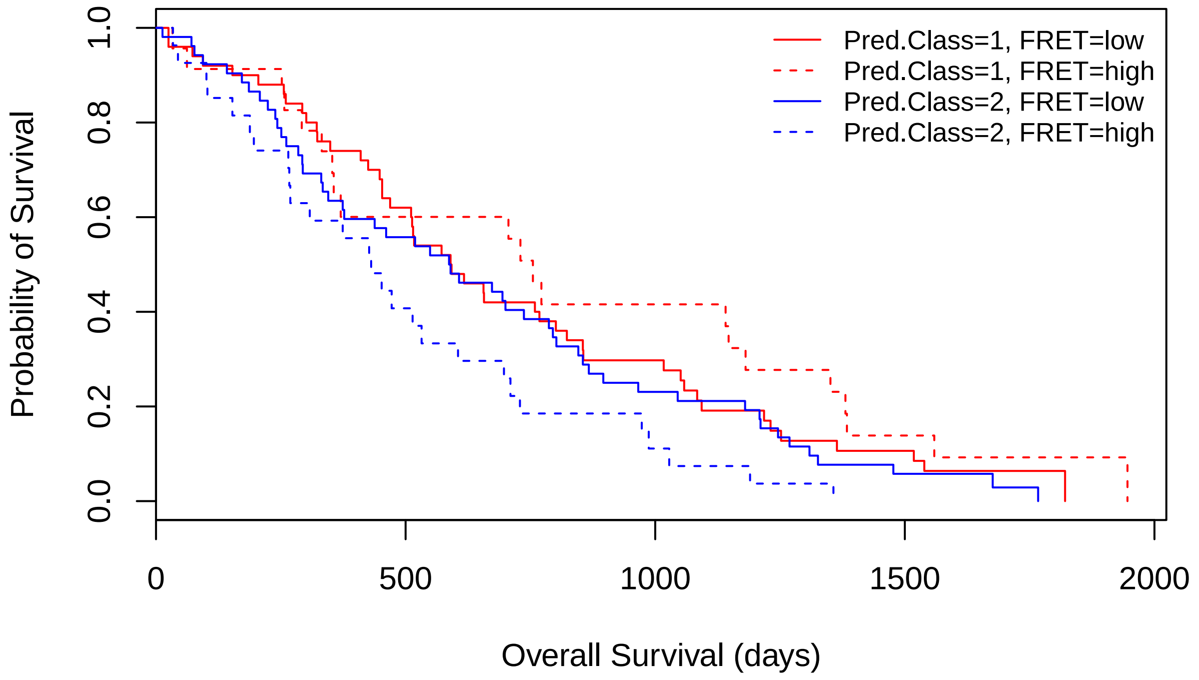
Number at risk

Pred.Class=1	73	44	23	8	0
Pred.Class=2	79	40	15	3	0

KM for raw FRET split by class

In Fig 4, FRET was split by tertiles of the original 398 patients. Use these same splits.

```r
col=c("red", "red", "blue", "blue")
lty=c(1,2,1,2)

#Split FRET by Tertiles, or comment ot use same breaks as the 398
breaks <- quantile(data$FRET, probs = c(0.0, 0.67, 1.0), na.rm = T)
data$FRETeff <- cut(data$FRET, breaks = breaks, labels = c("low", "high"),
                    include.lowest = T)
data$FRET_raw <- data$FRET
data$FRET <- data$FRETeff

km <- survfit(SurvObj.os ~ Pred.Class + FRET, data=data)
plot(km, col=col, lty=lty, xlab="Overall Survival (days)", ylab="Probability of Survival")
legend("topright", col=col, legend = names(km$strata), lty=lty, bty="n", cex=0.8)
```



```
print(km)
```

```
## Call: survfit(formula = SurvObj.os ~ Pred.Class + FRET, data = data)
##
##                             n events median 0.95LCL 0.95UCL
## Pred.Class=1, FRET=low  50     47    591     469     855
## Pred.Class=1, FRET=high 23     21    755     356    1351
## Pred.Class=2, FRET=low  52     51    588     374     802
## Pred.Class=2, FRET=high 27     27    431     269     697
```

```
survdiff(SurvObj.os ~ Pred.Class, data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os ~ Pred.Class, data = data)
##
##                N Observed Expected (O-E)^2/E (O-E)^2/V
## Pred.Class=1 73       68     80.4      1.90      4.35
## Pred.Class=2 79       78     65.6      2.33      4.35
##
##  Chisq= 4.4  on 1 degrees of freedom, p= 0.04
```

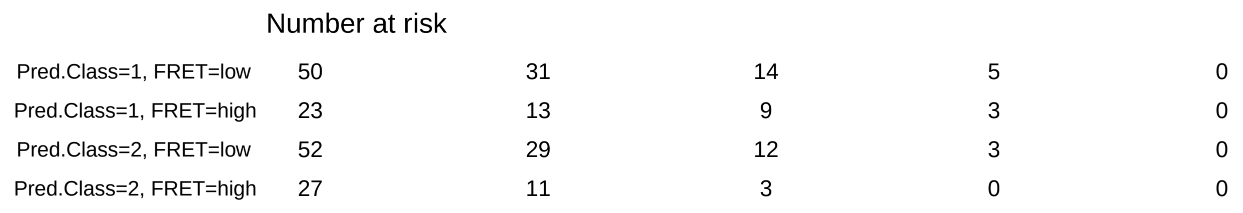
```
survdiff(SurvObj.os[Pred.Class==1] ~ FRET[Pred.Class==1], data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 1] ~ FRET[Pred.Class ==
##     1], data = data)
##
##                               N Observed Expected (O-E)^2/E (O-E)^2/V
## FRET[Pred.Class == 1]=low  50       47     43.1     0.362      1.04
## FRET[Pred.Class == 1]=high 23       21     24.9     0.624      1.04
##
##  Chisq= 1  on 1 degrees of freedom, p= 0.3
```

```
survdiff(SurvObj.os[Pred.Class==2] ~ FRET[Pred.Class==2], data=data)
```

```
## Call:
## survdiff(formula = SurvObj.os[Pred.Class == 2] ~ FRET[Pred.Class ==
##     2], data = data)
##
##                             N Observed Expected (O-E)^2/E (O-E)^2/V
## FRET[Pred.Class == 2]=low  52       51     57.7     0.782      3.12
## FRET[Pred.Class == 2]=high 27       27     20.3     2.226      3.12
##
##  Chisq= 3.1  on 1 degrees of freedom, p= 0.08
```

```
ggrisktable(km, data=data) + theme_cleantable()
```



Number at risk

Pred.Class=1, FRET=low	50	31	14	5	0
Pred.Class=1, FRET=high	23	13	9	3	0
Pred.Class=2, FRET=low	52	29	12	3	0
Pred.Class=2, FRET=high	27	11	3	0	0

Session Information

```
sessionInfo()
```

```
## R version 3.5.1 (2018-07-02)
## Platform: x86_64-w64-mingw32/x64 (64-bit)
## Running under: Windows 10 x64 (build 16299)
##
## Matrix products: default
##
## locale:
## [1] LC_COLLATE=English_United Kingdom.1252
## [2] LC_CTYPE=English_United Kingdom.1252
## [3] LC_MONETARY=English_United Kingdom.1252
## [4] LC_NUMERIC=C
## [5] LC_TIME=English_United Kingdom.1252
##
## attached base packages:
## [1] stats     graphics  grDevices utils     datasets  methods   base
##
## other attached packages:
##  [1] survival_2.42-3   pROC_1.15.3       kableExtra_1.1.0
##  [4] knitr_1.20        survminer_0.4.6   ggpubr_0.2
##  [7] magrittr_1.5      ggplot2_3.1.1     filesstrings_3.0.0
## [10] stringr_1.3.1
##
## loaded via a namespace (and not attached):
##  [1] zoo_1.8-4         tidyselect_0.2.5  purrr_0.2.5
##  [4] splines_3.5.1      lattice_0.20-35    colorspace_1.3-2
##  [7] generics_0.0.2     vctrs_0.2.0        viridisLite_0.3.0
## [10] htmltools_0.3.6    yaml_2.2.0         survMisc_0.5.5
## [13] rlang_0.4.0        pillar_1.4.1       glue_1.3.0
## [16] withr_2.1.2        matrixStats_0.54.0 lifecycle_0.1.0
## [19] plyr_1.8.4         munsell_0.5.0      gtable_0.2.0
## [22] rvest_0.3.4        evaluate_0.12      labeling_0.3
## [25] broom_0.5.2        Rcpp_1.0.1         xtable_1.8-4
## [28] readr_1.3.1        checkmate_1.9.3    scales_1.0.0
## [31] backports_1.1.2    webshot_0.5.1      km.ci_0.5-2
## [34] gridExtra_2.3      hms_0.4.2          digest_0.6.18
## [37] stringi_1.2.4      dplyr_0.8.3        KMsurv_0.1-5
## [40] grid_3.5.1         rprojroot_1.3-2    tools_3.5.1
## [43] lazyeval_0.2.1     tibble_2.1.1       crayon_1.3.4
## [46] tidyr_1.0.0        pkgconfig_2.0.2    zeallot_0.1.0
## [49] Matrix_1.2-14      xml2_1.2.2         data.table_1.12.2
## [52] strex_1.0.1        httr_1.4.1         assertthat_0.2.0
## [55] rmarkdown_1.10     rstudioapi_0.10    R6_2.3.0
## [58] nlme_3.1-137       compiler_3.5.1
```