Contents

Yoshihara 2012

Yoshihara K, Tsunoda T, Shigemizu D, Fujiwara H et al. *High-risk ovarian cancer based on 126-gene expression signature is uniquely characterized by downregulation of antigen presentation pathway.* Clin Cancer Res 2012 Mar 1;18(5):1374-85.

Implemented by Jie Ding.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/Yoshi2012.sig.csv"
[2] "22241791-TableS1.RData"
```

Load required libraries:

```
> library(survHD)
> library(curatedOvarianData)
> library(affy)
> library(HGNChelper)
```

The coefficients of the 126-gene signature are provided by the authors in Supplemental Table S1:

```
> Yoshi2012.sig <- read.csv(input_file, as.is=TRUE)
> head(Yoshi2012.sig)
```

```
  Gene.Symbol Entrez.ID     beta
1       AIMP2      7965  0.00013
2     ALOX5AP       241 -0.00258
3      ANKZF1     55139  0.00014
4       ANXA1       301 -0.00894
5       APOL1      8542 -0.00287
6      ARMCX2      9823 -0.00455
```

```
> dim(Yoshi2012.sig)
```

```
[1] 126   3
```

Check for and correct invalid HGNC symbols:

```
> hgnc.corrections <- checkGeneSymbols(Yoshi2012.sig$Gene.Symbol)
> hgnc.corrections[!hgnc.corrections$Approved, ]
```

```
          x Approved Suggested.Symbol
13 C19orf62    FALSE           BABAM1
70    MYST3    FALSE            KAT6A
```

```
> Yoshi2012.sig$Corrected.Gene.Symbol <- hgnc.corrections$Suggested.Symbol
```

Create the official model:

```
> coefs <- Yoshi2012.sig$beta
> names(coefs) <- Yoshi2012.sig$Corrected.Gene.Symbol
> model.official <- new("ModelLinear", coefficients=coefs, modeltype="plusminus")
```

Test using curatedOvarianData package. First, load the data:

```
> data(GSE32062.GPL6480_eset, package="curatedOvarianData")
> Yoshi2012.sig.exp <- exprs(GSE32062.GPL6480_eset)
> Yoshi2012.sig.exp <- Yoshi2012.sig.exp[rownames(Yoshi2012.sig.exp) %in% names(coefs), ]
> ##compared to 126 genes in author dataset, we lose a few in mapping in curatedOvarianData:
> dim(Yoshi2012.sig.exp)
```

```
[1] 122 260
```

```
> ##these are the genes lost in the expression data:
> names(coefs)[!names(coefs) %in% rownames(Yoshi2012.sig.exp)]
```

```
[1] "COX6A1" "FKBP1B" "RBX1"   "RPS7"
```

Get survival data from clinical data

```
> Yoshi2012.survival <- Surv(time=GSE32062.GPL6480_eset$days_to_death/30,
+                            event=GSE32062.GPL6480_eset$vital_status=='deceased')
```

Z-transfrom expression data:

```
> Yoshi2012.sig.z <- (Yoshi2012.sig.exp - rowMeans(Yoshi2012.sig.exp)) /
+   apply(Yoshi2012.sig.exp,1,sd)
```

Calculate scores:

```
> Yoshi2012.score <- predict(model.official, newdata=t(Yoshi2012.sig.z), type="lp")@lp
```

Divide into two groups using cutoff value 0.1517 from the paper:

```
> Yoshi2012.group <- ifelse(Yoshi2012.score >= 0.1517, "high", "low")
```

Make Kaplan-Meier curves. compare to Figure 1A:

Finally, save the model:

```
> save(model.official, file=model_file)
> plotKM(y=Yoshi2012.survival, strata=factor(Yoshi2012.group), censor.at=140)
```

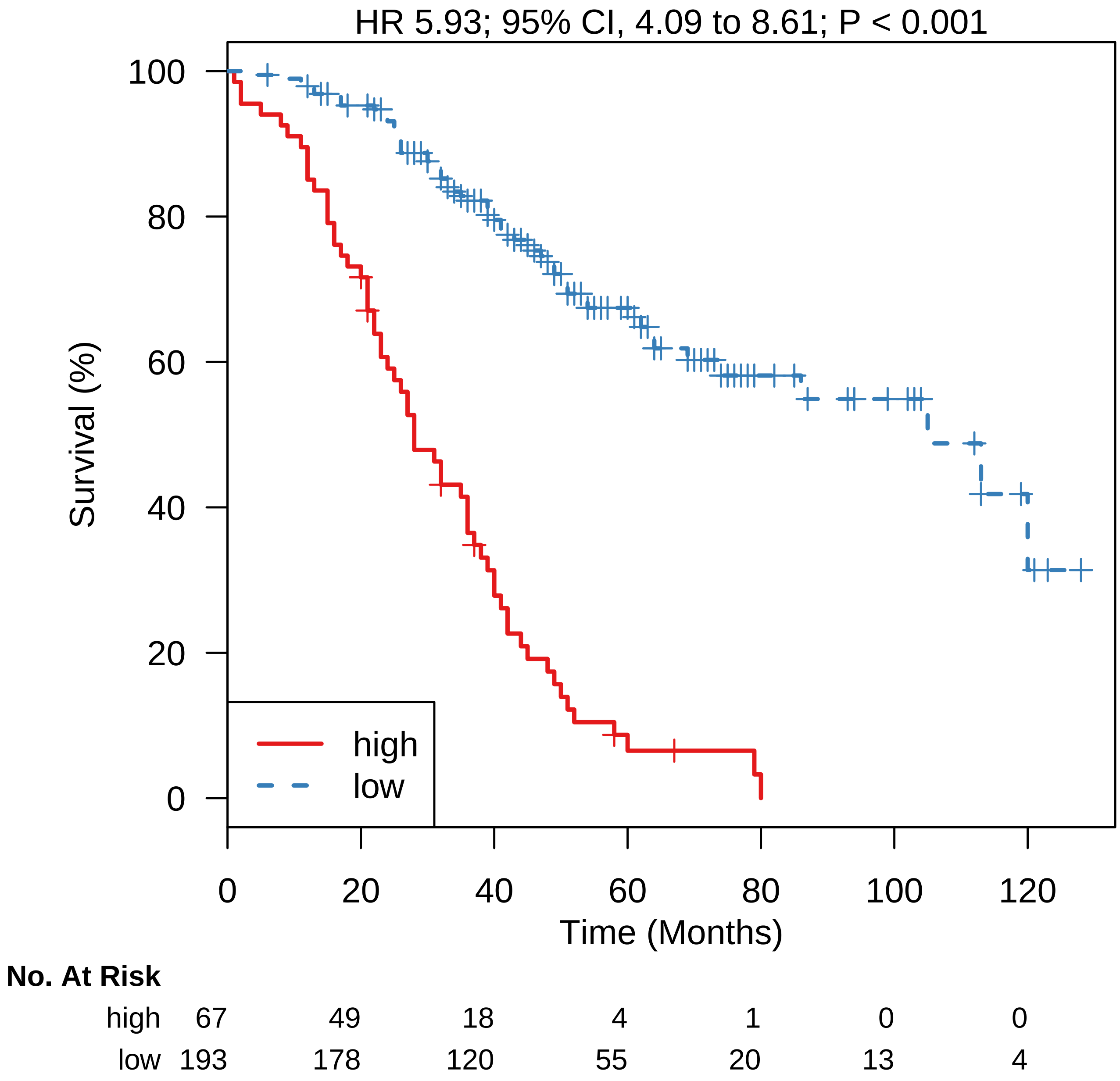


Figure 1: This looks identical to Figure 1A.

```
> data(GSE9891_eset, package="curatedOvarianData")
> tothill.dataset <- exprs(GSE9891_eset)
> tothill.dataset <- tothill.dataset[rownames(tothill.dataset) %in% names(coefs), ]
> tothill.dataset <- (tothill.dataset - rowMeans(tothill.dataset)) /
+   apply(tothill.dataset,1,sd)
> tothill.dataset.scores <- predict(model.official, newdata=t(tothill.dataset), type="lp")@lp
> tothill.dataset.group <- ifelse(tothill.dataset.scores >= 0.1517, "high", "low")
> tothill.dataset.survival <- Surv(time=GSE9891_eset$days_to_death/30,
+                                  event=GSE9891_eset$vital_status=='deceased')
> plotKM(y=tothill.dataset.survival, strata=factor(tothill.dataset.group), censor.at=80)
```

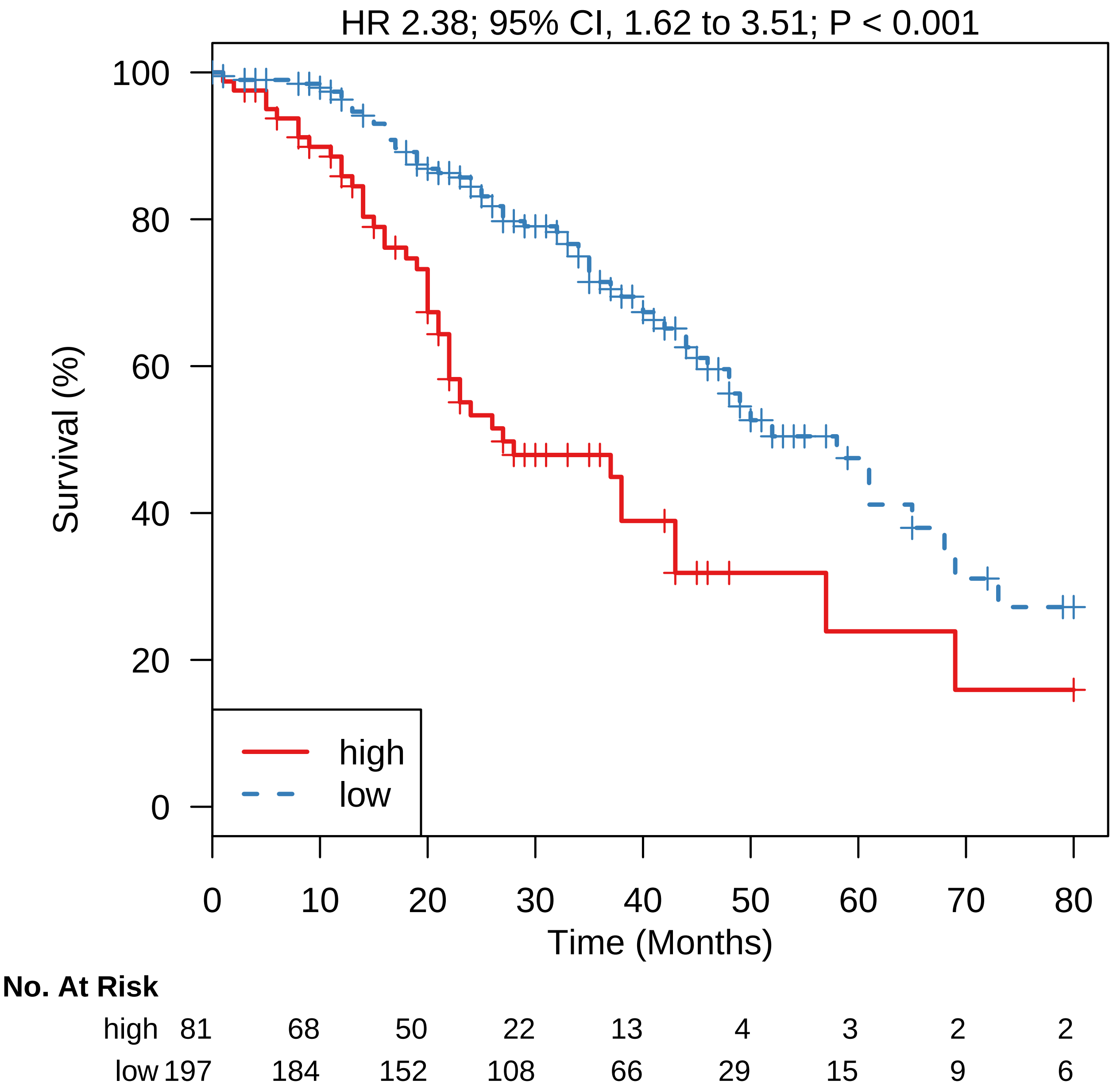


Figure 2: Compare to Figure 1B - very similar, only differs by several patients at times > 60 mo.

```
> data(GSE26712_eset, package="curatedOvarianData")
> bonome.dataset <- exprs(GSE26712_eset)
> bonome.dataset <- bonome.dataset[rownames(bonome.dataset) %in% names(coefs), ]
> bonome.dataset <- (bonome.dataset - rowMeans(bonome.dataset)) /
+   apply(bonome.dataset,1,sd)
> bonome.dataset.scores <- predict(model.official, newdata=t(bonome.dataset), type="lp")@lp
> bonome.dataset.group <- ifelse(bonome.dataset.scores >= 0.1517, "high", "low")
> bonome.dataset.survival <- Surv(time=GSE26712_eset$days_to_death/30,
+                                 event=GSE26712_eset$vital_status=='deceased')
> plotKM(y=bonome.dataset.survival, strata=factor(bonome.dataset.group), censor.at=180)
```

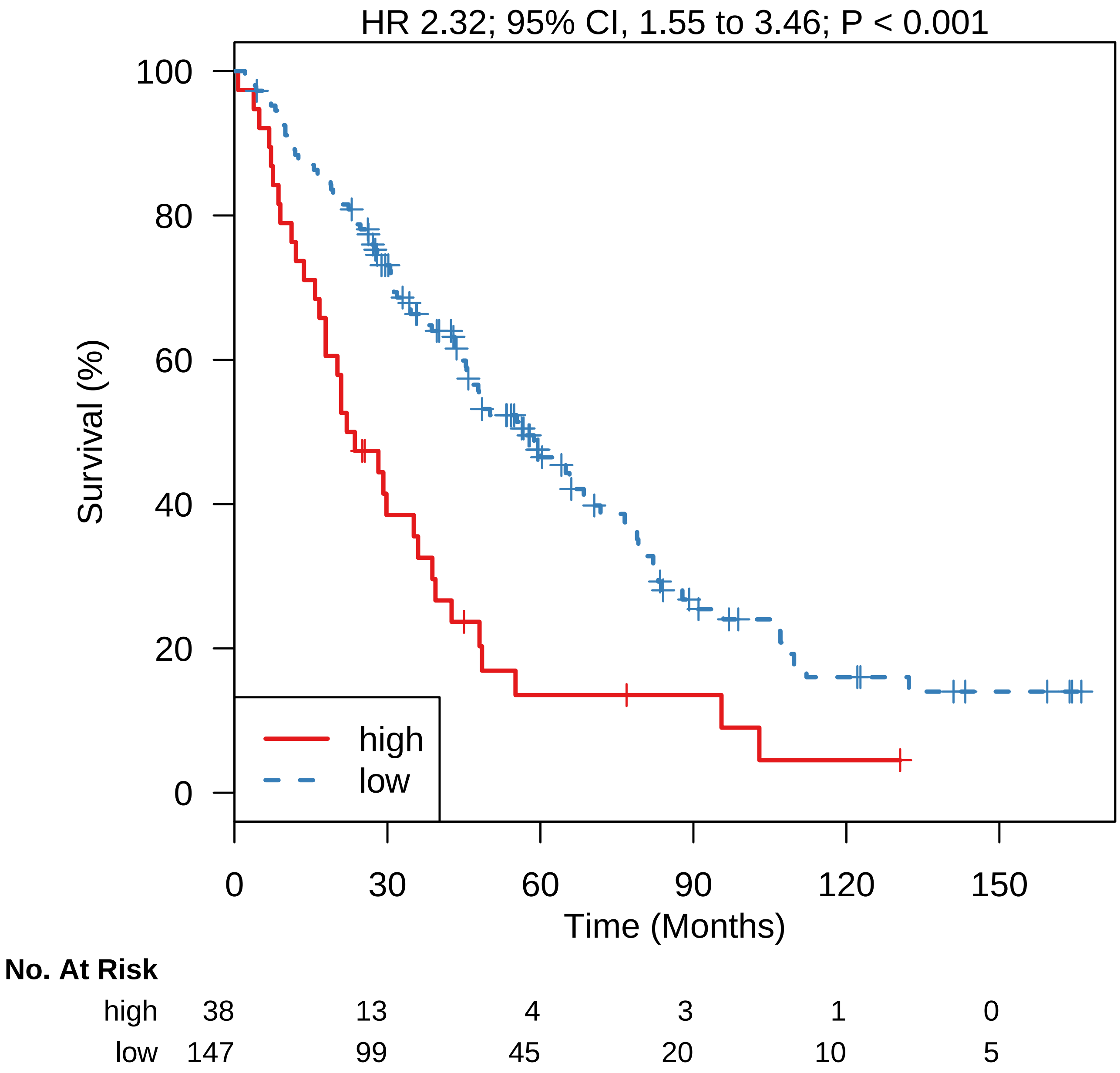
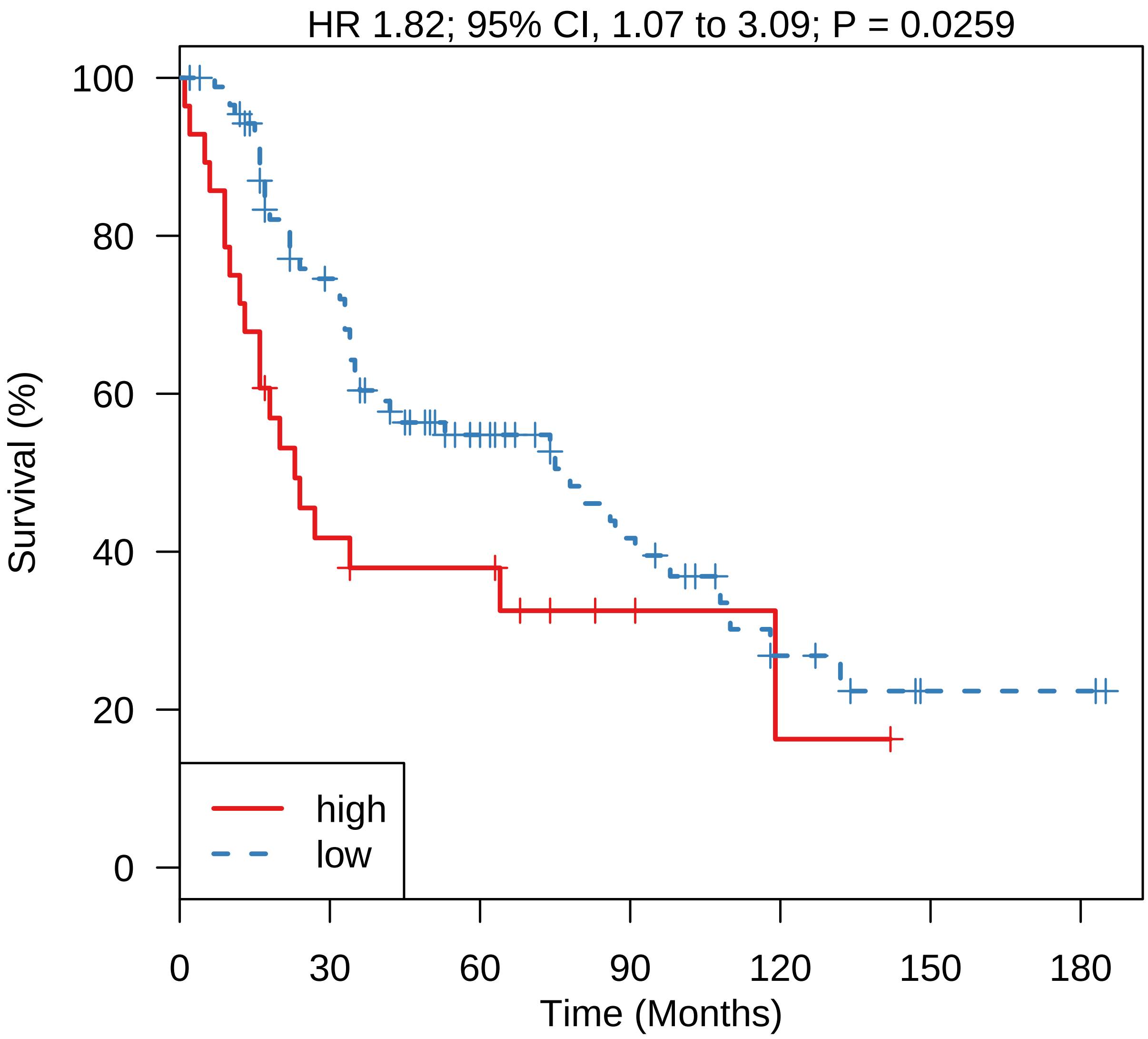


Figure 3: Compare to Figure 1C, Bonome dataset. Very nearly identical.

```
> data(PMID17290060_eset, package="curatedOvarianData")
> dressman.dataset <- exprs(PMID17290060_eset)
> dressman.dataset <- dressman.dataset[rownames(dressman.dataset) %in% names(coefs), ]
> dressman.dataset <- (dressman.dataset - rowMeans(dressman.dataset)) /
+   apply(dressman.dataset,1,sd)
> dressman.dataset.scores <- predict(model.official, newdata=t(dressman.dataset), type="lp")@lp
> dressman.dataset.group <- ifelse(dressman.dataset.scores >= 0.1517, "high", "low")
> dressman.dataset.survival <- Surv(time=PMID17290060_eset$days_to_death/30,
+                                   event=PMID17290060_eset$vital_status=='deceased')
> plotKM(y=dressman.dataset.survival, strata=factor(dressman.dataset.group), censor.at=200)
```



No. At Risk

high	28	11	8	3	1	0	0
low	89	58	32	19	7	2	2

Figure 4: Compare to Figure 1D, Dressman dataset. Looks identical except that high-risk line seems truncated in Figure 1D.

```
> data(TCGA_eset, package="curatedOvarianData")
> tcga.dataset <- exprs(TCGA_eset)
> tcga.dataset <- tcga.dataset[rownames(tcga.dataset) %in% names(coefs), ]
> tcga.dataset <- (tcga.dataset - rowMeans(tcga.dataset)) /
+   apply(tcga.dataset,1,sd)
> tcga.dataset.scores <- predict(model.official, newdata=t(tcga.dataset), type="lp")@lp
> tcga.dataset.group <- ifelse(tcga.dataset.scores >= 0.1517, "high", "low")
> tcga.dataset.survival <- Surv(time=TCGA_eset$days_to_death/30,
+                               event=TCGA_eset$vital_status=='deceased')
> plotKM(y=tcga.dataset.survival, strata=factor(tcga.dataset.group), censor.at=160)
```

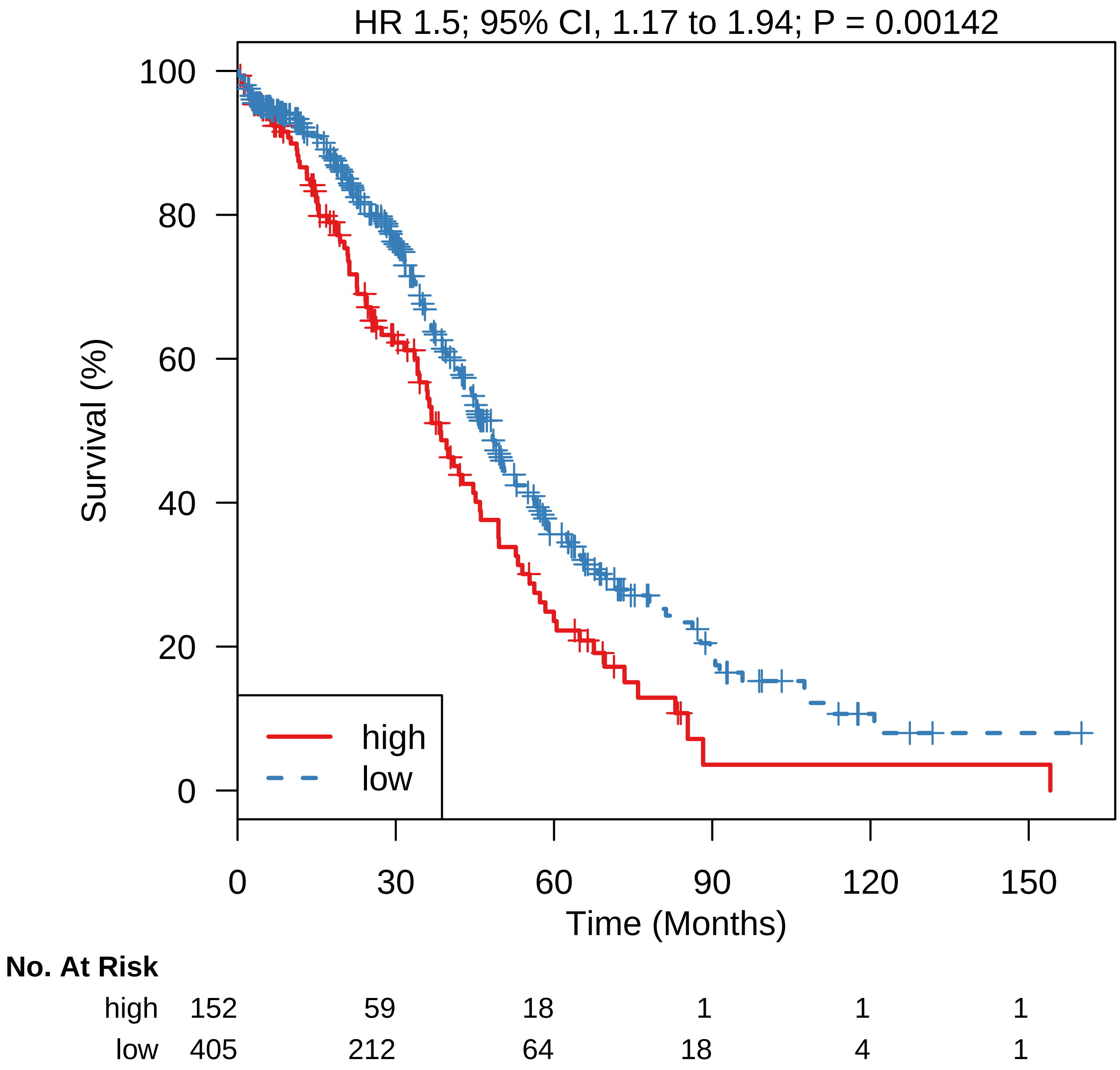


Figure 5: Compare to Figure 1E, TCGA dataset. Very nearly identical, again differing by probably one patient in the high-risk group with time > 80mo.

```
> data(GSE32063_eset, package="curatedOvarianData")
> japanese.datasetB <- exprs(GSE32063_eset)
> japanese.datasetB <- japanese.datasetB[rownames(japanese.datasetB) %in% names(coefs), ]
> japanese.datasetB <- (japanese.datasetB - rowMeans(japanese.datasetB)) /
+   apply(japanese.datasetB,1,sd)
> japanese.datasetB.scores <- predict(model.official, newdata=t(japanese.datasetB), type="lp")@lp
> japanese.datasetB.group <- ifelse(japanese.datasetB.scores >= 0.1517, "high", "low")
> japanese.datasetB.survival <- Surv(time=GSE32063_eset$days_to_death / 30,
+                                    event=GSE32063_eset$vital_status=='deceased')
> plotKM(y=japanese.datasetB.survival, strata=factor(japanese.datasetB.group), censor.at=120)
```

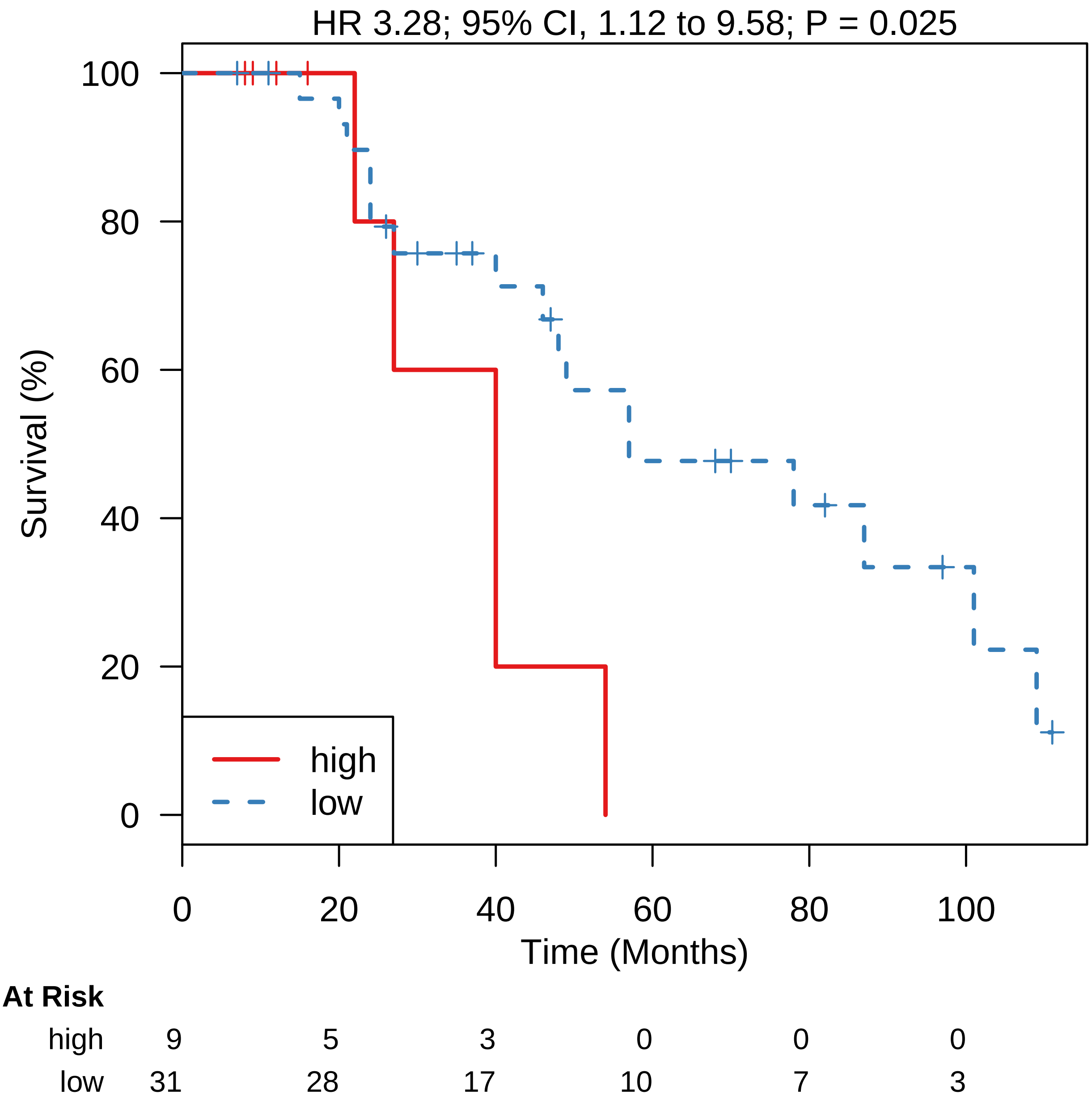


Figure 6: Compare to Figure 1F, Japanese dataset B. Very similar.

Bentink / Haibe-Kains 2012, Angiogenic score

Bentink S, Haibe-Kains B, Risch T, Fan JB, Hirsch MS, Holton K, Rubio R, April C, Chen J, Wickham-Garcia E, Liu J, Culhane A, Drapkin R, Quackenbush J, Matulonis UA. *Angiogenic mRNA and microRNA gene expression signature predicts a novel subtype of serous ovarian cancer.* PLoS One. 2012;7(2):e30269. Epub 2012 Feb 13. PMID: 22348002.

This model was implemented by Benjamin Haibe-Kains, also an author of the PLoS ONE paper, in the genefu package. Therefore here we simply implement a wrapper around that function and test it on the data from the paper.

No input files, just the output for saving the model.

```
> print(model_file)
```

```
[1] "22348002-FileS1.RData"
```

Load required libraries:

```
> library(survHD)
> library(survival)
```

This score is implemented by author Benjamin Haibe-Kains, in the *genefu* package, so we simply create a survHD wrapper around that prediction function.

Currently the development version of *genefu* if needed to test the survHD version of this model:

```
> if(!require(genefu)){
+     library(devtools)
+     install_github("genefu", username="bhaibeka", branch="master")
+ }
> library(genefu)
```

Write a simple wrapper function to the ovcAngiogenic function of the genefu package:

```
> genefuFun <- function(object=NULL,newdata, type="lp", ...){
+     newdata<-as.matrix(newdata)
+     if(type == "lp"){
+         typeinternal <- "score"
+         library(genefu)
+         annot.eset <- data.frame(hgnc_symbol = colnames(newdata))
+         rownames(annot.eset) <- colnames(newdata)
+         Official.output <- ovcAngiogenic(data=newdata,
+                                          annot=annot.eset,
+                                          gmap="hgnc_symbol",
+                                          do.mapping=TRUE)
+         ##take the negative of the score for increasing values to predict risk:
+         pred <- new("LinearPrediction",lp = -Official.output[["score"]])
+     }
+     return(pred)
+ }
```

And use this to define the official model as a ModelCustom object:

```
> model.official <- new("ModelCustom", predfun=genefuFun)
```

Double-check that the predictions are the same, data from this paper.

```
> data(E.MTAB.386_eset, package="curatedOvarianData")
> genefu.preds <- genefuFun(newdata=t(exprs(E.MTAB.386_eset)))@lp
> survhd.preds <- predict(model.official, newdata=t(exprs(E.MTAB.386_eset)), type="lp")@lp
> stopifnot( all.equal(genefu.preds, survhd.preds) )
```

Confirm that this positively predicts risk in the training data:

Finally, save the model:

```
> save(model.official, file=model_file)
> surv.data <- Surv(E.MTAB.386_eset$days_to_death / 30, E.MTAB.386_eset$vital_status == "deceased")
> plotKMStratifyBy(y=surv.data,
+                  n.risk.step=12,
+                  linearriskscore=survhd.preds,
+                  labels=rev(c("Non-angiogenic", "Angiogenic")),
+                  cutpoints = survhd.preds[ rank(survhd.preds) == 99 ])
```

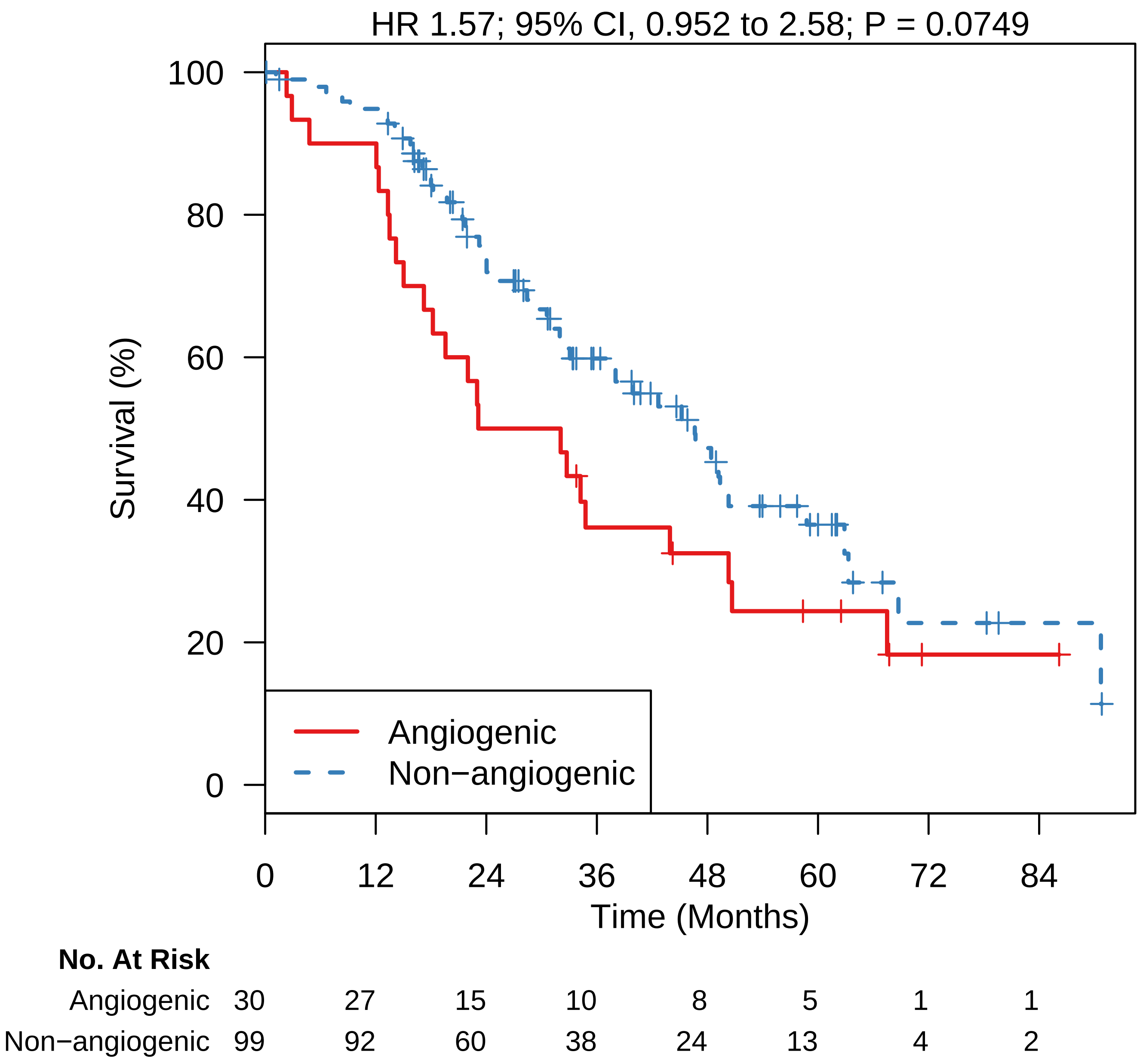


Figure 7: Compare to Figure 2G of the paper. In spite of differences in data pre-processing and probesets between the authors' methods and curatedOvarianData, we reproduce the results closely, with slightly better prediction seen here (the paper reported HR=1.3). Note that high scores correspond to high risk and the angiogenic subtype.

Kernagis 2012

Dawn N. Kernagis, Allison H.S. Hall, and Michael B. Datto: *Genes with Bimodal Expression Are Robust Diagnostic Targets that Define Distinct Subtypes of Epithelial Ovarian Cancer with Different Overall Survival.* JMD 2012, 14(3).

Implemented by Markus Riester and Levi Waldron.

Input arguments:

```
> print(c(model_file, uncurated_file, bimod_file, sig_file))
```

```
[1] "22490445-Table1.RData"
[2] "../../input/official_models/GSE9891_SuppTable.csv"
[3] "../../input/official_models/kernagis_2012_TableS6.csv"
[4] "../../input/official_models/kernagis_2012_Table1.csv"
```

Load required libraries:

```
> library(FULLVcuratedOvarianData)
> library(devtools)
> if (!require(ClassDiscovery)) {
+     source("http://bioinformatics.mdanderson.org/OOMPA/oompaLite.R")
+     oompaLite()
+     library(ClassDiscovery)
+ }
> library(survHD)
> library(curatedOvarianData)
> library(affy)
> library(devtools)
> library(GEOquery)
> library(annotate)
> library(hgu133plus2.db)
```

We will consider three versions of the dataset. First, curatedOvarianData with HGNC symbols as features. We substitute "-" with "hyphen" in HGNC symbols so they are valid R names:

```
> data(GSE9891_eset, package="curatedOvarianData")
> eset.genes <- GSE9891_eset
```

FULLVcuratedOvarianData version, which uses original probe set identifiers, and is pre-processed using frozen RMA:

```
> data(GSE9891_eset, package="FULLVcuratedOvarianData")
> eset.probes <- GSE9891_eset
```

And using the version from GEO:

```
> set.seed(1)
> fn <- "gse9891geo.rda"
> if (file.exists(fn)) {
+     load(fn)
+ } else {
+     gse <- getGEO("GSE9891")
+     eset.geo = BiocGenerics::combine(gse[[1]], gse[[2]])
+     save(eset.geo,file=fn)
+ }
```

Load the Supplementary Table to get the low malignant potential (LMP) samples, which the authors excluded from all analyses.

```
> lmp <- read.csv(uncurated_file)
> eset.genes <- eset.genes[ ,eset.genes$histological_type=="ser"]
> eset.genes <- eset.genes[,-match(make.names(lmp[lmp$Type=="LMP",1]), eset.genes$alt)]
```

Prepare the Surv objects (overall survival) for each ExpressionSet:

```
> eset.genes$y <- Surv(eset.genes$days_to_death / 30, eset.genes$vital_status == "deceased")
> eset.genes  <- eset.genes[, !is.na(eset.genes$y)]
> eset.probes <- eset.probes[ ,sampleNames(eset.genes)]
> eset.geo <- eset.geo[ ,sampleNames(eset.genes)]
> eset.probes$y <- eset.genes$y
> eset.geo$y    <- eset.genes$y
```

Read the bimodality indices from Supplementary Table 6 and the signature from Table 1:

```
> bimod.table <- read.csv(bimod_file, as.is=TRUE)
> geneset <- read.csv(sig_file,as.is=TRUE)
> geneset.probes <- geneset[,1]
> geneset.genes <- geneset[,2]
```

All of the probesets in this table should be found in the full versions of the ExpressionSets:

```
> summary(geneset.probes %in% featureNames(eset.probes))

   Mode    TRUE    NA's 
logical      16       0 

> summary(geneset.probes %in% featureNames(eset.geo))

   Mode    TRUE    NA's 
logical      16       0 
```

curatedOvarianData uses mappings from biomaRt, and some additional probesets are lost in the curatedOvarianData ExpressionSet. Those shown below as NA or blank are missing in both biomaRt and Bioconductor:

```
> summary(geneset.genes %in% featureNames(eset.genes))

   Mode   FALSE    TRUE    NA's 
logical       2      14       0 
```

```
> geneset.genes[!geneset.genes %in% featureNames(eset.genes)]

[1] ""          "LOC283392"

> ##keep the ones present in the data:
> geneset.genes <- geneset.genes[ geneset.genes %in% featureNames(eset.genes) ]
> geneset.genes <- geneset.genes[ !duplicated(geneset.genes) ]

> rowcox.probes <- rowCoxTests(X=exprs(eset.probes[geneset.probes,]), y=eset.probes$y)
> rowcox.geo <- rowCoxTests(X=exprs(eset.geo[geneset.probes,]), y=eset.geo$y)
```

Now we calculate the bimodality index for their top biomodal genes:

```
> bmi.probes <- bimodalIndex(exprs(eset.probes[bimod.table[,1],]))

1 .

> bmi.geo    <- bimodalIndex(exprs(eset.geo[bimod.table[,1],]))

1 .
```

Now get the high/low expression cutoffs for each gene from the mixture modes as described in the Methods.

```
> cutoffs.geo <- cbind(bmi.geo$mu1+2*bmi.geo$sigma,  bmi.geo$mu2-2*bmi.geo$sigma)
> # if they overlap, use average
> idx <- apply(cutoffs.geo,1,which.max)==1
> cutoffs.geo[idx,] = rep(apply(cutoffs.geo[idx,],1, mean),2)
> rownames(cutoffs.geo) <- rownames(bmi.geo)
```

Kernagis et al. now use a compound covariate score, but first trichotomized the expression based on the high/low cutoffs.

```
> X <- t(sapply(geneset.probes, function(x) ifelse(exprs(eset.geo)[x,]<
+     cutoffs.geo[x,1],-1, ifelse(exprs(eset.geo)[x,]> cutoffs.geo[x,2],1,0  ))))
```

The trichotomization is problematic for cross-platform validation, so we compare to a standard continous predictor:

Now we build a compound covariate prediction model:

```
> model.geo <- plusMinusSurv(eset.geo[geneset.probes,],
+                             y=eset.geo$y,
+                             modeltype="compoundcovariate")
> model.probes <- plusMinusSurv(eset.probes[geneset.probes,],
+                                y=eset.probes$y,
+                                modeltype="compoundcovariate")
> model.genes <- plusMinusSurv(eset.genes[geneset.genes,],
+                               y=eset.genes$y,
+                               modeltype="compoundcovariate")
```

Having established that a version of this model based on continuous expression values, which will be extensible across platforms, seems to be as predictive as the trichotomized version proposed (See Figures 10, 11, 12, 13), we save the gene-based model. Note that this model should be applied to *unscaled* data.

```
> plot(bmi.probes$BI, bimod.table$BI,ylab="Reported", xlab="Reproduced from COD")
```

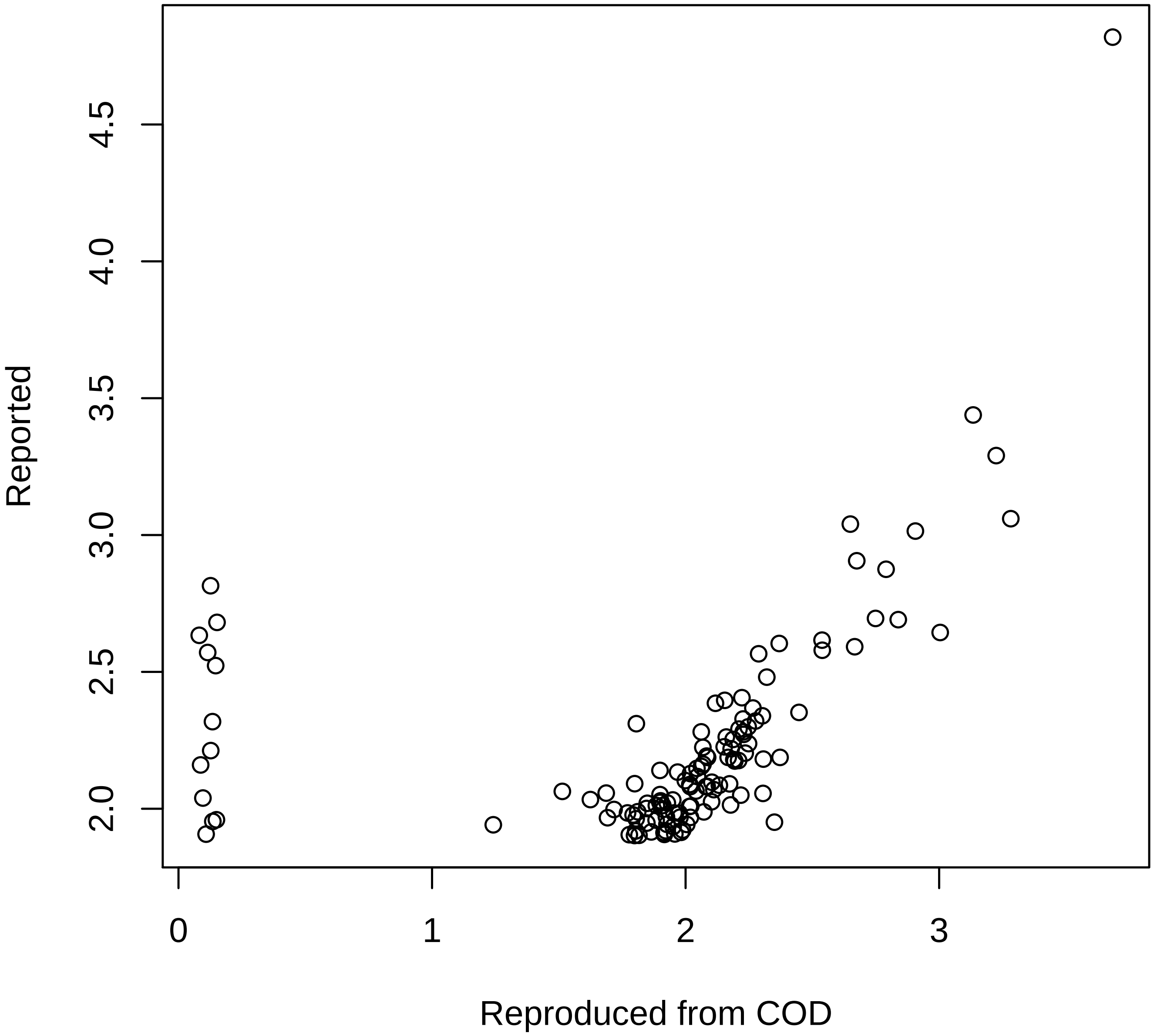


Figure 8: Comparison of reported BI and BI reproduced from curatedOvarianData package. Generally high correlation except for some genes which have a BI close to 0 in the reproduced version.

```
> plot(bmi.geo$BI, bimod.table$BI,ylab="Reported", xlab="Reproduced from GEO")
```

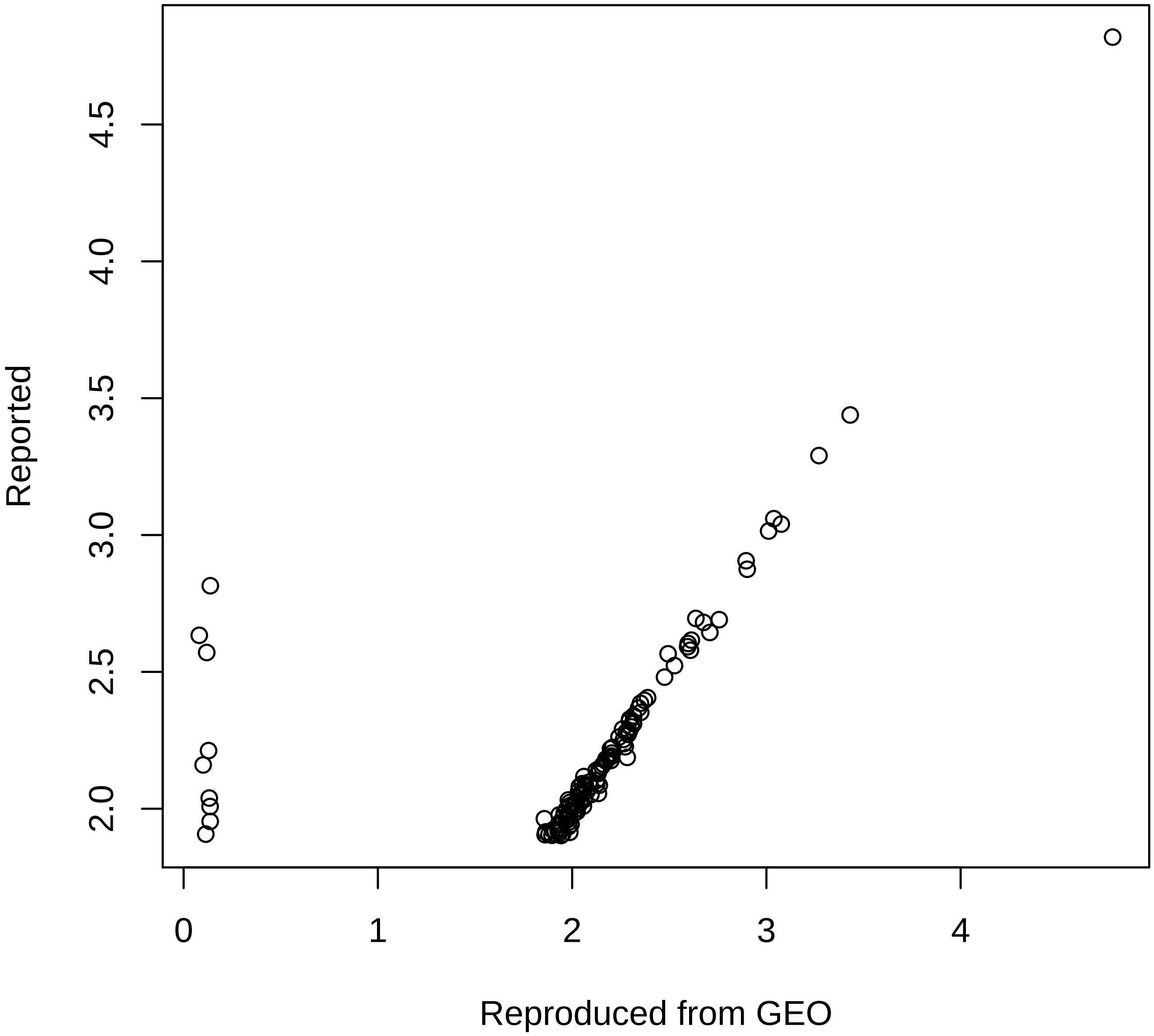


Figure 9: Comparison of reproduced and reported BI using the authors' expression data from GEO. Here we have exact reproduction of BI, except for those cases with BI near zero in reproduced version.

```
> rct.continuous <- rowCoxTests(exprs(eset.geo)[geneset.probes,],eset.geo$y)
> rct.trichotomized <- rowCoxTests(X,eset.geo$y)
> plot(rct.continuous[,3],
+ rct.trichotomized[,3],xlab="Continous",ylab="Trichotomized")
```

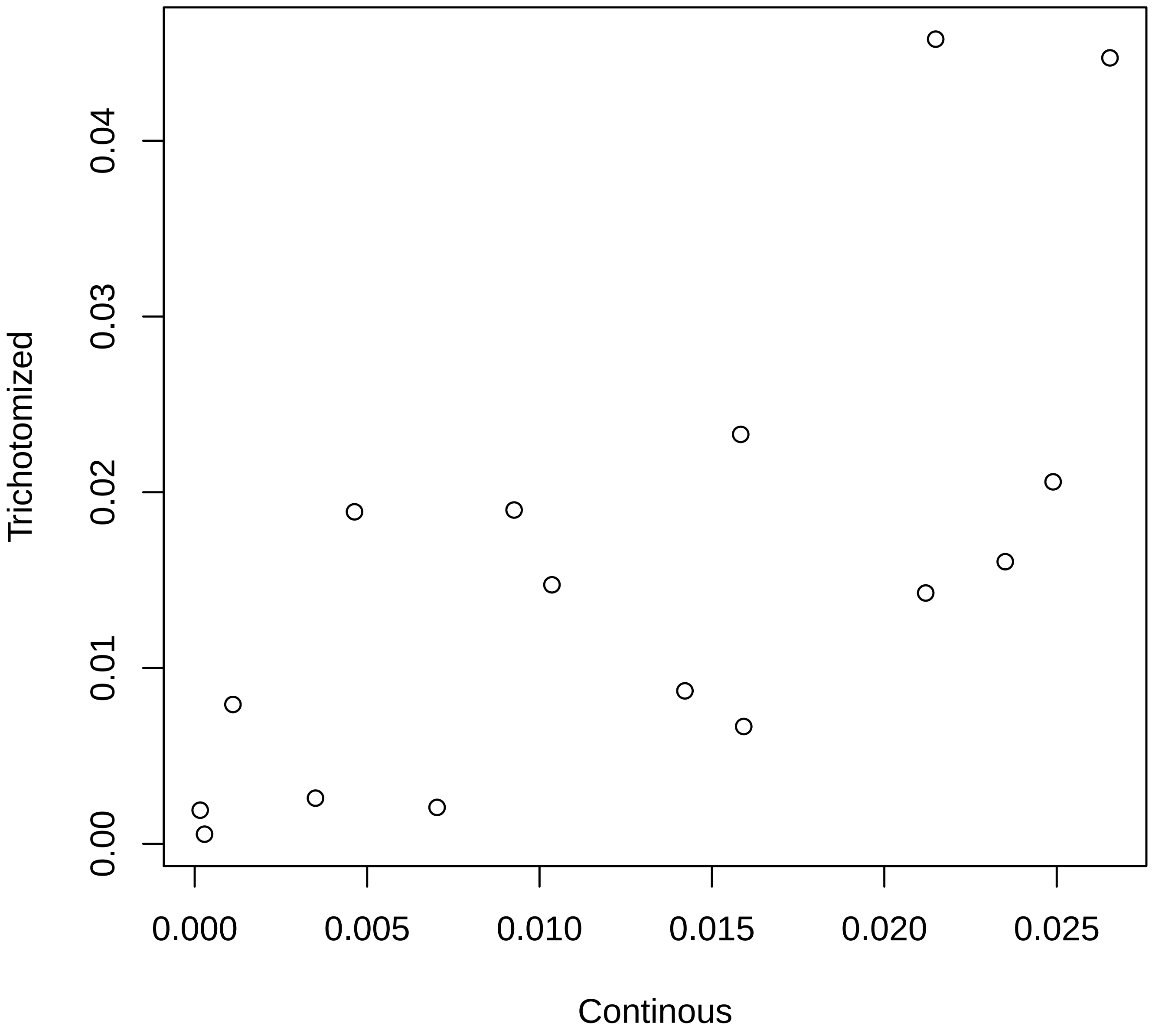


Figure 10: Comparison of per-gene Cox tests using continuous and trichotomized expression. High correlation, and overall smaller p-values with continuous expression, implies that trichotomization does not enhance the association of these genes with survival. We conclude that the continous predictors have similar prediction accuracy, which justifies our choice to not trichotomize in the interest of cross-platform validation.

```
> data(GSE18520_eset, package="FULLVcuratedOvarianData")
> eset.GSE18520.probes <- GSE18520_eset
> eset.GSE18520.probes$y <- Surv(GSE18520_eset$days_to_death/30,
+     GSE18520_eset$vital_status=="deceased")
> eset.GSE18520.probes <- eset.GSE18520.probes[, !is.na(eset.GSE18520.probes$y)]
> tmp <- plot(model.geo, newdata=eset.GSE18520.probes,
+     newy=eset.GSE18520.probes$y,cutpoints= median(predict(model.geo)@lp))
```

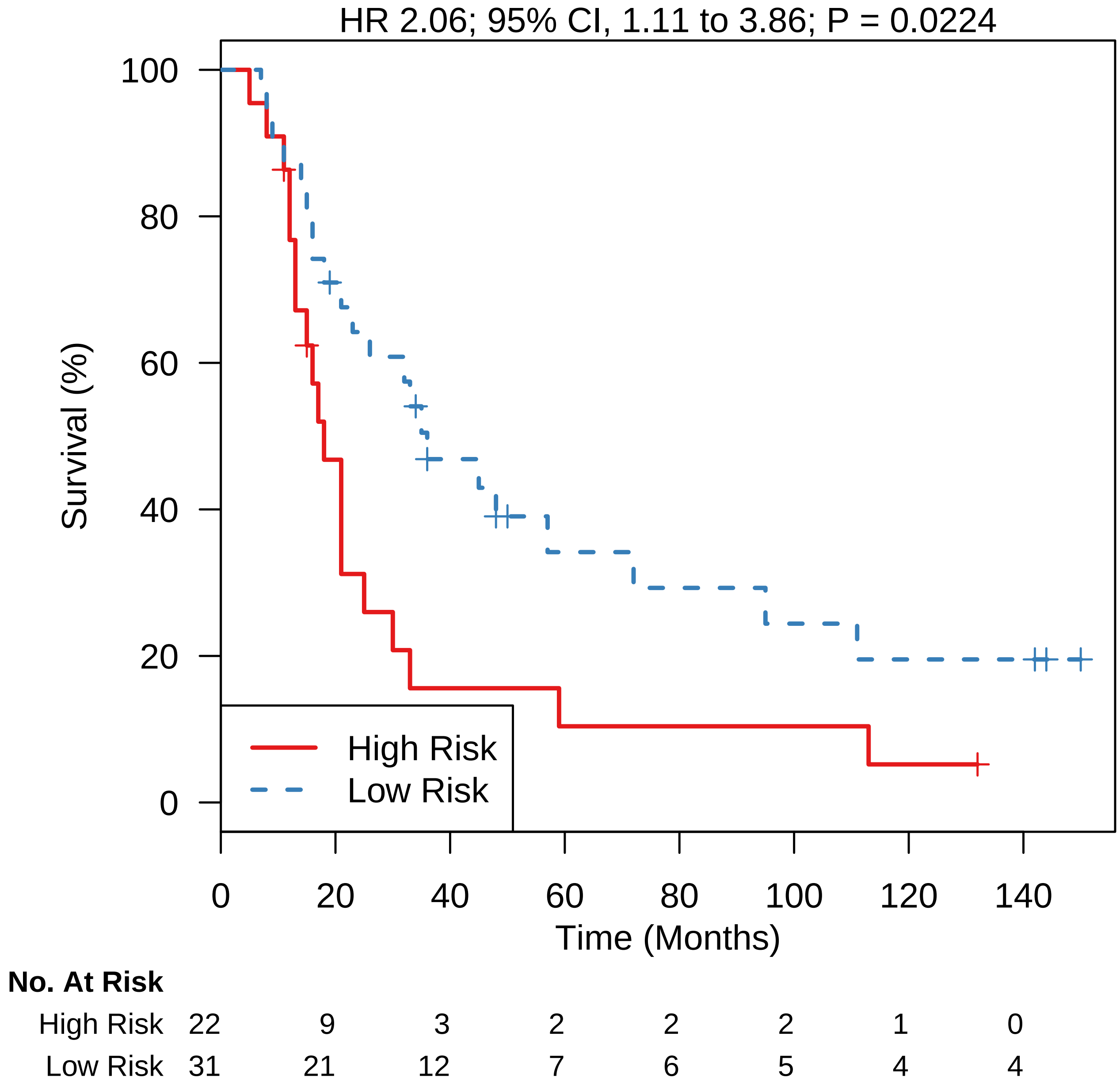


Figure 11: Reproducing Figure 4C as close as possible using the GEO data. Since the thresholds for high, low, and indeterminate risk used by the authors are not applicable to our use of continuous gene expression, we simply dichotomize at the median, and see a hazard ratio comparable to that reported for low vs. indeterminate/high scores (1/0.4565 = 2.19)

```
> plot(model.probes, newdata=eset.GSE18520.probes,
+     newy=eset.GSE18520.probes$y,cutpoints= median(predict(model.probes)@lp))
```

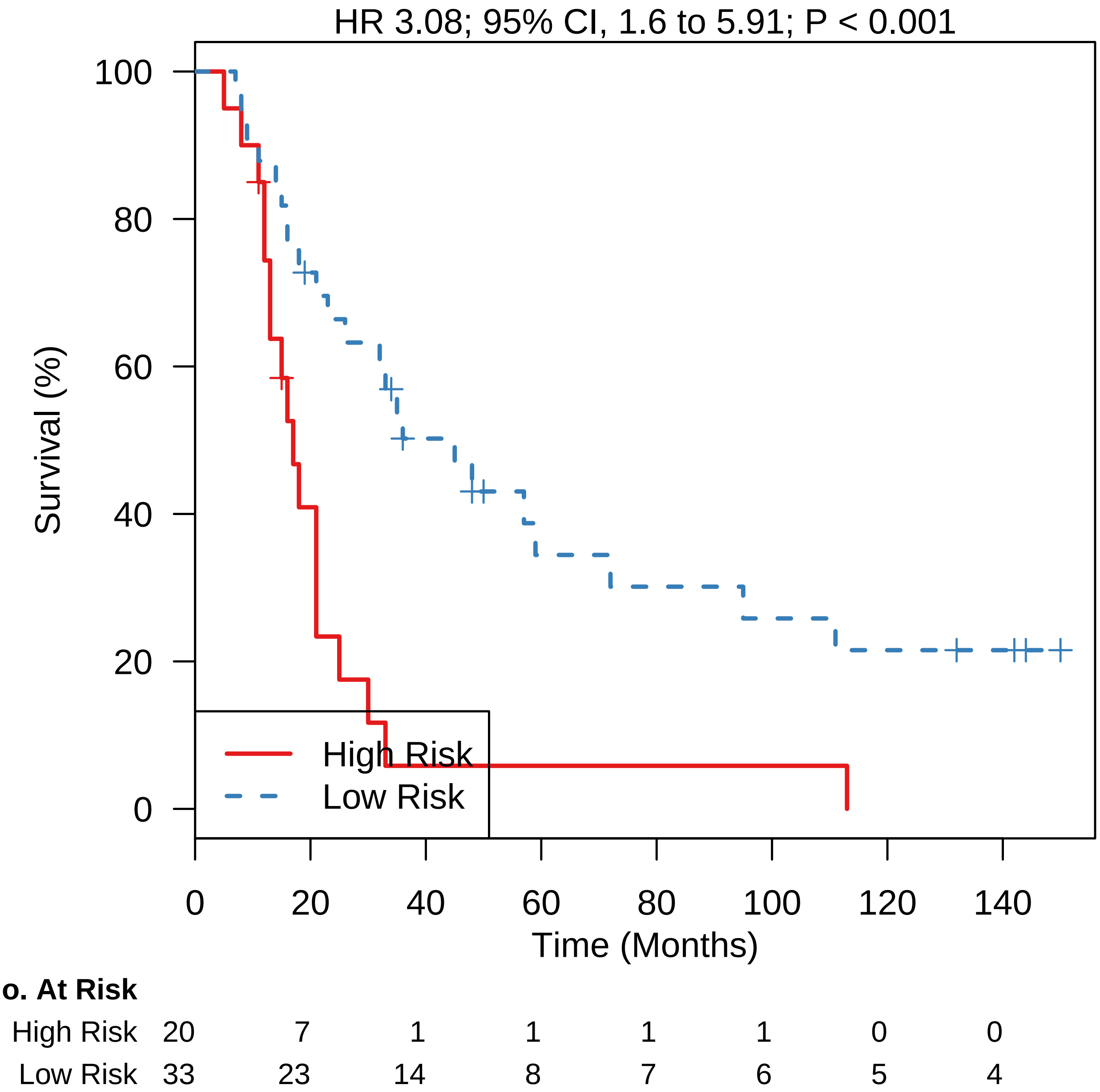


Figure 12: Reproducing Figure 4C using the uncollapsed COD data. HR even higher than when using the GEO data. These curves appear nearly identical to the low and IND/high curves shown in Figure 4C.

```
> data(GSE18520_eset, package="curatedOvarianData")
> eset.GSE18520.genes <- GSE18520_eset
> eset.GSE18520.genes$y <- Surv(GSE18520_eset$days_to_death/30,
+     GSE18520_eset$vital_status=="deceased")
> eset.GSE18520.genes <- eset.GSE18520.genes[, !is.na(eset.GSE18520.genes$y)]
> plot(model.genes, newdata=eset.GSE18520.genes,
+     newy=eset.GSE18520.genes$y,cutpoints= median(predict(model.genes)@lp))
```

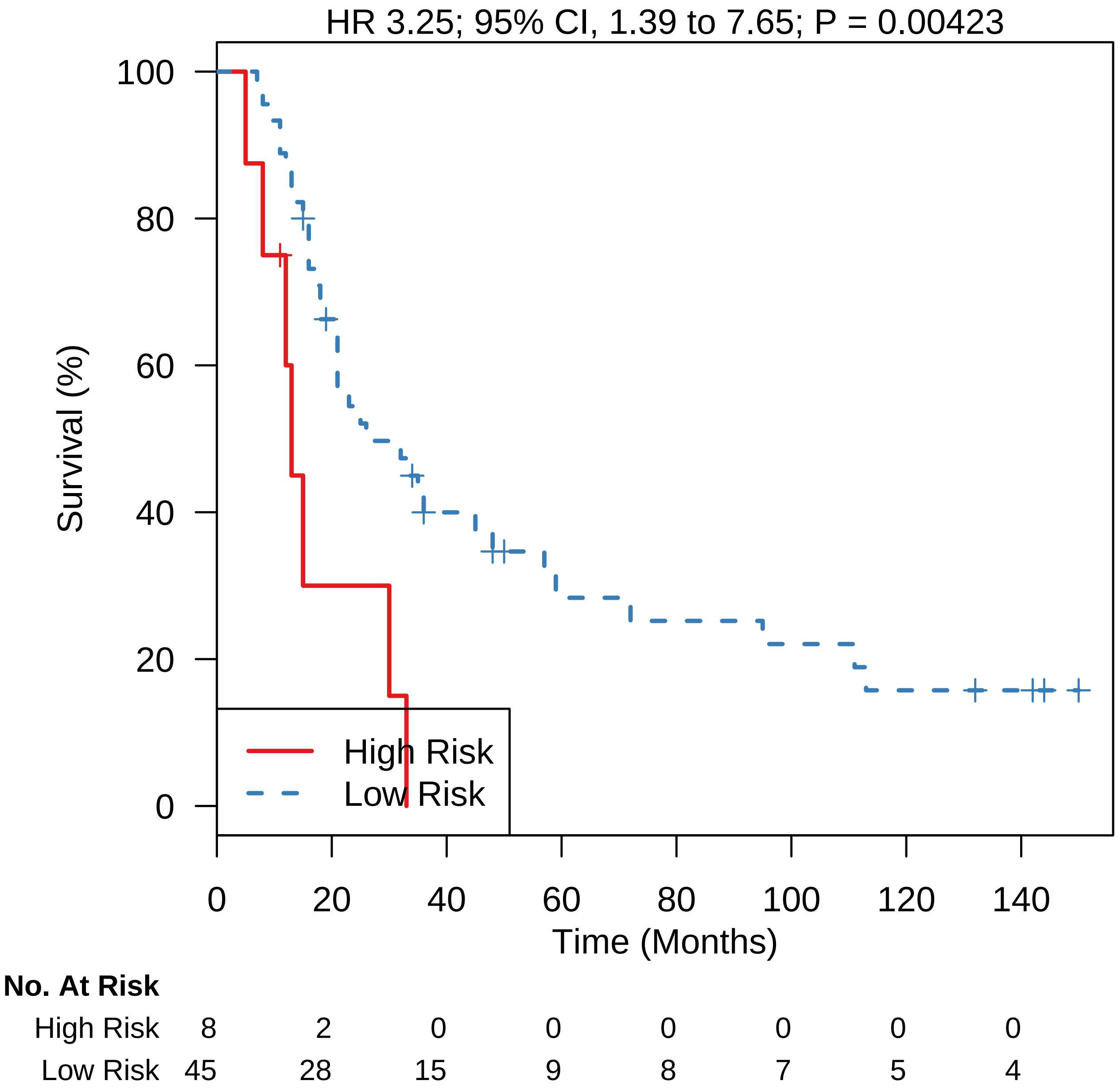


Figure 13: Reproducing Figure 4C using the collapsed COD data, which appears equivalent to the probeset version of the model in the previous figure.

```
> model.cod <- model.genes@model
> save(model.cod, file=model_file)
```

Kang 2012

The 23-gene DNA-damage repair score proposed by Kang et al:

> Kang, J., A. D'Andrea, et al. (2012). ”A DNA Repair Pathway-Focused Score for Prediction of Outcomes in Ovarian Cancer Treated With Platinum-Based Chemotherapy.” Journal of the National Cancer Institute 104(9):670-81.

and commented on:

> Swisher, E., T. Taniguchi, et al. (2012). ”Molecular scores to predict ovarian cancer outcomes: a worthy goal, but not ready for prime time.” Journal of the National Cancer Institute 104(9): 642-5.

Implemented here by Levi Waldron and Victoria Wang.

Input arguments:

```
> print(c(input_file, model_file, input_suppfile, input_tcgadata, input_table3data))

[1] "../../input/official_models/Kang12_Table2.txt"
[2] "22505474-Table2.RData"
[3] "../../input/official_models/Kang12_SuppTable1.txt"
[4] "../../input/official_models/kang_tcgaexprs.csv"
[5] "../../input/official_models/Kang_Table3_data.csv"
```

Load required libraries:

```
> library(survival)
> library(survcomp)
> library(affy)
> library(curatedOvarianData)
> library(survHD)
> library(HGNChelper)
```

Load the required datasets from curatedOvarianData:

```
> data(TCGA_eset, package="curatedOvarianData")
> data(GSE9891_eset, package="curatedOvarianData")
> data(PMID15897565_eset, package="curatedOvarianData")

> kang.tcgaexprs <- read.csv(input_tcgadata, as.is=TRUE, row.names=1)
> kang.table3data <- read.csv(input_table3data, as.is=TRUE)
> ##Keep only stage III-IV:
> kang.table3data <- kang.table3data[grep("III|IV", kang.table3data$tumor_stage), ]
> ##Keep only first-course platinum+taxane:
> kang.table3data <-
+     kang.table3data[na.omit(which(kang.table3data$Plat.taxane_first_course == 1)), ]
> dim(kang.table3data)   #304 patients used for training set in paper

[1] 304  21
```

Select these same patients from expression data:

```
> kang.tcgaexprs <- kang.tcgaexprs[ ,na.omit(match(kang.table3data$Identification,
+                                                   colnames(kang.tcgaexprs)))]
> stopifnot( all.equal( kang.table3data$Identification, colnames(kang.tcgaexprs)) )
```

Get rid of tumor_ at beginning and 01ABCD at end (see https://wiki.nci.nih.gov/display/TCGA/TCGA+barcode) of identifiers:

```
> colnames(kang.tcgaexprs) <- gsub("^tumor_|_01[ABCD]", "", colnames(kang.tcgaexprs))
```

Finally, substitute "." for "_" for consistency with curatedOvarianData:

```
> colnames(kang.tcgaexprs) <- gsub("_", ".", colnames(kang.tcgaexprs), fixed=TRUE)
```

Keep only curatedOvarianData samples which are also in the above version of the data:

```
> dim(TCGA_eset)

Features  Samples
   12981      578

> TCGA_eset <- TCGA_eset[, na.omit(match(colnames(kang.tcgaexprs), sampleNames(TCGA_eset)))]
> dim(TCGA_eset)

Features  Samples
   12981      304

> stopifnot( all.equal(sampleNames(TCGA_eset), colnames(kang.tcgaexprs)) )
```

Now we have data for the 304 samples used in the paper, as prepared by the authors and as provided by the curatedOvarianData package.

Read the data from Table 2 which defines the model:

```
> source.data <- read.delim(input_file, as.is=TRUE)
> source.data

       Gene Pathway Survival     P
1       ATM     ATM     high 0.120
2     H2AFX     ATM     high 0.026
3      MDC1     ATM     high 0.100
4      RNF8     ATM     high 0.020
5     TOP2A     ATM     high 0.110
6     BRCA2   FA/HR      low 0.069
7  C17orf70   FA/HR     high 0.059
8     FANCB   FA/HR     high 0.110
9     FANCE   FA/HR     high 0.055
10    FANCF   FA/HR     high 0.006
11    FANCG   FA/HR     high 0.047
```

```
12    FANCI   FA/HR     high 0.130
13    PALB2   FA/HR     high 0.034
14    MUS81   FA/HR     high 0.110
15      NBN   FA/HR      low 0.083
16    SHFM1   FA/HR     high 0.120
17     DDB1     NER     high 0.045
18    ERCC8     NER      low 0.110
19   RAD23A     NER     high 0.034
20      XPA     NER      low 0.140
21   MAD2L2     TLS     high 0.110
22     POLH     TLS     high 0.150
23    UBE2I     TLS     high 0.049
```

For each gene, “high” means higher than median gene expression was associated with improved overall survival in The Cancer Genome Atlas set, and “low” means higher than median gene expression was associated with worse overall survival.

```
> coefs.all <- ifelse(source.data$Survival=="high", -1, 1)
> names(coefs.all) <- source.data$Gene
```

Note that we are missing three of the 23 genes in curatedOvarianData:

```
> names(coefs.all)[!names(coefs.all) %in% featureNames(TCGA_eset)]
```

```
[1] "FANCB"  "FANCG"  "MAD2L2"
```

This ”voting” scheme is implemented in the survHD package. Genes in validation data will be median-centered and voting based on whether expression of the gene is above or below the median, so the model is defined simply as:

```
> model.official <- new("ModelLinear",
+                       coefficients=coefs.all,
+                       votingthresholds=0,
+                       modeltype="negativeriskvoting")
```

TCGA Cox analysis for comparison to Figure 1A. Create predictions using authors’ data:

```
> yhat.kang.23genes <- predict(model.official, newdata=t(kang.tcgaexprs), type="lp")@lp
```

Also make predictions using curatedOvarianData:

```
> tcga.exprs.cod <- exprs(TCGA_eset)[featureNames(TCGA_eset) %in% names(coefs.all), ]
> tcga.exprs.cod <- sweep(tcga.exprs.cod, 1, apply(tcga.exprs.cod, 1, median))
> yhat.cod <- predict(model.official, newdata=t(tcga.exprs.cod), type="lp")@lp
```

Create Surv object containing observed overall survival times:

```
> y <- Surv(TCGA_eset$days_to_death/365, TCGA_eset$vital_status=="deceased")
```

Our predictions yhat are negative, because they are both a risk score and consistent with the voting scheme proposed in this paper where the magnitude of the score increases with high expression of good-prognosis genes. Their absolute values correspond to the score discussed by Kang et al, although with a smaller magnitude of possible values since we have only 20 genes of their 23. The median and range of the score here are:

```
> summary(yhat.cod)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
    -18     -12     -10     -10      -8      -2 

> summary(yhat.kang.23genes)

   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  -20.0   -14.0   -12.0   -11.7   -10.0    -1.0 
```

Having confirmed that our implementation of the model is correct by reproducing Figure 1A, we now also do the survival analysis for the TCGA, Tothill, and Berchuck datasets, using curatedOvarianData, for comparison to Figure 1B.

First define a function which takes the ExpressionSet as input and returns the coefficients of the univariate Cox fit for each feature:

```
> getCoefs <- function(eset, model){
+     ##eliminate samples missing survival information:
+     if(sum( c(is.na(eset$days_to_death), is.na(eset$vital_status)) ) < 2*ncol(eset)){
+         ##has continuous overall survival
+         eset <- eset[, (!is.na(eset$days_to_death) & !is.na(eset$vital_status))]
+         y <- Surv(eset$days_to_death/30, eset$vital_status=="deceased")
+     }else{
+         eset <- eset[, !is.na(eset$os_binary)]
+         time <- rep(3.0*12, ncol(eset))
+         time[ eset$os_binary == "long" ] <- 7.0*12
+         cens <- eset$os_binary == "short"
+         y <- Surv(time, cens)
+     }
+     expr.mat <- exprs(eset)[featureNames(eset) %in% names(coefs.all), ]
+     expr.mat <- sweep(expr.mat, 1, apply(expr.mat, 1, median))
+     yhat <- predict(model, newdata=t(expr.mat), type="lp")@lp
+     fit <- coxph(y ~ (yhat < median(yhat)))
+     output <- summary(fit)$coefficients
+     return(output)
+ }
```

Note that Kang et al. use only the Tohill dataset patients who received a platinum and taxane regimen:

```
> GSE9891_eset_pltx <- GSE9891_eset[ ,which(GSE9891_eset$pltx=="y" & GSE9891_eset$tax=="y")]
```

Now do the Cox analysis for each dataset:

```
> all.coefs <- lapply(list(TCGA_eset, PMID15897565_eset, GSE9891_eset_pltx),
+                     getCoefs, model=model.official)
> all.coefs <- do.call(rbind, all.coefs)
> rownames(all.coefs) <- c("TCGA dataset", "Berchuck dataset", "Tothill dataset")
> round(all.coefs, 2)

                  coef exp(coef) se(coef)     z Pr(>|z|)
TCGA dataset     -0.59      0.56     0.17 -3.48     0.00
```

```
> plotKMStratifyBy(y=y, linearriskscore=yhat.kang.23genes,
+                   cutpoints=-10.5,
+                   n.risk.step=5)
```

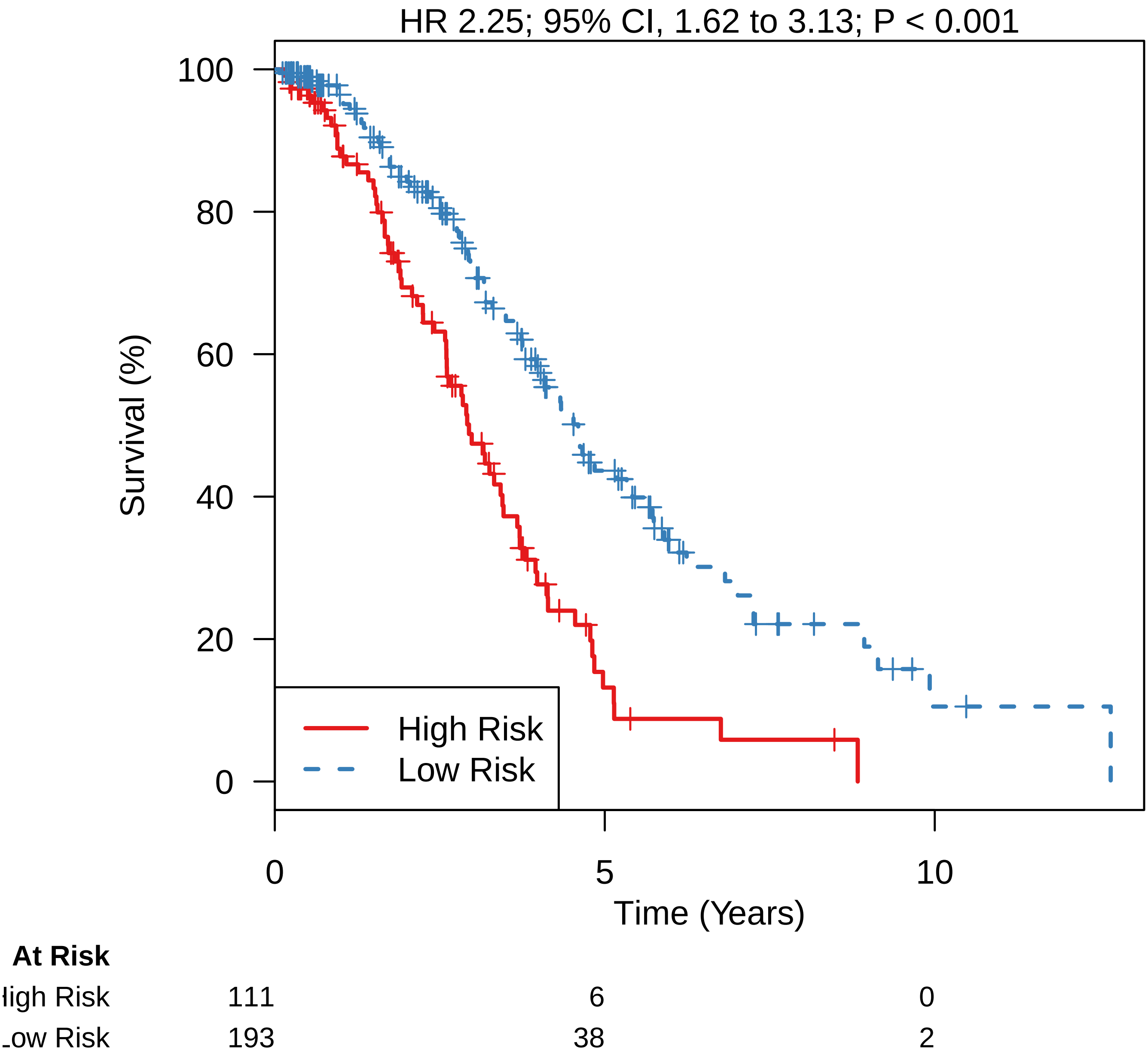


Figure 14: Kaplan-Meier plot for TCGA training set prepared using authors' data and methods, and the 23-gene voting risk score. This appears identical to Figure 1A.

```
Berchuck dataset -0.40      0.67     0.39 -1.02     0.31
Tothill dataset  -0.14      0.87     0.24 -0.58     0.56
```

And make a forest plot of the results using the official model:

Finally, we build a curatedOvarianData (COD) version of the model, using the prior-specified list of genes provided by the authors in Supplemental Table 1, but doing selection and coefficient determination independently using the curatedOvarianData version of the TCGA dataset (Affymetrix arrays, RMA normalized, mapped to gene symbols using BiomaRt).

These are the DNA-damage repair related genes provided in Supplemental Table 1:

```
> dna.repair.genes <- read.delim(input_suppfile, header=FALSE, as.is=TRUE)[ ,1]
> checked <- checkGeneSymbols(dna.repair.genes)
> checked[!checked$Approved, ]

          x Approved Suggested.Symbol
22   OBFC2B    FALSE            NABP2
98   PMS2L3    FALSE           PMS2P3
104 RAD51L1    FALSE           RAD51B
129 RAD51L3    FALSE           RAD51D

> dna.repair.genes <- checked$Suggested.Symbol
```

Some of these genes were not mapped by BiomaRt in curatedOvarianData:

```
> summary(dna.repair.genes %in% featureNames(TCGA_eset))

   Mode   FALSE    TRUE    NA's
logical      19     132       0

> dna.repair.genes[!dna.repair.genes %in% featureNames(TCGA_eset)]

 [1] "CLSPN"  "ALKBH2" "NHEJ1"  "SHPRH"  "RDM1"   "MAD2L2" "RAD18"  "POLN"
 [9] "GTF2H4" "FANCB"  "FANCG"  "FBXO18" "POLK"   "ATRIP"  "FANCD2" "NEIL2"
[17] "ALKBH3" "PMS2"   "FANCM"
```

So we will just use the ones which are available:

```
> TCGA.reduced <- TCGA_eset[featureNames(TCGA_eset) %in% dna.repair.genes, ]
> exprs(TCGA.reduced) <- sweep(exprs(TCGA.reduced), 1, apply(exprs(TCGA.reduced), 1, median))
> TCGA.reduced$y <- Surv(TCGA.reduced$days_to_death/30, TCGA.reduced$vital_status=="deceased")
```

Remove samples which are missing survival data:

```
> TCGA.reduced <- TCGA.reduced[ ,!is.na(TCGA.reduced$y)]
```

Train the model using survHD, using a p-value threshold of 0.15 as specified by the authors:

```
> model.cod <- survHD::plusMinusSurv(X=t(exprs(TCGA.reduced)),
+                                    y=TCGA.reduced$y,
+                                    lambda=0.15, tuningpar="pval",  #cutoff of p < 0.15
+                                    modeltype="negativeriskvoting")@model
```

```
> metaplot.surv(mn        = all.coefs[ ,"coef"],
+               se        = all.coefs[ ,"se(coef)"],
+               labels    = rownames(all.coefs),
+               main      = "Compare to Figure 1B",
+               logeffect = TRUE,
+               boxsize   = 0.5)
```

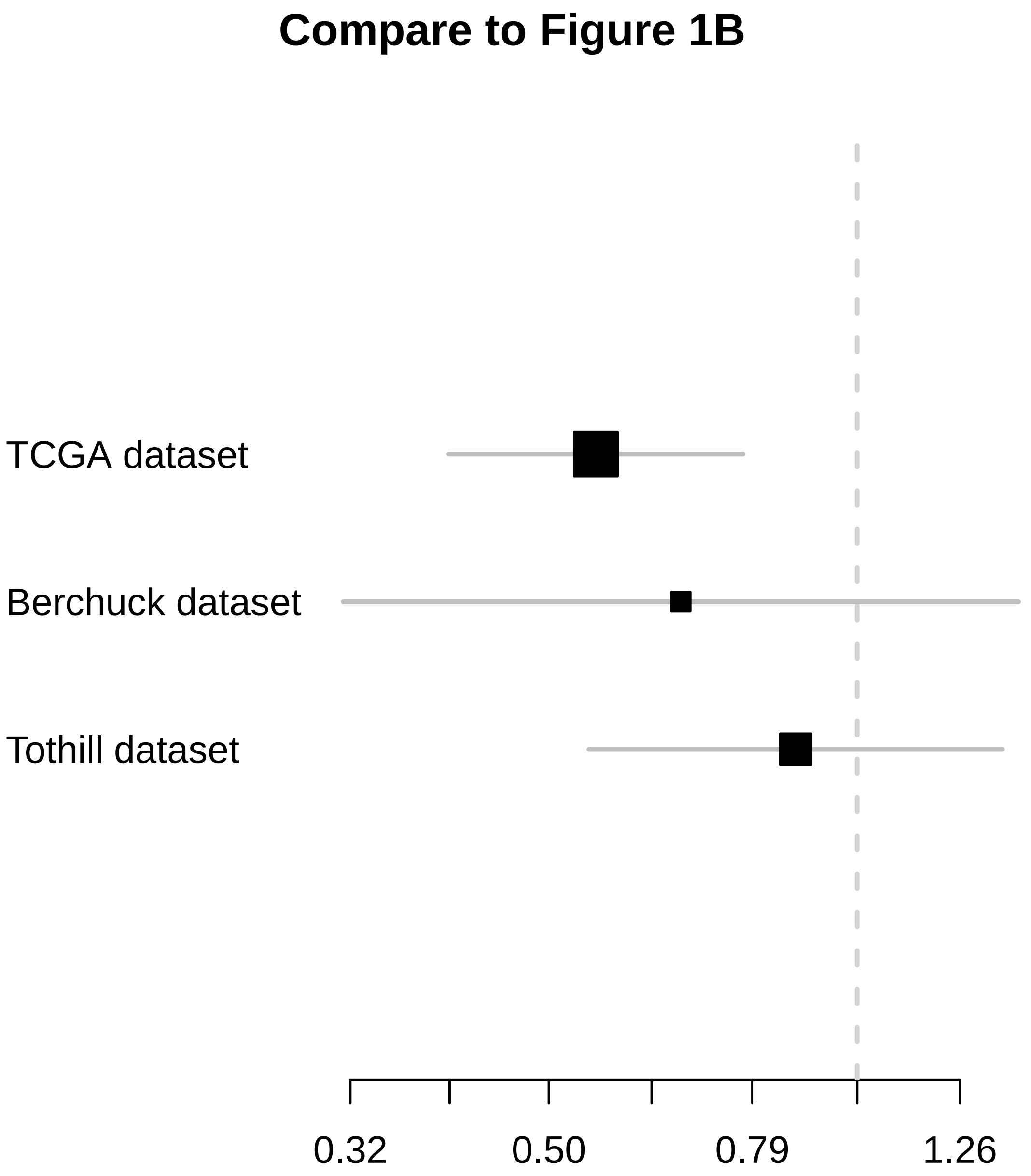


Figure 15: Forest plot for prediction of the voting model in validation sets, for comparison to Figure 1B. The direction of effects is consistent with Figure 1B, although their magnitudes are smaller, using the data as processed in the curatedOvarianData package.

Which genes are selected in the alternative COD model?

```
> coefs.cod <- model.cod@coefficients[abs(model.cod@coefficients) > 0]
> coefs.cod[coefs.cod < 0]  ##Good prognosis genes

  APEX1     ATM   CETN2 DCLRE1A   H2AFX    MLH3   NTHL1   PARP2    POLE    RAD1
     -1      -1      -1      -1      -1      -1      -1      -1      -1      -1
 RAD54L   RAD9A   SHFM1   SMUG1    TDP1   XRCC2
     -1      -1      -1      -1      -1      -1

> coefs.cod[coefs.cod > 0]  ##Bad prognosis genes

  BRCA2   ERCC1     NBN    PER1  RAD23B   RAD50   RAD52 TP53BP1
      1       1       1       1       1       1       1       1
```

There is relatively little overlap between the official and the curatedOvarianData versions of the model:

```
> length(model.official@coefficients)

[1] 23

> length(coefs.cod)

[1] 24

> length(intersect(names(coefs.cod), names(model.official@coefficients)))

[1] 5
```

Finally, make a forest plot equivalent to Figure 1B using the alternative COD model:

```
> all.coefs.cod <- lapply(list(TCGA_eset, PMID15897565_eset, GSE9891_eset_pltx),
+                          getCoefs, model=model.cod)
> all.coefs.cod <- do.call(rbind, all.coefs.cod)
> rownames(all.coefs.cod) <- c("TCGA dataset", "Berchuck dataset", "Tothill dataset")
> round(all.coefs.cod, 2)

                  coef exp(coef) se(coef)     z Pr(>|z|)
TCGA dataset     -0.55      0.58     0.17 -3.31     0.00
Berchuck dataset -0.19      0.83     0.38 -0.49     0.62
Tothill dataset  -0.03      0.97     0.23 -0.11     0.91
```

Although reported results seem to be sensitive to the expression data processing methods used, the model is fully-specified, so we save the official model from Table 2, and the version re-created using curatedOvarianData for further validation.

```
> save(model.official, model.cod, file=model_file)
> metaplot.surv(mn        = all.coefs.cod[ ,"coef"],
+               se        = all.coefs.cod[ ,"se(coef)"],
+               labels    = rownames(all.coefs.cod),
+               main      = "Compare to Figure 1B",
+               logeffect = TRUE,
+               boxsize   = 0.5)
```

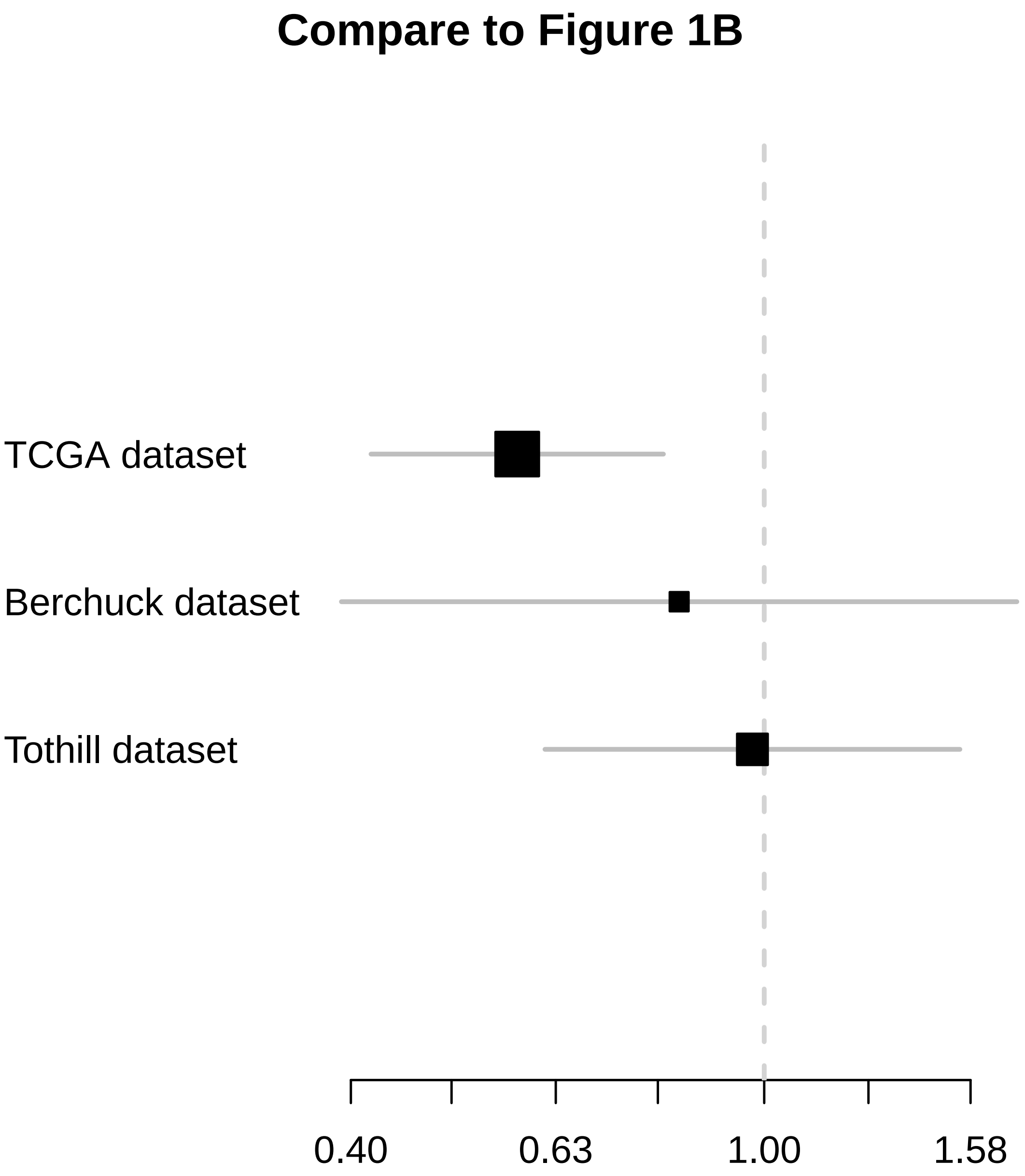


Figure 16: Alternative version of the model where the model coefficients are determined using the curatedOvarianData version of the TCGA dataset and the survHD package for model training. Results look similar to the official model, although with less separation in both validation sets.

The Cancer Genome Atlast 2011

Integrated genomic analyses of ovarian carcinoma. Nature 2011, 474:609-615.

Implemented by Ben Haibe-Kains and Levi Waldron.

We are able to reproduce the TCGA model fully, using the list of "good" and "poor" prognosis genes provided in Supplemental Table S6 and the t-score method described for calculating risk scores. Validation for this model matches very closely with the validation provided in the publication. We did not reproduce the model or performance as closely when attempting to create the model from scratch using the TCGA expression data from curatedOvarianData, so we use the former, "official" version for meta-analysis.

Input arguments:

```
> print(c(input_file, model_file))

[1] "../../input/official_models/TCGA_2010-09-11380C-Table_S6.1.xls"
[2] "21720365-SuppTable_S6.RData"
```

Load required libraries:

```
> library(survHD)
> library(curatedOvarianData)
> library(HGNChelper)
> library(gdata)   #for read.xls
> library(affy)
> library(biomaRt)
```

From the TCGA paper Supplement S6:

> Using a training dataset of gene expression profiles from 215 stage II-IV ovarian tumors from TCGA (batches 9, 11-15), a prognostic gene signature for overall survival was defined (genes with univariate Cox P < 0.01), comprised of 108 genes correlated with poor (worse) prognosis and 85 genes correlated with good (better) prognosis.

So prepare the training dataset, adding a leading "X" to Affymetrix probe set IDs to make them valid R names. Note that we have larger numbers of samples than reported in the paper.

```
> data(TCGA_eset, package="FULLVcuratedOvarianData")
> featureNames(TCGA_eset) <- paste("X", featureNames(TCGA_eset), sep="")  #to make probesets valid R names.
> TCGA.training.eset <- TCGA_eset[ ,TCGA_eset$batch %in% c("9", "11", "12", "13", "14", "15")]
> TCGA.training.eset <- TCGA.training.eset[ ,which(TCGA.training.eset$tumorstage == 2 |
+                                                 TCGA.training.eset$tumorstage == 3 |
+                                                 TCGA.training.eset$tumorstage == 4)]
> TCGA.training.eset <- TCGA.training.eset[ ,!is.na(TCGA.training.eset$days_to_death) &
+                                           !is.na(TCGA.training.eset$vital_status)]
> TCGA.training.eset$y <- Surv(TCGA.training.eset$days_to_death / 30,
+                               TCGA.training.eset$vital_status == "deceased")
> dim(TCGA.training.eset)  #215 samples reported in supplemental Methods

Features  Samples
   22277      229
```

Prepare validation set 1, a second set of the TCGA. For validation sets, we use curatedOvarianData package which summarizes probesets to gene symbols. Note that the TCGA supplement provides its model in terms of gene symbols, not probe sets.

```
> rm(TCGA_eset)
> data(TCGA_eset, package="curatedOvarianData")
> TCGA.validation.eset <- TCGA_eset[ ,TCGA_eset$batch %in% c("17", "18", "19", "21", "22", "24")]
> dim(TCGA.validation.eset)  #280 samples, rather than the 255 cited in Supplemental Methods S6

Features  Samples
   12981      280

> featureNames(TCGA.validation.eset) <- sub("-", "hyphen", featureNames(TCGA.validation.eset))
> TCGA.validation.eset <- TCGA.validation.eset[ ,!is.na(TCGA.validation.eset$days_to_death) &
+                                              !is.na(TCGA.validation.eset$vital_status)]
> TCGA.validation.eset$y <- Surv(TCGA.validation.eset$days_to_death / 30,
+                                 TCGA.validation.eset$vital_status == "deceased")
```

Validation set 2, Tothill et al.:

```
> ##Validation set 2:
> data(GSE9891_eset, package="curatedOvarianData")
> featureNames(GSE9891_eset) <- sub("-", "hyphen", featureNames(GSE9891_eset))
> GSE9891_eset <- GSE9891_eset[ ,!is.na(GSE9891_eset$days_to_death) &
+                               !is.na(GSE9891_eset$vital_status)]
> GSE9891_eset$y <- Surv(GSE9891_eset$days_to_death / 30, GSE9891_eset$vital_status == "deceased")
```

Validation set 3, Bonome et al. (2008):

```
> data(GSE26712_eset, package="curatedOvarianData")
> featureNames(GSE26712_eset) <- sub("-", "hyphen", featureNames(GSE26712_eset))
> GSE26712_eset <- GSE26712_eset[ ,!is.na(GSE26712_eset$days_to_death) &
+                                 !is.na(GSE26712_eset$vital_status)]
> dim(GSE26712_eset)

Features  Samples
   12981      185

> GSE26712_eset$y <- Surv(GSE26712_eset$days_to_death / 30,
+                          GSE26712_eset$vital_status == "deceased")
```

Validation set 4, Dressman et al:

```
> data(PMID17290060_eset, package="curatedOvarianData")
> featureNames(PMID17290060_eset) <- sub("-", "hyphen", featureNames(PMID17290060_eset))
> PMID17290060_eset <- PMID17290060_eset[ ,!is.na(PMID17290060_eset$days_to_death) &
+                                         !is.na(PMID17290060_eset$vital_status)]
> dim(PMID17290060_eset)
```

```
Features  Samples
   12981      117
```

```
> PMID17290060_eset$y <- Surv(PMID17290060_eset$days_to_death / 30,
+                              PMID17290060_eset$vital_status == "deceased")
```

Create the "official" version of the model, using the good and poor prognosis genes as specified in Supplemental Table S6.

```
> source.data <- gdata::read.xls(input_file, skip=2, header=TRUE, as.is=TRUE)
> source.data <- source.data[source.data$Gene.set == "poor" | source.data$Gene.set == "good", ]
> nrow(source.data)  #193 genes
```

```
[1] 193
```

This 193-gene signature contains some gene symbols not approved by HGNC; try to map these to HGNC-approved symbols:

```
> coefs <- ifelse(source.data$Gene.set == "poor", 1, -1)
> names(coefs) <- source.data$Name
> symbol.map <- checkGeneSymbols(source.data$Name)   #package HGNChelper
> symbol.map[!symbol.map$Approved, ]
```

```
                x Approved Suggested.Symbol
4          CCDC49    FALSE            CWC25
16           MLL4    FALSE             MLL2
19       FLJ10241    FALSE           ATP5SL
24    RP11-125A7.3    FALSE             <NA>
28         PERLD1    FALSE            PGAP3
32          PPM2C    FALSE             PDP1
48        C19ORF7    FALSE            ZC3H4
87        C20ORF4    FALSE             AAR2
89       FLJ20323    FALSE             MIOS
90        C20ORF3    FALSE            APMAP
92       C17ORF63    FALSE          FAM222B
105     C20ORF121    FALSE            TTPAL
126         THEM2    FALSE           ACOT13
141       TXNDC13    FALSE             TMX4
184     C14ORF159    FALSE        C14orf159
188       C6ORF64    FALSE           SAYSD1
189      FLJ14213    FALSE            PRR5L
```

```
> names(coefs) <- symbol.map$Suggested.Symbol
> coefs <- coefs[!is.na(names(coefs))]
> model.official <- new("ModelLinear", coefficients=coefs, modeltype="tscore")
```

Confirm that coefficients we get in the training set are similar. Fit a cox model on TCGA training samples from batches 9, 11, 12, 13, 14, and 15:

```
> data(TCGA_eset, package="curatedOvarianData")
> TCGA.training.eset.symbols <- TCGA_eset[ ,TCGA_eset$batch %in%
+                                         c("9", "11", "12", "13", "14", "15")]
> TCGA.training.eset.symbols$y <- Surv(TCGA.training.eset.symbols$days_to_death / 30,
+                                       TCGA.training.eset.symbols$vital_status == "deceased")
> featureNames(TCGA.training.eset.symbols) <- sub("-", "hyphen",
+                                                  featureNames(TCGA.training.eset.symbols))
> TCGA.training.eset.symbols <-
+     TCGA.training.eset.symbols[na.omit(match(source.data$Name,
+                                               featureNames(TCGA.training.eset.symbols))), ]
> TCGA.training.eset.symbols <-
+     TCGA.training.eset.symbols[ ,!is.na(TCGA.training.eset.symbols$y)]
> coxresults <- rowCoxTests(X=exprs(TCGA.training.eset.symbols),
+                           y=TCGA.training.eset.symbols$y)
```

Plot these Cox coefficients against those reported in the paper's supplement.

Now we try training the model using FULLVcuratedOvarianData (probesets) and the survHD package plusMinusSurv function, with the option modeltype="tscore", selecting genes with a Wald test p-value for the univariate Cox regression less than 0.01.

```
> model.probesets <- plusMinusSurv(X=t(exprs(TCGA.training.eset)),
+                                   y=TCGA.training.eset$y,
+                                     modeltype="tscore",
+                                     lambda=0.01, tuningpar="pval")@model
> ##prune zero coefficients:
> coefs.probesets <- model.probesets@coefficients[abs(model.probesets@coefficients) > 0]
> names(coefs.probesets) <- sub("X", "", names(coefs.probesets))  ##Remove leading X again.
```

Map probesets back to gene symbols:

```
> ensembl <- useMart("ensembl", dataset="hsapiens_gene_ensembl")
> ##listFilters(ensembl)
> platform <- "affy_hg_u133a"
> map <- getBM(attributes = c(platform, "hgnc_symbol"), mart=ensembl,
+              filters=platform, values=names(coefs.probesets))
> map <- map[map$hgnc_symbol != "", ]
> map <- map[na.omit(match(names(coefs.probesets), map[ ,1])), ]
> map$hgnc_symbol <- sub("-", "hyphen", map$hgnc_symbol)
> ##Do the mapping
> coefs.mapped <- data.frame(probesets=coefs.probesets[match(map[ ,1], names(coefs.probesets))])
> if( identical(rownames(coefs.mapped), map[ ,1]) ){
+     coefs.mapped$hgnc <- map[ ,2]   #hgnc_symbols
+ }else{
+     stop("Mapping error.")
+ }
> ##average coefficients with multiple probesets
> coefs.symbols <- aggregate(coefs.mapped$probesets, list(coefs.mapped$hgnc), mean)
> coefs.symbols <- structure(coefs.symbols[ ,2], .Names=coefs.symbols[ ,1])
> model.cod <- new("ModelLinear", coefficients=coefs.symbols, modeltype="tscore")
```

The attempt at reproducing the model produces a larger signature, with relatively small overlap:

```
> ##overlap
> length(model.official@coefficients)
> coxresults$original <- source.data$beta[match(rownames(coxresults), source.data$Name)]
> plot(coef~original, data=coxresults, asp=1)
> abline(v=0, lty=2)
> abline(h=0, lty=2)
> legend("topleft",
+        bty='n',
+        lty=-1, pch=-1,
+        legend=c("original: 215 samples", paste("curatedOvarianData:",
+                 ncol(TCGA.training.eset.symbols), "samples")))
```

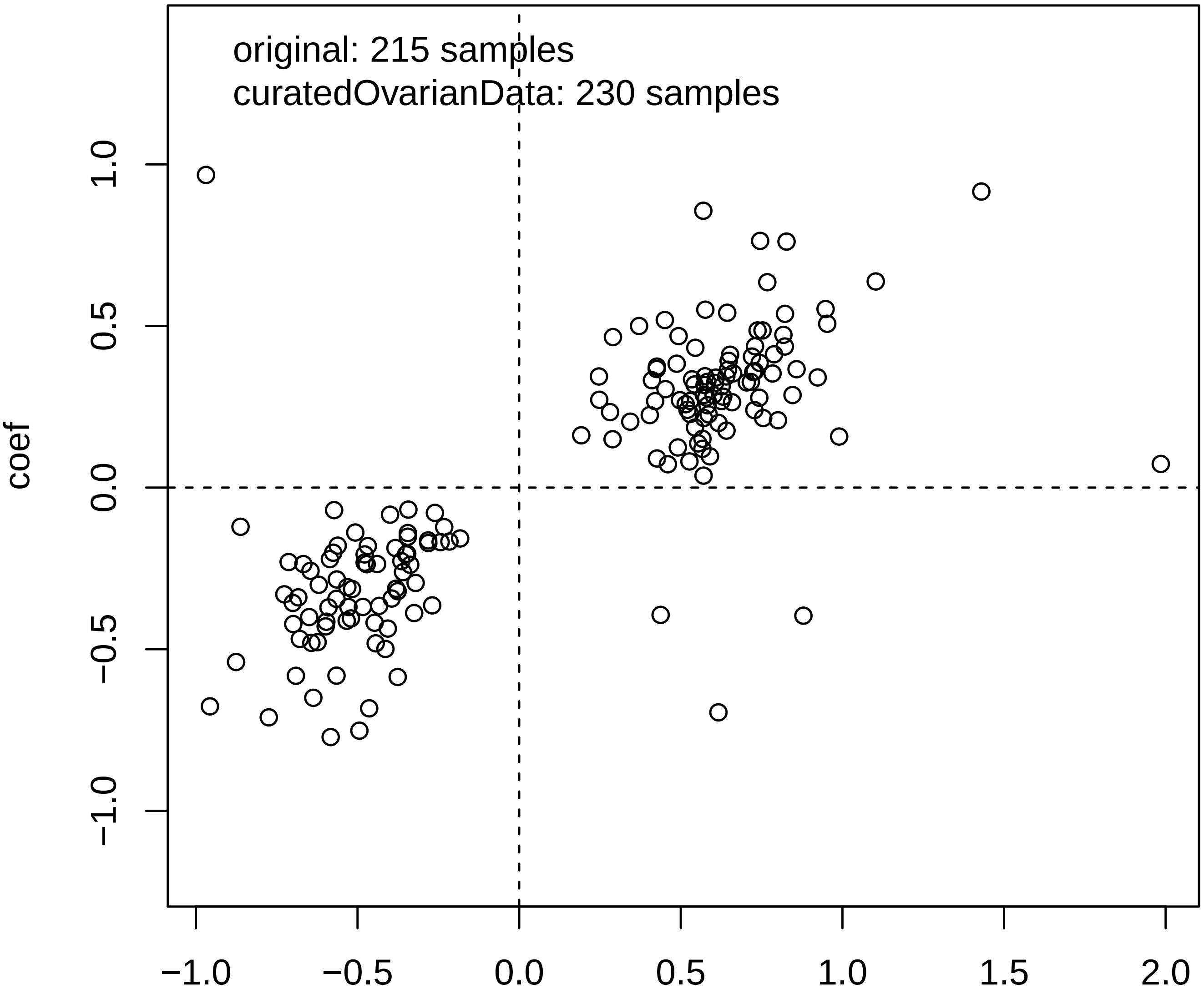


Figure 17: Cox coefficients reported in TCGA paper Supplemental S6.1 (original), plotted against coefficients determined for curatedOvarianData package. Correlation is good, but not exact. Note that the original paper reported using only 215 samples from batches 17, 18, 19, 21, 22, and 24.

```
[1] 192
```

```
> length(model.cod@coefficients)
```

```
[1] 312
```

```
> length(intersect(names(model.cod@coefficients), names(model.official@coefficients)))
```

```
[1] 49
```

validation set 1, TCGA validation set (using Official model):

validation set 1, TCGA validation set (using retrained model from curatedOvarianData):

validation set 2, Tothill validation set (using Official model):

validation set 2, Tothill validation set (using retrained model from curatedOvarianData):

validation set 3, Bonome et al. (bottom left-hand corner of Figure 2C):

validation set 4, Dressman et al. (bottom right-hand corner of Figure 2C):

Finally, save these models:

```
> save(model.official, model.cod, file=model_file)
> risk1A <- predict(model.official, newdata=t(exprs(TCGA.validation.eset)), type="lp")
> tmp <- plotKMStratifyBy("median",
+                        y=TCGA.validation.eset$y,
+                        linearriskscore=risk1A@lp,
+                        censor.at=60,
+                        main="Official model\n Compare to Figure 2C - TCGA test set")
```

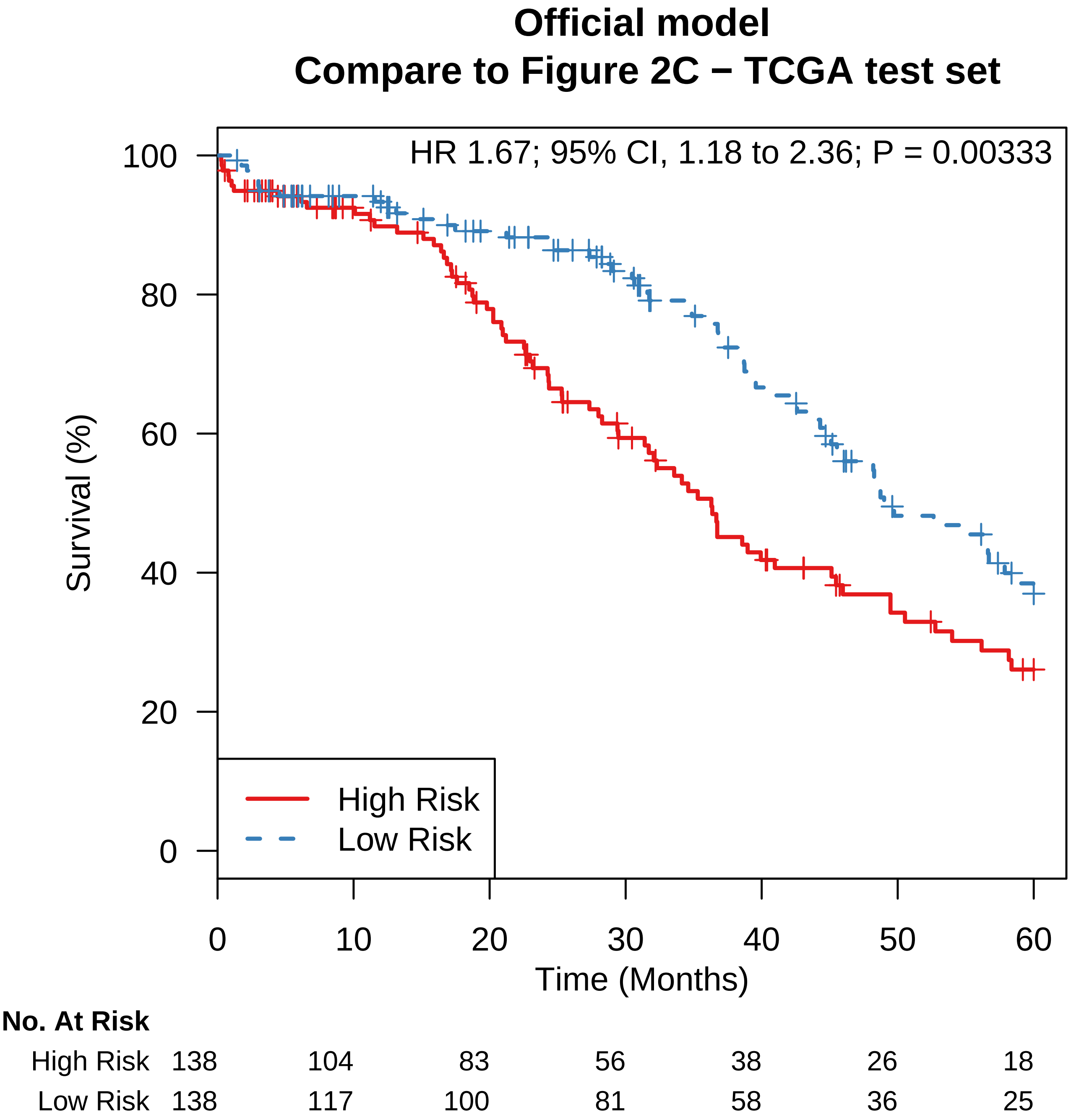


Figure 18: First validation of Figure 2C using the official TCGA model provided in Supplemental file 2010-09-11380C-Table$_S$6.1.$xls.Looksidenticalalthoughthepaperreports$N=255.

```
> risk1B <- predict(model.cod, newdata=t(exprs(TCGA.validation.eset)), type="lp")
> tmp <- plotKMStratifyBy("median", y=TCGA.validation.eset$y, linearriskscore=risk1B@lp,
+                   censor.at=60,
+                   main="COD model\n Compare to Figure 2C - TCGA test set")
```

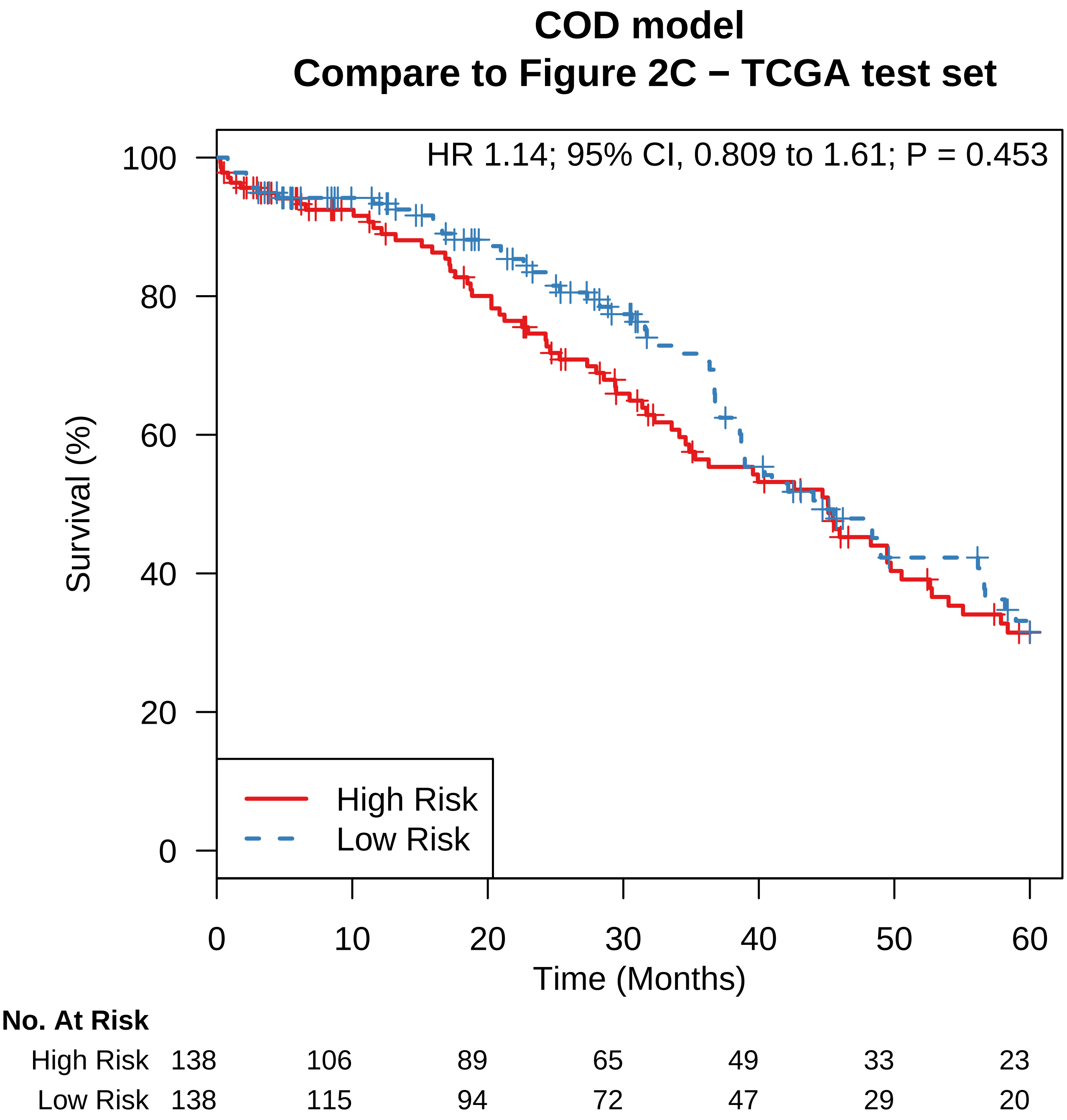


Figure 19: First validation of Figure 2C using the TCGA methods but with curatedOvarianData (probesets already mapped to genes, patients added since original analysis). Validation is not as good as the official model.

```
> risk2A <- predict(model.official, newdata=t(exprs(GSE9891_eset)), type="lp")
> tmp <- plotKMStratifyBy("median", y=GSE9891_eset$y, linearriskscore=risk2A@lp,
+                   censor.at=60,
+             main="Official model\n Compare to Figure 2C ref. 25")
```

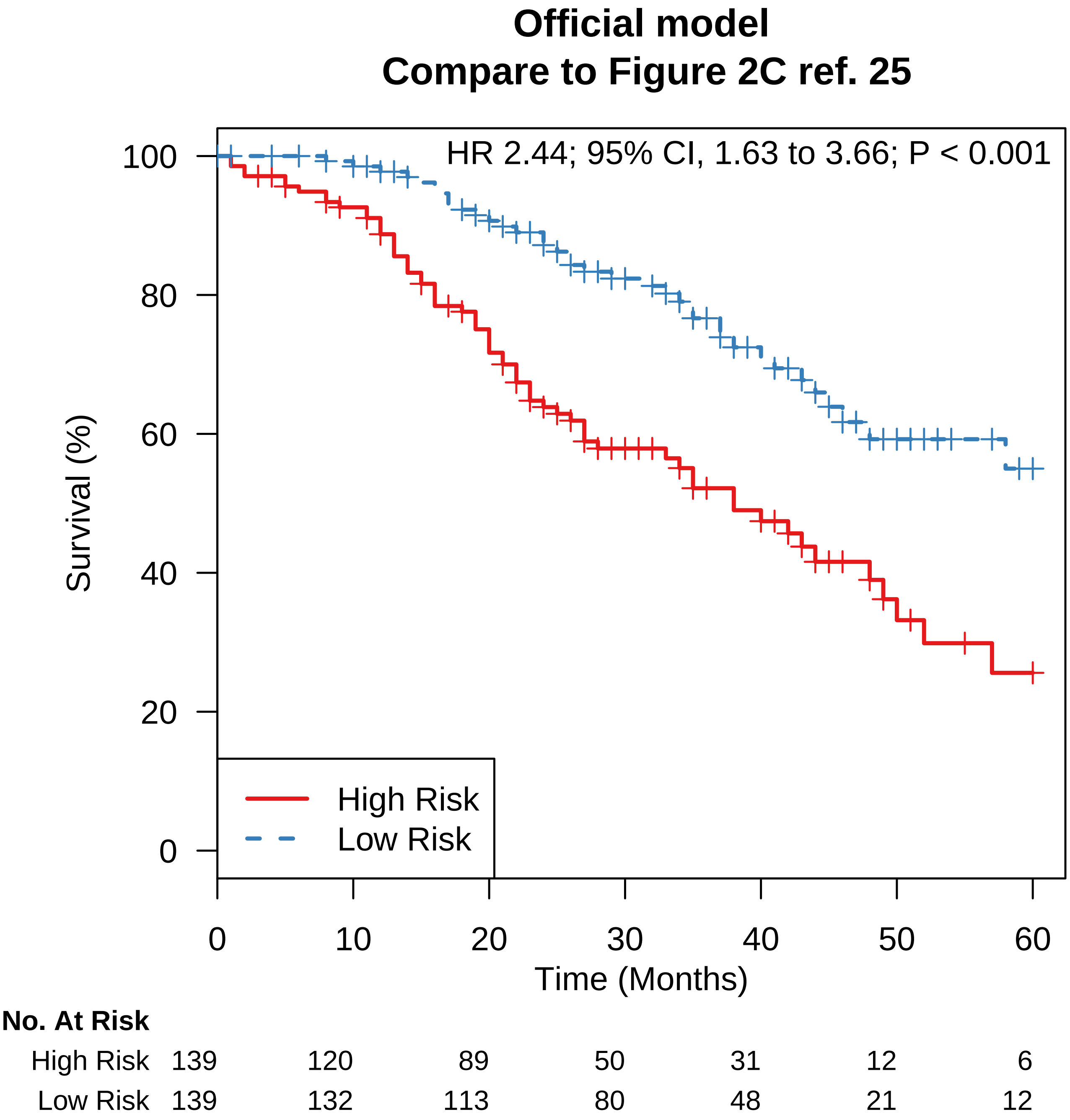
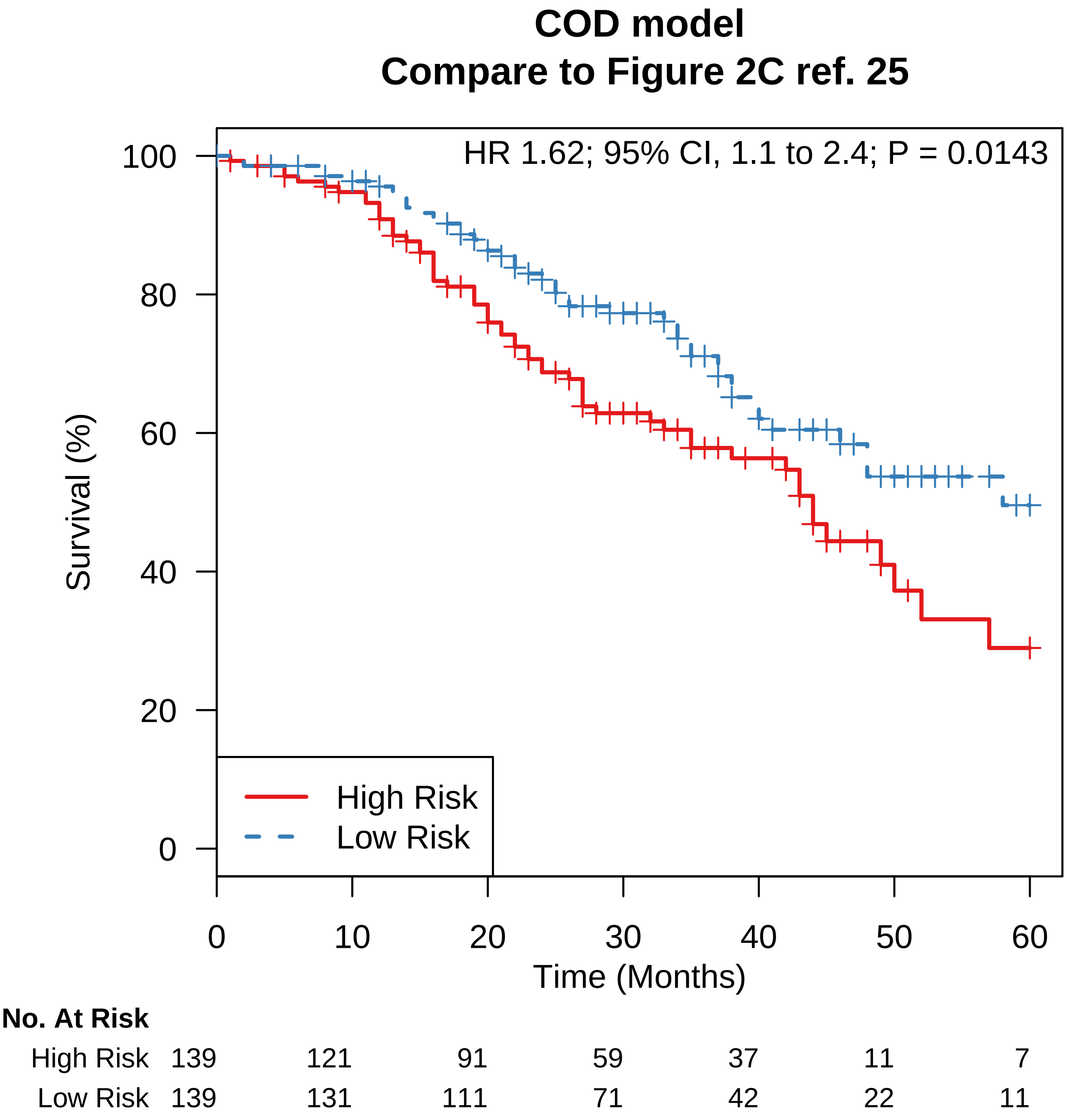


Figure 20: Upper right-hand corner validation of Figure 2C (Tothill et al) using the official TCGA model provided in Supplemental file 2010-09-11380C-Table$_S6.1.xls.Looksidenticalalthoughthepaperreports$N=237.

```
> risk2B <- predict(model.cod, newdata=t(exprs(GSE9891_eset)), type="lp")
> tmp <- plotKMStratifyBy("median", y=GSE9891_eset$y, linearriskscore=risk2B@lp,
+                     censor.at=60,
+                     main="COD model\n Compare to Figure 2C ref. 25")
```



No. At Risk

High Risk	139	121	91	59	37	11	7
Low Risk	139	131	111	71	42	22	11

Figure 21: Upper right-hand corner validation of Figure 2C (Tothill et al), retraining with curatedOvarianData package.

```
> risk3A <- predict(model.official, newdata=t(exprs(GSE26712_eset)), type="lp")
> tmp <- plotKMStratifyBy("median", y=GSE26712_eset$y, linearriskscore=risk3A@lp,
+                   censor.at=60,
+             main="Official model\n Compare to Figure 2C ref. 29")
```

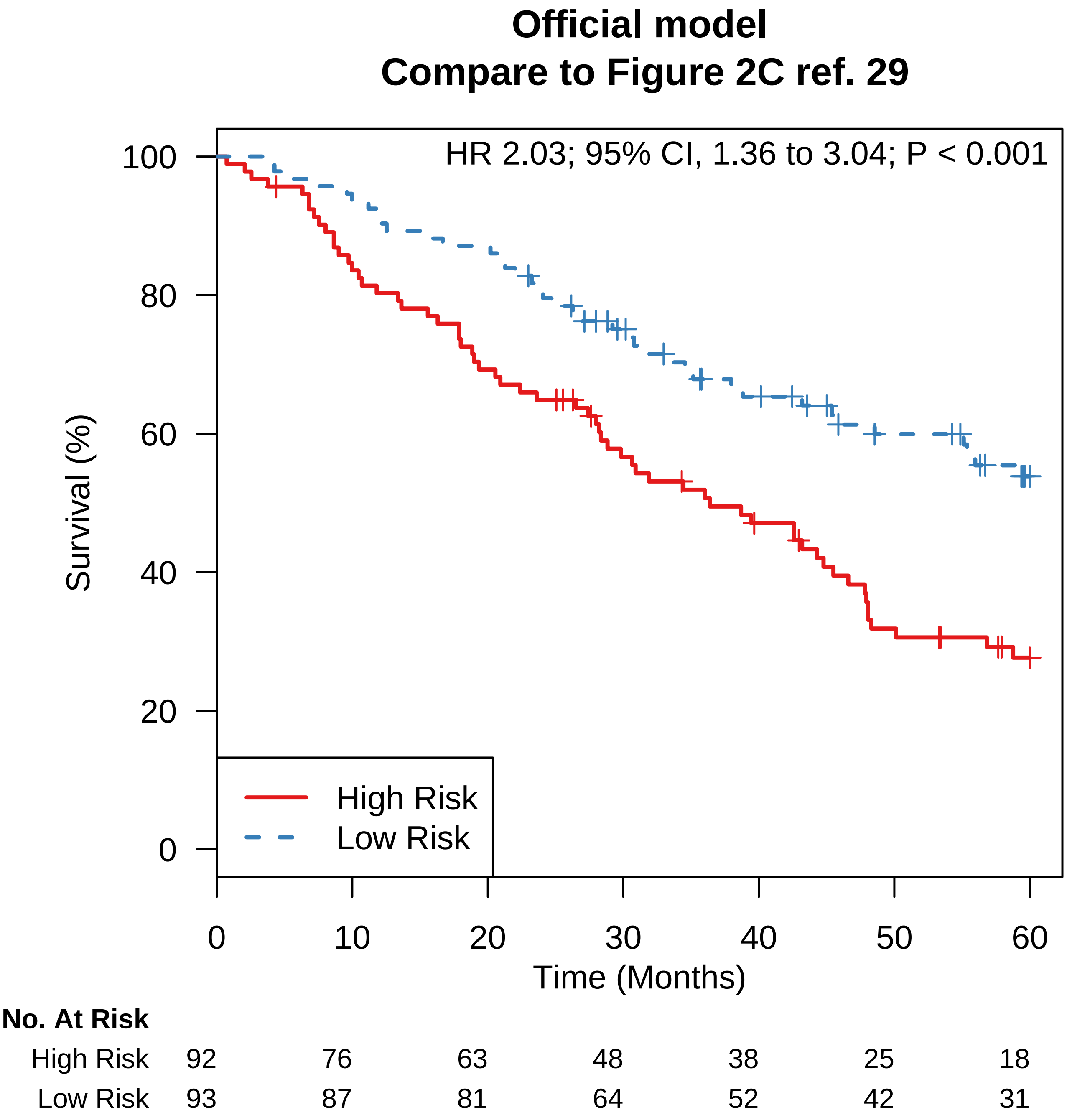


Figure 22: Lower left-hand corner validation of Figure 2C (Bonome 2008 et al). Looks identical although the paper reports $N = 169$.

```
> risk3B <- predict(model.cod, newdata=t(exprs(GSE26712_eset)), type="lp")
> tmp <- plotKMStratifyBy("median", y=GSE26712_eset$y, linearriskscore=risk3B@lp,
+                    censor.at=60,
+             main="COD model\n Compare to Figure 2C ref. 29")
```

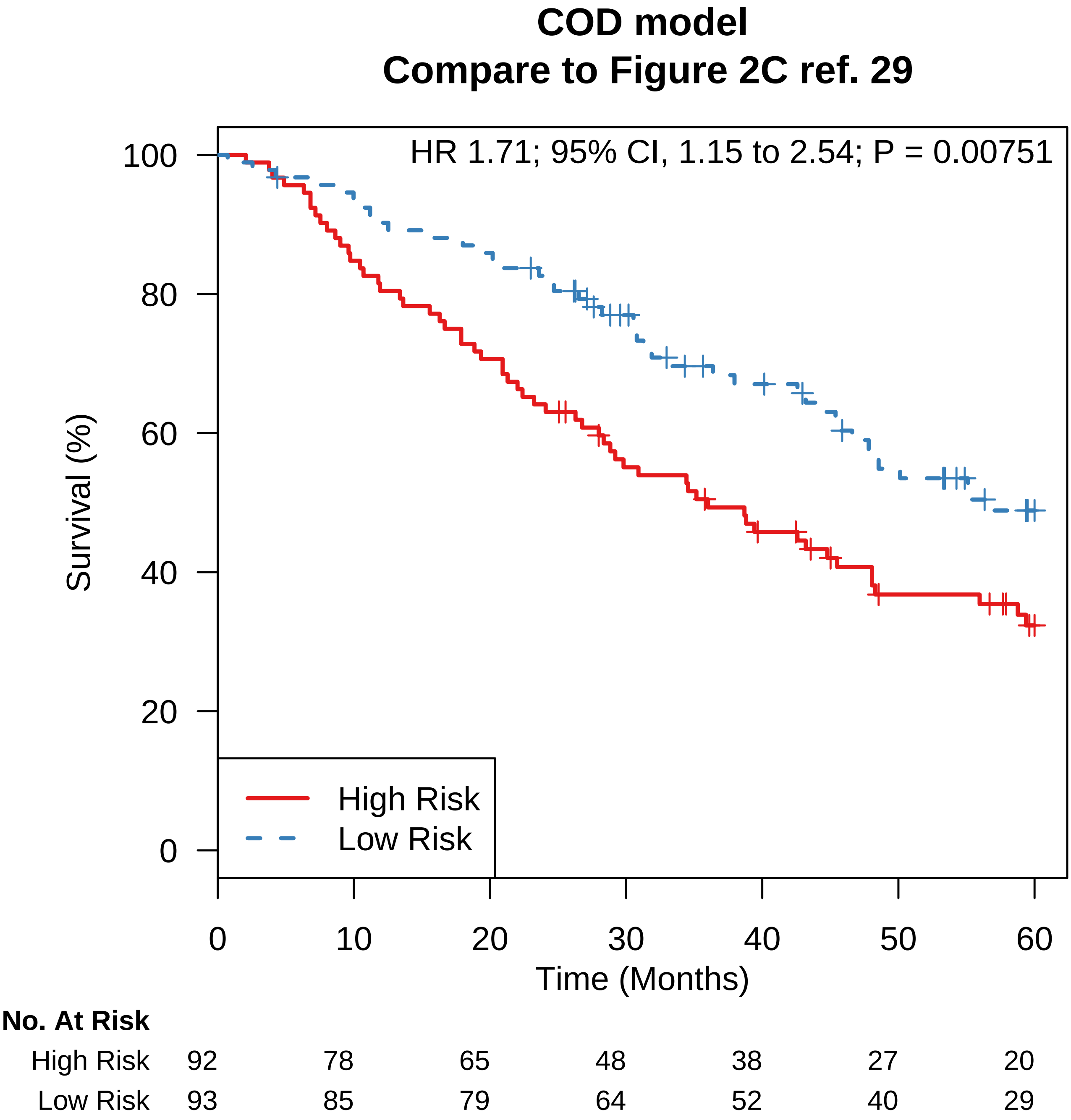


Figure 23: Lower left-hand corner validation of Figure 2C (Bonome 2008 et al), retraining with curatedOvarianData package. Validation is very similar but slightly better than with official model.

```
> risk4A <- predict(model.official, newdata=t(exprs(PMID17290060_eset)), type="lp")
> tmp <- plotKMStratifyBy("median", y=PMID17290060_eset$y, linearriskscore=risk4A@lp,
+                   censor.at=60,
+            main="Official model\n Compare to Figure 2C ref. 30")
```

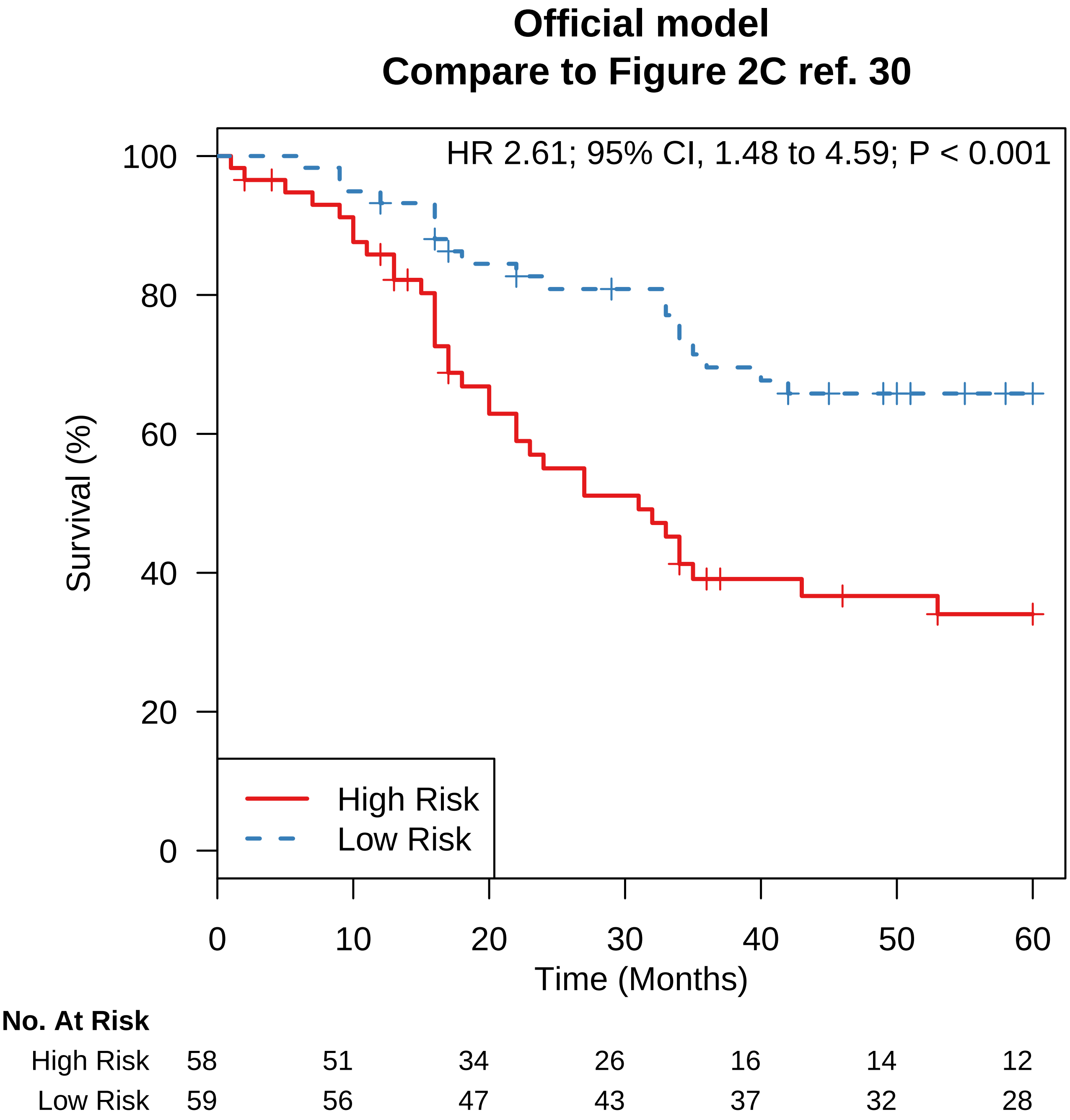


Figure 24: Lower right-hand corner validation of Figure 2C (Dressman et al., PMID 17290060), using the official TCGA model provided in Supplemental file 2010-09-11380C-Table$_S6.1.xls.Looksidenticalalthoughthepaperreports$N=118.

```
> risk4B <- predict(model.cod, newdata=t(exprs(PMID17290060_eset)), type="lp")
> tmp <- plotKMStratifyBy("median", y=PMID17290060_eset$y, linearriskscore=risk4B@lp,
+                    censor.at=60,
+                    main="COD model\n Compare to Figure 2C ref. 30")
```

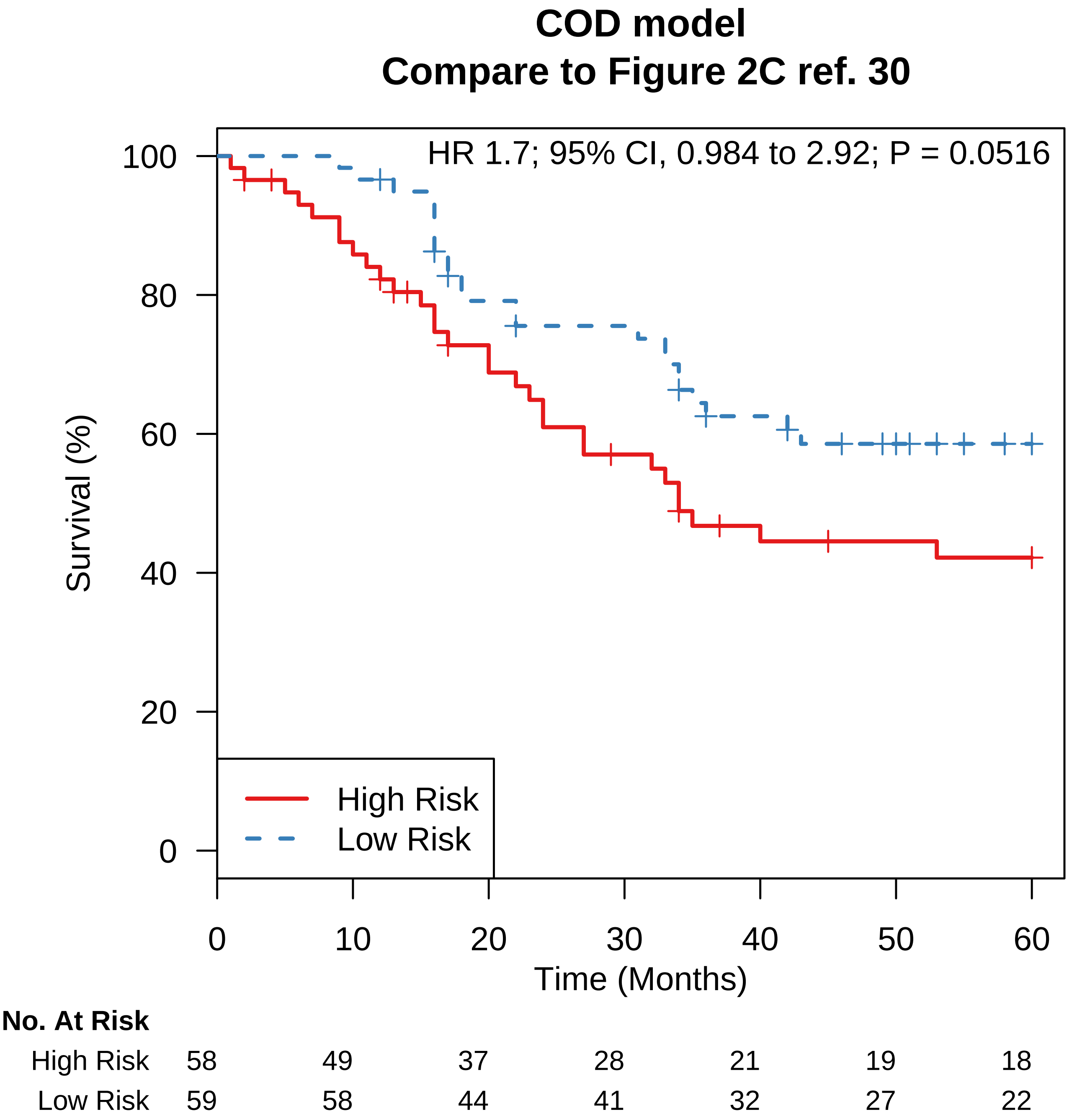


Figure 25: Retraining using TCGA methods with curatedOvarianData as training set, validation in Dressman (2008) validation set. Not as good as the official model.

Sabatier 2011

R Sabatier, P Finetti, J Bonensea, J Jacquemier, J Adelaide, E Lambaudie, P Viens, D Birnbaum and F Bertucci. A seven-gene prognostic model for platinum-treated ovarian carcinomas. *British Journal of Cancer* (2011) 105, 304 - 311.

Implemented by Levi Waldron.

Availability of training data is not stated in the paper. However, the seven genes of the proposed risk score are provided, along with their association with good or poor prognosis. So we will build a risk score from these seven genes, using coefficients of +/-1.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/Sabatier11-7gene.txt"
[2] "21654678-SuppTable3.RData"
```

Load required libraries:

```
> library(survHD)
```

Load good/poor prognosis genes provided on p. 307 and in Supplemental Table 3:

```
> source.data <- read.delim(input_file, as.is=TRUE)
> source.data
```

```
       gene        class
1      A1BG unfavourable
2       PAG unfavourable
3    SLC7A2   favourable
4     ALCAM   favourable
5   TMPRSS3   favourable
6    TSPAN6   favourable
7 C14orf101   favourable
```

Numeric coefficients are not provided, so we assume coefficients of +1 for poor prognosis genes and -1 for good prognosis genes:

```
> coefs <- ifelse(source.data$class=="unfavourable", 1, -1)
> names(coefs) <- source.data$gene
```

Create a survHD ModelLinear model:

```
> model.official <- new("ModelLinear", coefficients=coefs, modeltype="plusminus")
```

Save the model:

```
> save(model.official, file=model_file)
```

Yoshihara 2010

Yoshihara K, Tajima A, Yahata T, Kodama S, Fujiwara H, Suzuki M, Onishi Y, Hatae M, Sueyoshi K, Fujiwara H, Kudo Y, Kotera K, Masuzaki H, Tashiro H, Katabuchi H, Inoue I, Tanaka K. *Gene expression profile for predicting survival in advanced-stage serous ovarian cancer across two independent datasets.* PLoS One 2010 Mar 12;5(3):e9615. PMID: 20300634

Implemented by Benjamin Haibe-Kains, tested by Levi Waldron.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/yoshihara2010_sig.csv"
[2] "20300634-T2.RData"
```

Load required libraries:

```
> library(survHD)
> library(curatedOvarianData)
> library(affy)
> library(HGNChelper)
```

The coefficients of the 88-gene signature are provided by the authors in Table 2:

```
> Yoshi2010.sig <- read.csv(input_file, as.is=TRUE)
> head(Yoshi2010.sig)
```

```
       GenBank GeneSymbol Cytoband beta_ridge
1    NM_001123        ADK  10q22.2      0.006
2    NM_006408       AGR2   7p21.1      0.128
3    NM_080429      AQP10   1q21.3     -0.162
4 NM_001040118      ARAP1  11q13.4      0.141
5    NM_006420    ARFGEF2 20q13.13      0.032
6    NM_181575       AUP1   2p13.1      0.129
```

```
> dim(Yoshi2010.sig)
```

```
[1] 88  4
```

Check for and correct invalid HGNC symbols:

```
> hgnc.corrections <- checkGeneSymbols(Yoshi2010.sig$GeneSymbol)
> hgnc.corrections[!hgnc.corrections$Approved, ]
```

```
           x Approved Suggested.Symbol
10   C13orf3    FALSE             SKA3
12  C1orf230    FALSE           RIIAD1
13 C20orf177    FALSE          FAM217B
32   FAM176A    FALSE            EVA1A
```

```
> Yoshi2010.sig$Corrected.Gene.Symbol <- hgnc.corrections$Suggested.Symbol
```

Create the official model:

```
> coefs <- Yoshi2010.sig$beta_ridge
> names(coefs) <- Yoshi2010.sig$Corrected.Gene.Symbol
> model.official <- new("ModelLinear", coefficients=coefs, modeltype="plusminus")
```

Test using curatedOvarianData package. First, load the data:

```
> data(GSE17260_eset, package="curatedOvarianData")
> Yoshi2010.sig.exp <- exprs(GSE17260_eset)
> Yoshi2010.sig.exp <- Yoshi2010.sig.exp[rownames(Yoshi2010.sig.exp) %in% names(coefs), ]
> dim(Yoshi2010.sig.exp)  ##compare to 88 genes in author dataset, we lose some during mapping in curatedOva
```

```
[1]  85 110
```

```
> names(coefs)[!names(coefs) %in% rownames(Yoshi2010.sig.exp)]  ##these are the genes lost in the expression
```

```
[1] "EHD1"    "EVA1A"   "TMEM189"
```

Z-transfrom expression data:

```
> Yoshi2010.sig.z <- (Yoshi2010.sig.exp - rowMeans(Yoshi2010.sig.exp)) /
+   apply(Yoshi2010.sig.exp,1,sd)
```

Calculate scores:

```
> Yoshi2010.score <- predict(model.official, newdata=t(Yoshi2010.sig.z), type="lp")@lp
```

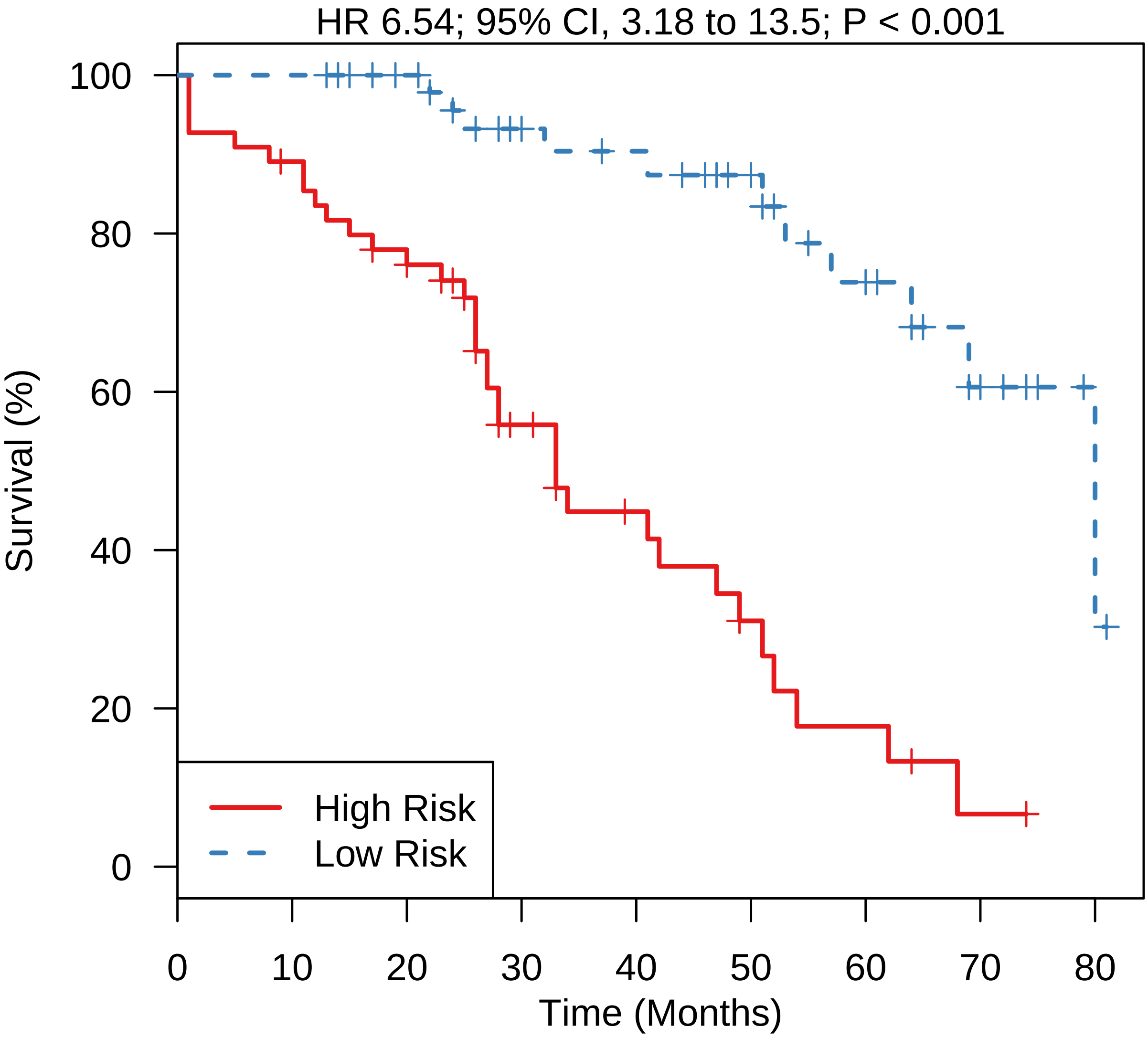
Make Kaplan-Meier curves for training set using overall survival (Figure 1C):

Now load and subset the independent test data (GSE9891, Tothill et al.), and do the filtering specified in the paper:

> From this original dataset (n = 285), we selected 87 samples that were (i) diagnosed as advanced-stage serous adenocarcinoma, (ii) treated by platinum/taxane-based chemotherapy, (iii) obtained from primary lesion, and (iv) followed-up for more than 12 months (Table S1).

Samples are not annotated as adenocarcinoma or not, but we can apply the rest of these filters:

```
> data(GSE9891_eset, package="curatedOvarianData")
> ## (i) advanced stage serous
> GSE9891_eset <- GSE9891_eset[, GSE9891_eset$tumorstage %in% c(3, 4)]    #advanced stage
> GSE9891_eset <- GSE9891_eset[, GSE9891_eset$histological_type == "ser"]  #serous
> ## (ii) treated by platinum/taxane-based chemotherapy
> GSE9891_eset <- GSE9891_eset[, GSE9891_eset$pltx == "y"]     #platinum treated
> GSE9891_eset <- GSE9891_eset[, GSE9891_eset$tax == "y"]      #taxane treated
> ## (iii) obtained from primary lesion, ie primary and arrayed site are the same:
> GSE9891_eset <- GSE9891_eset[, GSE9891_eset$arrayedsite == GSE9891_eset$primarysite]
> plotKMStratifyBy("median",
+                    y=Surv(GSE17260_eset$days_to_death/30, GSE17260_eset$vital_status == "deceased"),
+                    linearriskscore=Yoshi2010.score)
>
```



No. At Risk

High Risk	55	48	41	22	13	7	4	1	0
Low Risk	55	55	47	35	30	23	15	7	2

Figure 26: Compare to Figure 1C, training set overall survival. Nearly identical to Figure 1C which was produced with different methods for microarray preprocessing and collapsing duplicate probesets.

```
> ## (iii) 12 months follow-up using recurrence and survival:
> GSE9891_eset <- GSE9891_eset[, !(GSE9891_eset$days_to_death < 12*30 & GSE9891_eset$vital_status == "living
> GSE9891_eset <- GSE9891_eset[, !(GSE9891_eset$days_to_tumor_recurrence < 12*30 & GSE9891_eset$recurrence_s
> ## Analysis requires samples with overall survival, recurrence, and debulking information
> GSE9891_eset <- GSE9891_eset[, !is.na(GSE9891_eset$days_to_death)]
> GSE9891_eset <- GSE9891_eset[, !is.na(GSE9891_eset$days_to_tumor_recurrence)]
> GSE9891_eset <- GSE9891_eset[, !is.na(GSE9891_eset$debulking)]
> ##still have too many patients (paper said 87), but no other basis to reject them.
> dim(GSE9891_eset)
```

```
Features  Samples 
   19093       98 
```

Scale genes to z-scores:

```
> tothill.data <- exprs(GSE9891_eset)
> tothill.data <- tothill.data[rownames(tothill.data) %in% names(coefs), ]
> dim(tothill.data)
```

```
[1] 87 98
```

```
> tothill.data.z <- (tothill.data - rowMeans(tothill.data)) /
+   apply(tothill.data,1,sd)
```

Calculate scores and create overall survival object:

```
> tothill.score <- predict(model.official, newdata=t(tothill.data.z), type="lp")@lp
> tothill.os <- Surv(time=GSE9891_eset$days_to_death/30, event=GSE9891_eset$vital_status=='deceased')
```

Finally, save the model:

```
> save(model.official, file=model_file)
> scores.highlow <- factor(ifelse(tothill.score >= median(Yoshi2010.score), "high", "low"))
> plotKM(y=tothill.os, strata=scores.highlow, xlim=c(0, 100))
```

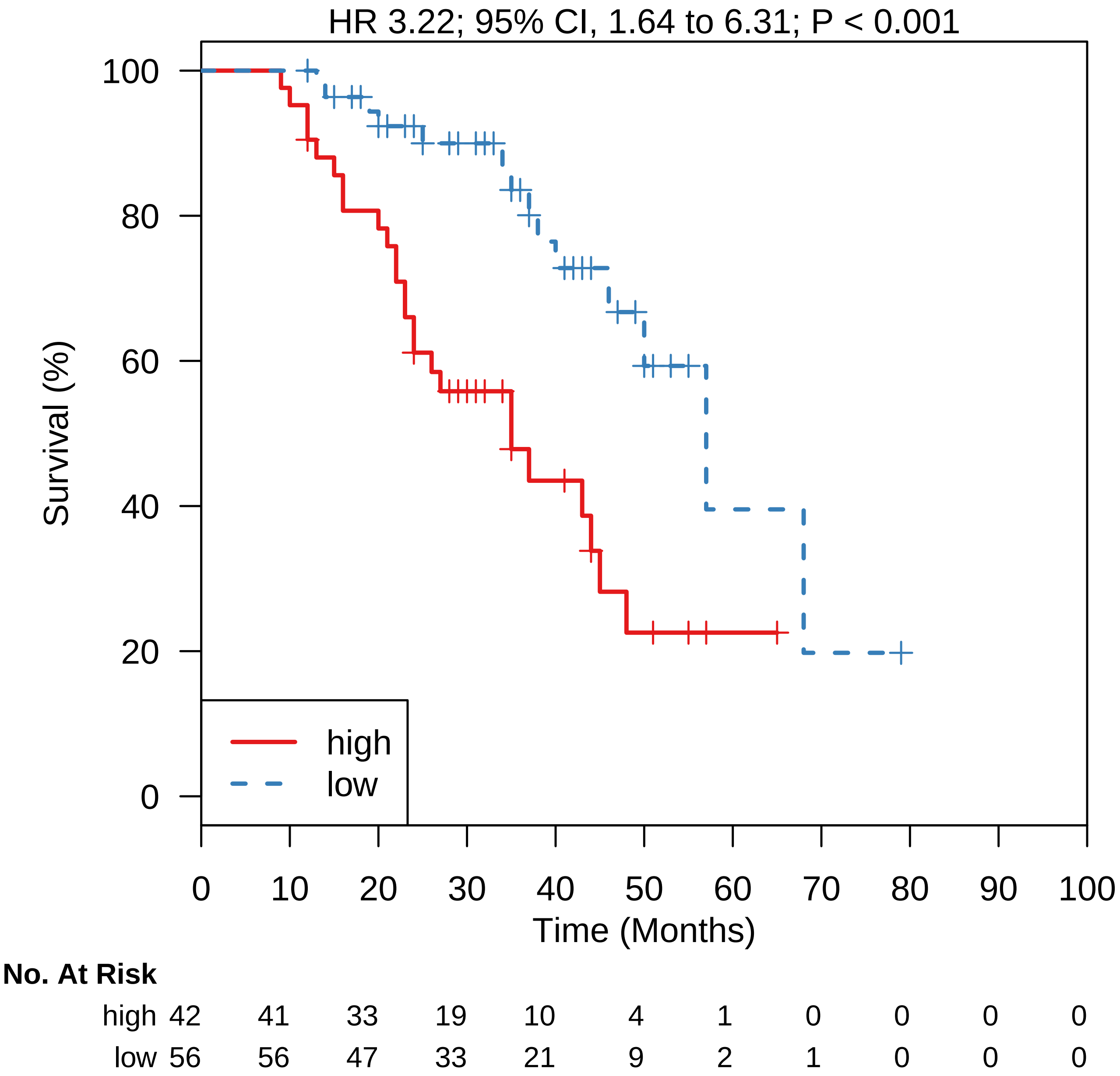


Figure 27: Similar to Figure 1D, test set for overall survival, using training set median for cutoff.

Konstantinopoulos 2010 (BRCAness signature)

Konstantinopoulos PA, Spentzos D, Karlan BY, Taniguchi T, Fountzilas E, Francoeur N, Levine DA, Cannistra SA. *Gene expression profile of BRCAness that correlates with responsiveness to chemotherapy and with outcome in patients with epithelial ovarian cancer.* J Clin Oncol. 2010 Aug 1;28(22):3555-61. Epub 2010 Jun 14. PMID: 20547991.

Implemented by Levi Waldron.

The authors developed one score for the Affymetrix HGU133 Plus2 platform, and another score for the HGU95Av2 platform. Only the weights the HGU133 Plus2 platform are provided with the paper (Supplemental Table 2), so we test this model.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/Supplementary_Table_1.xlsx"
[2] "20547991-ST1.RData"
```

Load required libraries:

```
> library(survHD)
> library(curatedOvarianData)
> library(affy)
> library(HGNChelper)
> library(gdata)
> library(survival)
```

The coefficients of the 60-gene signature are provided by the authors in Supplemental Table 2:

```
> Konstantinopoulos2010.sig <- read.xls(input_file, as.is=TRUE)
> head(Konstantinopoulos2010.sig)
```

```
  X Gene.Symbol                                          Gene.Description Weight
1 1        DAD1                               defender against cell death 1 0.0997
2 2       RAD21                                   RAD21 homolog (S. pombe) 0.1743
3 3        LDHA                                    lactate dehydrogenase A 0.0165
4 4       SPARC   secreted protein, acidic, cysteine-rich (osteonectin) 0.1571
5 5        SKP1                         S-phase kinase-associated protein 1 0.1370
6 6      PPP1CC protein phosphatase 1, catalytic subunit, gamma isoform 0.0249
```

```
> dim(Konstantinopoulos2010.sig)
```

```
[1] 60  4
```

Check for and correct invalid HGNC symbols:

```
> hgnc.corrections <- checkGeneSymbols(Konstantinopoulos2010.sig$Gene.Symbol)
> hgnc.corrections[!hgnc.corrections$Approved, ]
```

```
      x Approved  Suggested.Symbol
24 CDC2    FALSE    CDK1 /// POLD1
43  P11    FALSE ENDOU /// S100A10
```

```
> Konstantinopoulos2010.sig$Corrected.Gene.Symbol <- as.character(hgnc.corrections$Suggested.Symbol)
```

Create the official model. Note that high values of the score predicted by these coefficients predicts the BRCA-like profile, which is lower risk. Therefore we will use the *negative* of these coefficients to predict risk.

```
> coefs <- Konstantinopoulos2010.sig$Weight
> names(coefs) <- Konstantinopoulos2010.sig$Corrected.Gene.Symbol
> model.official <- new("ModelLinear",
+                       coefficients=-coefs,
+                       modeltype="plusminus")
```

We will test this model using GSE19829 from the curatedOvarianData package (N=70, for comparison to Fig. 3 in the paper).

```
> data(GSE19829.GPL570_eset, package="curatedOvarianData")
> data(GSE19829.GPL8300_eset, package="curatedOvarianData")
```

These two platforms must be merged to get N=70 as seen in Figure 3B. The authors trained two different models (one for each microarray platform), then combine the predictions of these models for Figure 3B. However we need a model which can be applied across platforms for fair comparison to all other models, so we explore the possibility of using the provided weights and gene symbols provided on both platforms of the validation set.

To do this, we will combine the datasets then apply batch correction. To combine the platforms, we take the intersection of the genes found on both platforms:

```
> available.genes <- intersect(names(coefs),
+                              intersect(featureNames(GSE19829.GPL570_eset),
+                                        featureNames(GSE19829.GPL8300_eset)))
> Konstantinopoulos2010.exp.GPL570 <- exprs(GSE19829.GPL570_eset)[available.genes, ]
> Konstantinopoulos2010.exp.GPL8300 <- exprs(GSE19829.GPL8300_eset)[available.genes, ]
```

Combine these into a single expression matrix:

```
> Konstantinopoulos2010.exp <- cbind(Konstantinopoulos2010.exp.GPL570,
+                                    Konstantinopoulos2010.exp.GPL8300)
```

The authors used ComBat to merge these datasets. We create a function here to enable the use of ComBat on R objects rather than external data files:

```
> ComBatWrapper <-
+     function(expr.uncorrected, batchvar,
+              combat.source="http://www.bu.edu/jlab/wp-assets/ComBat/Download_files/ComBat.R",
+              cleanup=TRUE,
+              prior.plots=FALSE){
+         source(combat.source)
+         sam.info <- data.frame("Array name"=colnames(expr.uncorrected),
+                                "Sample name"=colnames(expr.uncorrected),
+                                Batch=batchvar,
+                                check.names=FALSE)
+         write.table(expr.uncorrected, "combat_expression_xls.txt",
+                     quote=FALSE, sep="\t", row.names=TRUE)
+         write.table(sam.info, file="combat_sample_info_file.txt",
+                     quote=FALSE,sep="\t",row.names=FALSE)
+         ComBat(expression_xls="combat_expression_xls.txt",
+                sample_info_file="combat_sample_info_file.txt",
+                prior.plots=prior.plots,
+                skip=1)
+         expr.corrected <-
+             as.matrix(
+                       read.delim(paste("Adjusted_combat_expression_xls.txt_.xls",sep=""),
+                                  row.names=1))
+         if(cleanup)
+             unlink(c("combat_expression_xls.txt",
+                      "combat_sample_info_file.txt",
+                      "Adjusted_combat_expression_xls.txt_.xls"))
+         return(expr.corrected)
+     }
```

Create the batch variable for ComBat correction:

```
> batchvar <- c(rep("GPL570", ncol(Konstantinopoulos2010.exp.GPL570)),
+               rep("GPL8300", ncol(Konstantinopoulos2010.exp.GPL8300)))
```

Now we run the ComBat algorithm:

```
> Konstantinopoulos2010.exp.combat <- ComBatWrapper(Konstantinopoulos2010.exp, batchvar)
```

Note that if the ComBat.R link provided as the function default is unavailable, the version used here is also saved in this public repository at: input/official$_{m}odels/ComBat.R$.

Now calculate risk scores:

```
> Konstantinopoulos2010.score <- predict(model.official,
+                                         newdata=t(Konstantinopoulos2010.exp.combat), type="lp")@lp
```

Figure 3B shows 20 BL tumors and 50 NBL-like tumors, so we set the threshold using this ratio:

```
> Konstantinopoulos2010.group <- ifelse(rank(Konstantinopoulos2010.score) > 20,
+                                       "NBL profile", "BL profile")
> Konstantinopoulos2010.group <- factor(Konstantinopoulos2010.group)
```

We test the model by reproducing Figure 3B (overall survival in N=70 cohort).

```
> Konstantinopoulos2010.time <- c(GSE19829.GPL570_eset$days_to_death,
+                                  GSE19829.GPL8300_eset$days_to_death) / 30
> Konstantinopoulos2010.cens <- c(GSE19829.GPL570_eset$vital_status,
+                                  GSE19829.GPL8300_eset$vital_status) == "deceased"
> Konstantinopoulos2010.surv <- Surv(Konstantinopoulos2010.time, Konstantinopoulos2010.cens)
```

Make Kaplan-Meier plot to compare to Figure 3B:

Finally, save the model:

```
> save(model.official, file=model_file)
> plotKM(y=Konstantinopoulos2010.surv,
+        strata=Konstantinopoulos2010.group,
+        n.risk.step=24,
+        inverse.HR=FALSE)
```

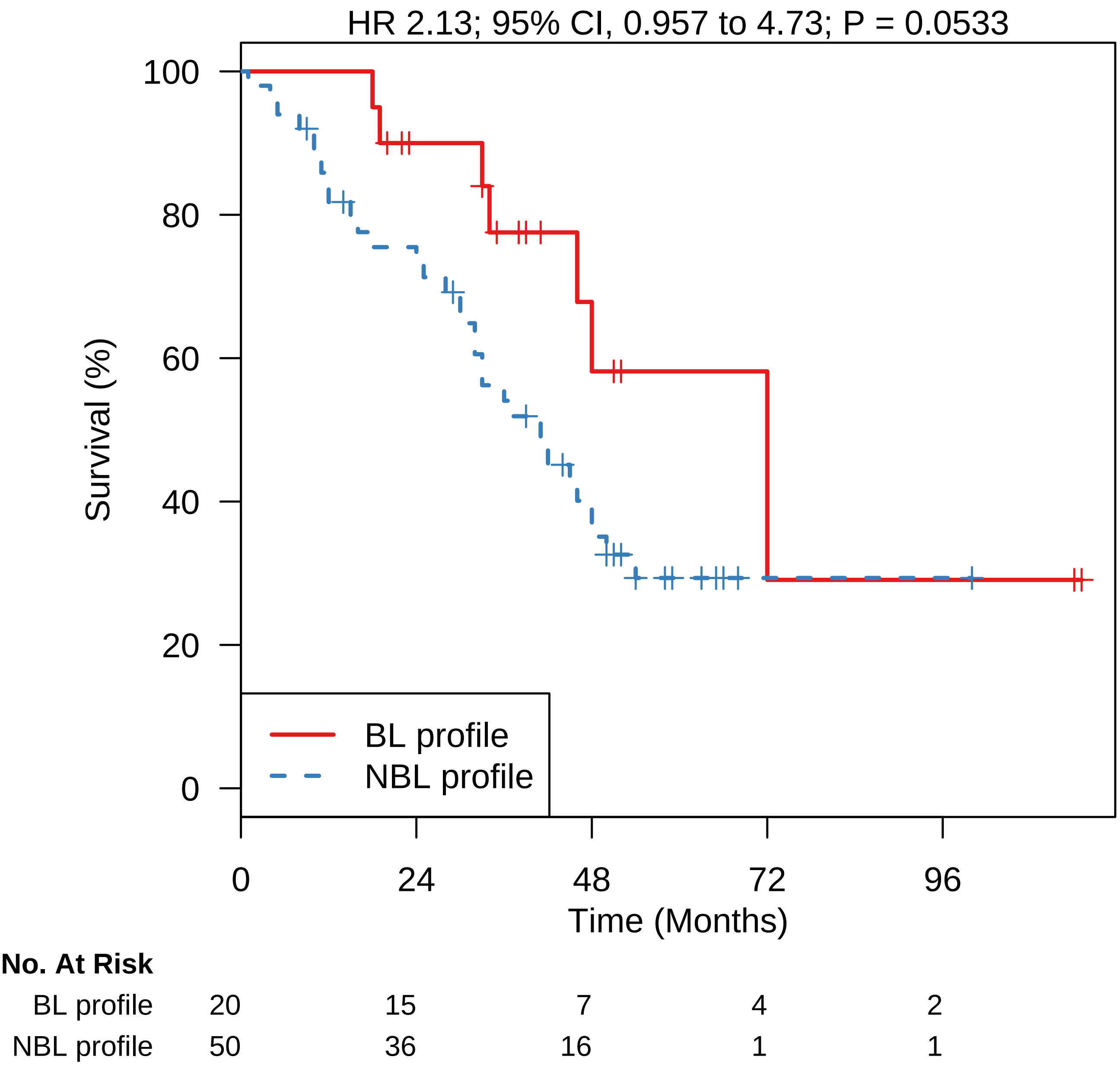


Figure 28: Compare to Figure 3B, validation set overall survival. The Hazard Ratio here is not quite as good (HR=2.13 here vs HR=3.29 in the paper), but the shape of the curves is very similar. In particular the sizes of the risk sets seen here are identical to Figure 3B except a difference of one patient at T=24mo (in the paper, the No. at risk for the NBL profile are 50-35-16-1-1). Additionally, the shape of these curves appears identical to Figure 3B between 0-24mo. So although this model not tuned to each platform, it performs nearly as well, and is worth testing on additional external datasets. Note also that for comparison to other models in additional data we will use this model as a continuous risk score, not dividing into discrete BL and NBL categories.

Hernandez 2010

Hernandez L, Hsu SC, Davidson B, Birrer MJ, Kohn EC, Annunziata CM. *Activation of NF-kappaB signaling by inhibitor of NF-kappaB kinase beta increases aggressiveness of ovarian cancer.* Cancer Research 2010 May 15;70(10):4005-14.

Implemented by Levi Waldron.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/Hernandez10-9gene.csv"
[2] "20424119-MaterialsandMethods.RData"
```

Load required libraries:

```
> library(curatedOvarianData)
> library(survHD)
> library(survival)
> library(GEOquery)
```

Load the 9-gene signature. Note that eight genes are down-regulated by an NFKB inhibitor, so are positively correlated with NFKB. One gene is negatively correlated with NFKB (PTGS1). However the caption of Figure 3 says, “Samples are ranked according to the average expression of the nine IKKB target genes” implying that PTGS1 is treated the same as the others. For this reason we create a column “poscoefficient” with coefficients of +1, which will use a simple average of the expression as a risk score.

```
> hernandez.sig <- read.csv(input_file, as.is=TRUE)
> hernandez.sig$poscoefficient <- 1
> hernandez.sig
```

```
    gene    probeset coefficient poscoefficient
1  CLDN1 218182_s_at           1              1
2  CXCL1   204470_at           1              1
3  CXCL2 209774_x_at           1              1
4    IL8 211506_s_at           1              1
5 INSIG1 201627_s_at           1              1
6  ITGB6 208083_s_at           1              1
7 PTGER2   206631_at           1              1
8  PTGS1 215813_s_at          -1              1
9   SOD2   215078_at           1              1
```

In order to validate across multiple platforms we would like to use gene symbols rather than particular probesets, so we will compare both versions here. First create a gene symbols version of the model:

```
> coefs.gene <- hernandez.sig$poscoefficient
> names(coefs.gene) <- hernandez.sig$gene
> model.official.gene <- new("ModelLinear", coefficients=coefs.gene, modeltype="plusminus")
```

Also create a probe version:

```
> coefs.probeset <- hernandez.sig$poscoefficient
> names(coefs.probeset) <- hernandez.sig$probeset
> model.official.probeset <- new("ModelLinear",
+                                 coefficients=coefs.probeset,
+                                 modeltype="plusminus")
```

We validate the signature in the Bonome et al. (2008) dataset. First using the gene symbols provided in the paper, then using the specific probesets and the FULLVcuratedOvarianData package (fRMA processed for GSE26712). Finally, we use the processed data from GEO as we believe the authors did, although no information on the handling of public data is provided in the paper.

Get gene-centered data from curatedOvarianData:

```
> data(GSE26712_eset, package="curatedOvarianData")
> GSE26712_eset <- GSE26712_eset[ ,!is.na(GSE26712_eset$days_to_death)]
```

It is not necessary to remove unused features from the expression data when using the survHD predict function, but we do it here to see how similar the probesets used by curatedOvarianData are to those proposed by the authors. In the curatedOvarianData package, representative probesets are chosen as those with the maximum mean across all datasets of that same platform.

```
> GSE26712_eset <- GSE26712_eset[ match(names(coefs.gene), featureNames(GSE26712_eset)), ]
> author.COD.comparison <- pData(featureData(GSE26712_eset))
> author.COD.comparison$hernandez_probeset <- names(coefs.probeset)
> author.COD.comparison$equal <-
+     author.COD.comparison$probeset == author.COD.comparison$hernandez_probeset
> author.COD.comparison
```

```
          probeset   gene hernandez_probeset equal
CLDN1  218182_s_at  CLDN1        218182_s_at  TRUE
CXCL1    204470_at  CXCL1          204470_at  TRUE
CXCL2  209774_x_at  CXCL2        209774_x_at  TRUE
IL8    202859_x_at    IL8        211506_s_at FALSE
INSIG1   201626_at INSIG1        201627_s_at FALSE
ITGB6    208084_at  ITGB6        208083_s_at FALSE
PTGER2   206631_at PTGER2          206631_at  TRUE
PTGS1  205128_x_at  PTGS1        215813_s_at FALSE
SOD2   221477_s_at   SOD2          215078_at FALSE
```

Create the expression matrix and median center each gene as specified in the caption of Figure 3 ("Expression was centered based on the median value").

```
> bonome08.data.gene <- t(exprs(GSE26712_eset))
> dim(bonome08.data.gene)
```

```
[1] 185   9
```

```
> bonome08.data.gene <- sweep(bonome08.data.gene, 2, apply(bonome08.data.gene, 2, median))
> apply(bonome08.data.gene, 2, median)  ##confirm median centering
```

```
 CLDN1  CXCL1  CXCL2    IL8 INSIG1  ITGB6 PTGER2  PTGS1   SOD2
     0      0      0      0      0      0      0      0      0
```

Create the Surv object and get the prediction scores.

```
> bonome08.surv <- Surv(GSE26712_eset$days_to_death/30, GSE26712_eset$vital_status == "deceased")
> bonome08.scores.gene <- predict(model.official.gene, newdata=bonome08.data.gene, type="lp")@lp
```

We now do the same exercise, but using the probesets specified by the authors. We use FULLVcuratedOvarianData, where the GSE26712 dataset is normalized by fRMA.

```
> data(GSE26712_eset, package="FULLVcuratedOvarianData")
> GSE26712_eset <- GSE26712_eset[ ,!is.na(GSE26712_eset$days_to_death)]
> GSE26712_eset <- GSE26712_eset[ match(names(coefs.probeset), featureNames(GSE26712_eset)), ]
> dim(GSE26712_eset)
```

```
Features  Samples 
       9      185 
```

```
> all.equal(featureNames(GSE26712_eset), hernandez.sig$probeset)
```

```
[1] TRUE
```

```
> bonome08.data.probeset <- t(exprs(GSE26712_eset))
> bonome08.data.probeset <- sweep(bonome08.data.probeset, 2,
+                                 apply(bonome08.data.probeset, 2, median))
> bonome08.scores.probeset <- predict(model.official.probeset,
+                                     newdata=bonome08.data.probeset, type="lp")@lp
```

Now use the GEO version of GSE26712, which may have been used by the authors:

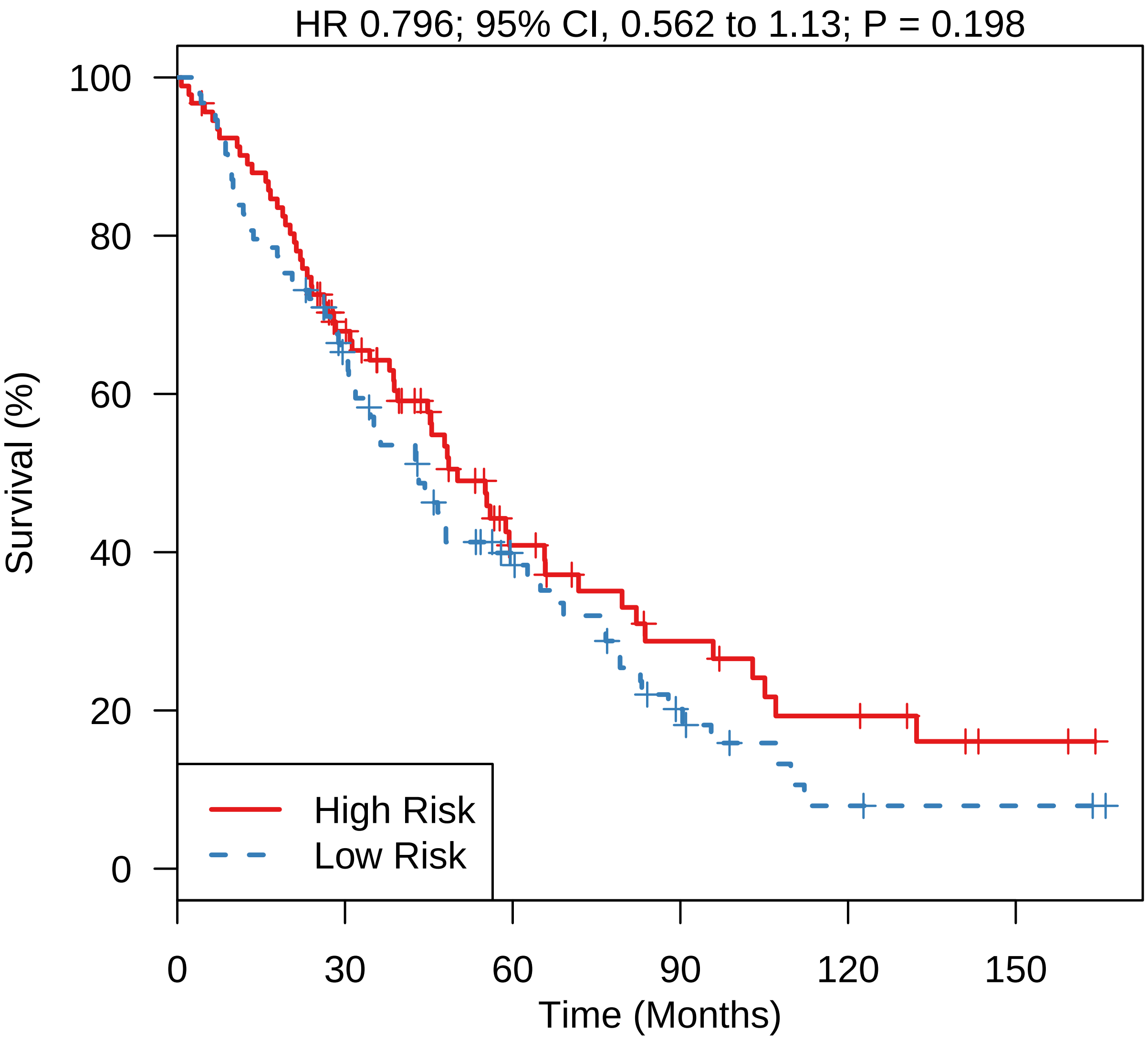
```
> GSE26712 <- getGEO("GSE26712")[[1]]
> GSE26712$time <- as.numeric(sub("survival years: ", "", GSE26712$characteristics_ch1.3))
> GSE26712$cens <- ifelse(grepl("dead", GSE26712$characteristics_ch1.2), 1, 0)
> GSE26712 <- GSE26712[ ,!is.na(GSE26712$time)]
> GSE26712 <- GSE26712[ match(names(coefs.probeset), featureNames(GSE26712)), ]
> GSE26712$surv <- Surv(GSE26712$time * 12, GSE26712$cens)
> dim(GSE26712)
```

```
Features  Samples 
       9      185 
```

```
> all.equal(featureNames(GSE26712), hernandez.sig$probeset)
```

```
[1] TRUE
```

```
> bonome08.data.geo <- t(exprs(GSE26712))
> bonome08.data.geo <- sweep(bonome08.data.geo, 2, apply(bonome08.data.geo, 2, median))
> ##bonome08.data.geo <- scale(bonome08.data.geo)
> bonome08.scores.geo <- predict(model.official.probeset, newdata=bonome08.data.geo, type="lp")@lp
> plotKMStratifyBy(y=bonome08.surv,
+                   linearriskscore=bonome08.scores.gene, method="median", inverse.HR=TRUE)
```



No. At Risk						
High Risk	92	57	23	13	8	3
Low Risk	93	55	26	10	3	2

Figure 29: Validation of the 9-gene expression signature on the Bonome 2008 dataset (N=185), using the gene symbol-defined signature and curatedOvarianData package. High and low risk-groups are reversed compared to Figure 3C.

```
> plotKMStratifyBy(y=bonome08.surv,
+                  linearriskscore=bonome08.scores.probeset,
+                  method="median", inverse.HR=TRUE)
```

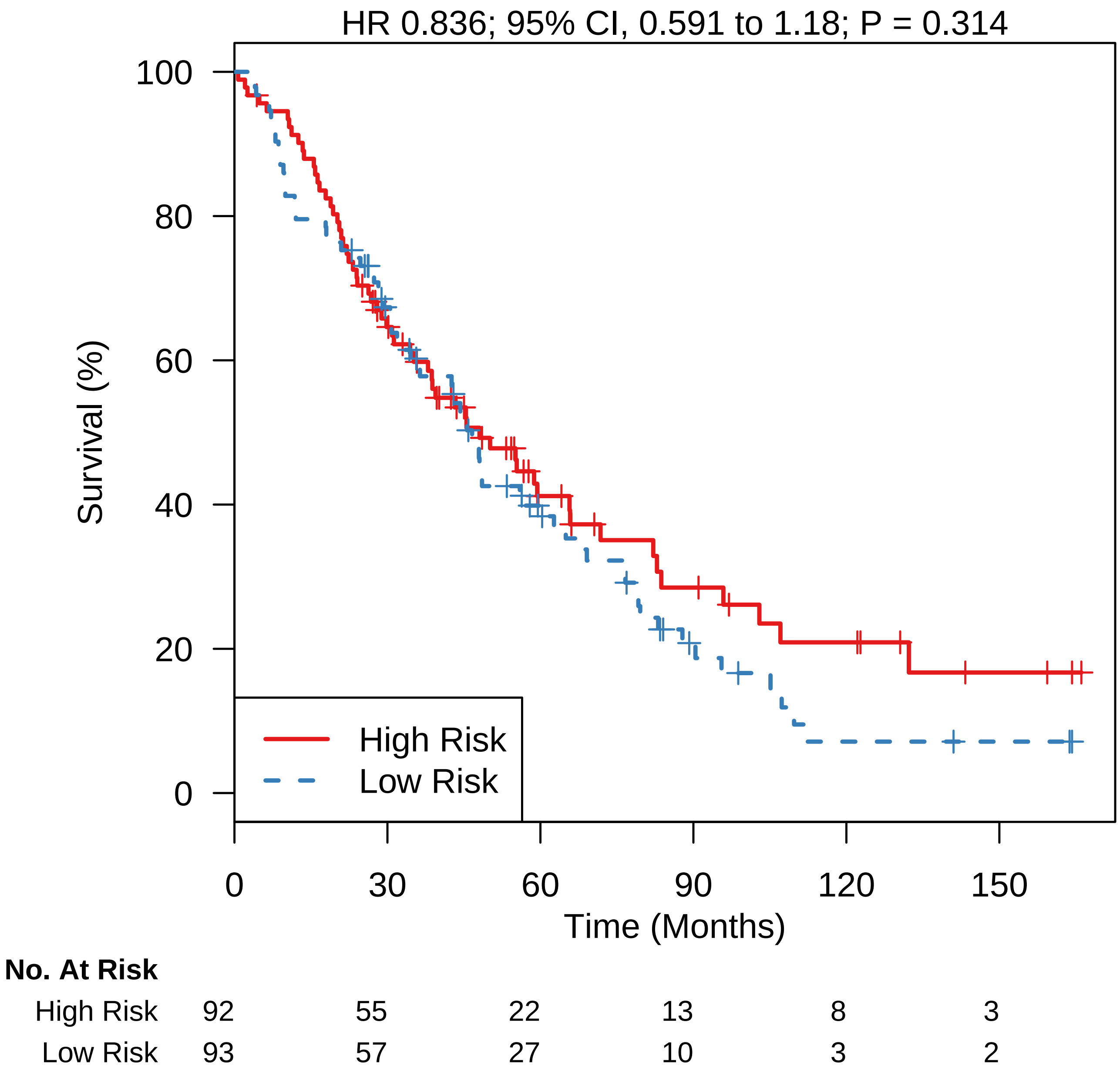
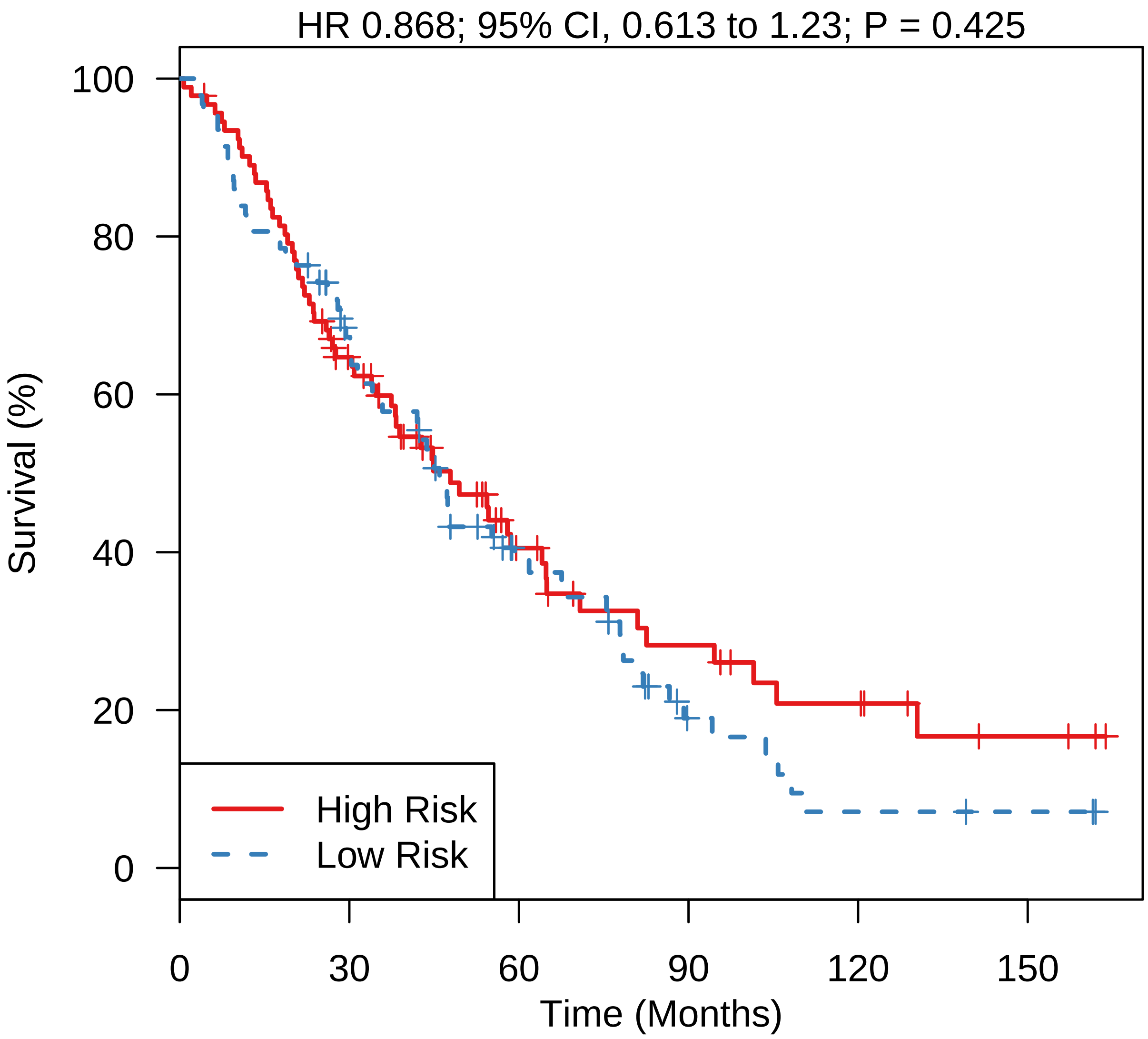


Figure 30: Validation of the 9-gene expression signature on the Bonome 2008 dataset (N=185), using the probesets specified by the authors. Again validation is not seen. Compare to Figure 3C.

```
> plotKMStratifyBy(y=GSE26712$surv,
+                   linearriskscore=bonome08.scores.geo,
+                   method="median", inverse.HR=TRUE)
```



No. At Risk

High Risk	92	54	22	13	8	3
Low Risk	93	57	25	8	3	2

Figure 31: Validation of the 9-gene expression signature on the Bonome 2008 dataset (N=185), using the probesets specified by the authors and the public dataset as downloaded from GEO. Again poor validation is seen. Compare to Figure 3C.

Finally, we found that we could most closely reproduce Figure 3C of the paper by using a -1 coefficient for the anti-correlated PTGS1 gene, z-scaling expression of each gene, and using the curatedOvarianData package:

```
> coefs.gene.ptgs1 <- hernandez.sig$coefficient
> names(coefs.gene.ptgs1) <- hernandez.sig$gene
> model.official.gene.ptgs1 <- new("ModelLinear",
+                                  coefficients=coefs.gene.ptgs1, modeltype="plusminus")

> data(GSE26712_eset, package="curatedOvarianData")
> GSE26712_eset <- GSE26712_eset[ ,!is.na(GSE26712_eset$days_to_death)]
> GSE26712_eset <- GSE26712_eset[ match(names(coefs.gene.ptgs1), featureNames(GSE26712_eset)), ]
> dim(GSE26712_eset)

Features  Samples
       9      185

> all.equal(featureNames(GSE26712_eset), hernandez.sig$gene)

[1] TRUE

> bonome08.data.gene <- t(exprs(GSE26712_eset))
> bonome08.data.gene <- scale(bonome08.data.gene)  #z-score scale data
> bonome08.scores.gene.ptgs1 <- predict(model.official.gene.ptgs1,
+                                       newdata=bonome08.data.gene, type="lp")@lp
```

This model seems to reproduce results from the paper most closely, so we use this version to define the official model:

```
> model.official <- model.official.gene.ptgs1
> save(model.official, file=model_file)
> plotKMStratifyBy(y=bonome08.surv,
+                  linearriskscore=bonome08.scores.gene.ptgs1, method="median", inverse.HR=TRUE)
```

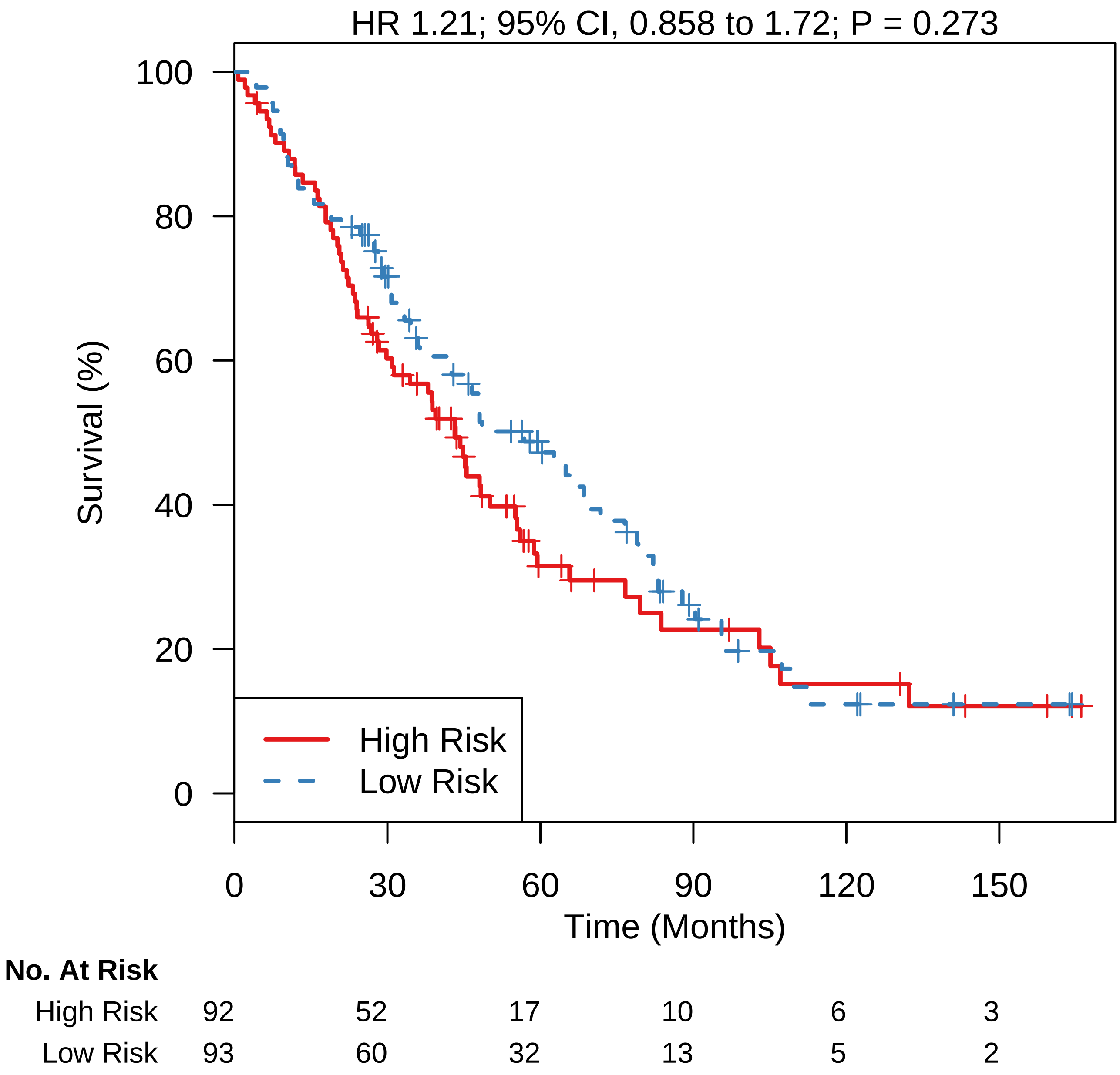


Figure 32: Validation of the 9-gene expression signature on the Bonome 2008 dataset (N=185), using the methods we found to produce results the most similar to Figure 3C. PTGS1 is assigned a negative coefficient so the negative of its value is used in the average, and expression of each gene is scaled to mean zero and unit variance. Optimal probesets selected in curatedOvarianData are used.

Mok 2009

Implemented by Jie Ding.

Mok Samuel, Bonome Tomas, Vathipadiekal Vinod, Bell Aaron, Johnson Michael, Wong Kwong-kwok, Park Dong-Choon, Hao Ke, Yip Daniel, Donninger Howard, Ozbun Laurent, Samimi Goli, Brady John, Randonovich Mike, Pise-Masison Cindy, Barrett J, Wong Wing, Welch William, Berkowitz Ross and Birrer Michael. *A gene signature predictive for outcome in advanced ovarian cancer identifies a survival factor: microfibril-associated glycoprotein 2.* Cancer cell 16:6(521-553).

Coefficients of the linear risk score are not provided, but original data and methods are reproducible. We use the authors' original data by GEO Accession ID GSE18520 with Affymetrix probeset identifiers, as well as the curatedOvarianData curated version with probesets mapped to gene symbols to generate coefficients of the risk score.

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/mmc2.xls"
[2] "19962670-TableS2.RData"
```

Load required libraries:

```
> library(Biobase)
> library(GEOquery)
> library(survival)
> library(curatedOvarianData)
> library(survHD)
> library(devtools)
> library(gdata)
```

Obtain authors' original data and GEO platform mapping (GPL570 is hgu133plus2):

```
> set.seed(1)  #to make getGEO check/store data in the same place when re-running.
> Mok2009.ori.eset <- getGEO("GSE18520")[[1]]
> GPL570 <- getGEO("GPL570")
```

Take survival data from curatedOvarianData:

```
> data(GSE18520_eset)
> Mok2009.pdata <- pData(GSE18520_eset)
> Mok2009.pdata.t <- Mok2009.pdata[Mok2009.pdata$sample_type == "tumor",]
> Mok2009.survival <- Surv(time=Mok2009.pdata.t$days_to_death / 30,
+                          event=Mok2009.pdata.t$vital_status == "deceased")
```

Read signature file:

```
> Mok2009.sig <- read.xls(input_file,stringsAsFactors=FALSE, as.is=TRUE)
```

Map signature probe sets to gene symbols:

```
> GPL570.table <- Table(GPL570)
> Mok2009.sig.gene <- GPL570.table$"Gene Symbol"[match(Mok2009.sig[,1],
+                                                      GPL570.table[,1])]
> Mok2009.sig.gene <- as.character(Mok2009.sig.gene)
> Mok2009.sig.gene.unique <- unique(Mok2009.sig.gene[Mok2009.sig.gene != ""])
```

Use original data to fit coxph model:

```
> Mok2009.ori.exp <- log2(exprs(Mok2009.ori.eset))
> Mok2009.ori.exp <- Mok2009.ori.exp[,Mok2009.pdata$sample_type == "tumor"]
> Mok2009.ori.exp[Mok2009.ori.exp < 0] <- 0 ##remove negative values
> ##Remove genes with any NA (there is only one, 230189_x_at:
> Mok2009.ori.exp <- Mok2009.ori.exp[!apply(Mok2009.ori.exp, 1, function(x) any(is.na(x))), ]
> Mok2009.ori.sig.exp <- Mok2009.ori.exp[Mok2009.sig[,1],]
> Mok2009.ori.sig.pc <- prcomp(t(Mok2009.ori.sig.exp),scale.=TRUE)
> Mok2009.ori.sig.coxph <- coxph(Mok2009.survival~Mok2009.ori.sig.pc$x[,1:5])
> ##coefficients for scaled gene values:
> Mok2009.ori.sig.prob.coef.s <- drop(Mok2009.ori.sig.pc$rotation[,1:5]
+                                    %*% Mok2009.ori.sig.coxph$coefficients)
```

Tabulate according to gene symbols:

```
> Mok2009.ori.sig.coef <- xtabs(Mok2009.ori.sig.prob.coef.s ~ Mok2009.sig.gene)
> Mok2009.ori.sig.coef <- Mok2009.ori.sig.coef[names(Mok2009.ori.sig.coef)!=""]
> temp <- names(Mok2009.ori.sig.coef)
> Mok2009.ori.sig.coef <- as.vector(Mok2009.ori.sig.coef)
> names(Mok2009.ori.sig.coef) <- temp
```

Use curatedOvariandata to fit coxph model:

```
> Mok2009.c.exp <- exprs(GSE18520_eset)
> Mok2009.c.exp <- Mok2009.c.exp[,Mok2009.pdata$sample_type == "tumor"]
> Mok2009.c.sig.exp <- Mok2009.c.exp[rownames(Mok2009.c.exp) %in%
+                                    Mok2009.sig.gene.unique,]
> Mok2009.c.sig.pc <- prcomp(t(Mok2009.c.sig.exp),scale.=TRUE)
> Mok2009.c.sig.coxph <- coxph(Mok2009.survival~Mok2009.c.sig.pc$x[,1:5])
> Mok2009.c.sig.coef <- drop((Mok2009.c.sig.pc$rotation[,1:5] /
+                             Mok2009.c.sig.pc$scale)
+                            %*% Mok2009.c.sig.coxph$coefficients)
```

Create survHD model objects:

```
> model.official <- new("ModelLinear", coefficients=Mok2009.ori.sig.coef, modeltype="plusminus")
> model.cod <- new("ModelLinear", coefficients=Mok2009.c.sig.coef, modeltype="plusminus")
```

```
> getRisk <- function(i, surv.obj, exp.obj) {
+     rownames(exp.obj) <- make.names(rownames(exp.obj))
+     ##remove left-out sample:
+     Xlearn <- exp.obj[, -i]
+     Ylearn <- surv.obj[-i]
+     ##step 1: feature selection, take top 200 cox scores:
+     model.step1 = plusMinus(X=t(Xlearn), y=Ylearn, tuningpar="nfeatures", lambda=200)
+     keep.genes <- names(model.step1@coefficients)[ abs(model.step1@coefficients) > 0 ]
+     Xlearn.200genes <- Xlearn[keep.genes, ]
+     exp.200genes <- exp.obj[keep.genes, ]
+     ##step 2: principal components regression with PC1-5:
+     sig.pc <- prcomp(t(Xlearn.200genes),scale.=TRUE)
+     sig.coxph <- coxph(Ylearn~sig.pc$x[,1:5])
+     ##final step: extract coefficients:
+     sig.coef <- drop((sig.pc$rotation[,1:5] /
+                       sig.pc$scale)
+                      %*% sig.coxph$coefficients)
+     model <- new("ModelLinear", coefficients=sig.coef, modeltype="plusminus")
+     ##Make predictions:
+     ret = predict(model, newdata=t(exp.200genes), type="lp")
+     return( ret@lp[i] )
+ }
```

Leave-one-out cross-validated good/bad prognosis predictions, using original data:

```
> risk.probes <- sapply(1:ncol(Mok2009.ori.exp), getRisk,
+                       surv.obj=Mok2009.survival, exp.obj=Mok2009.ori.exp)
```

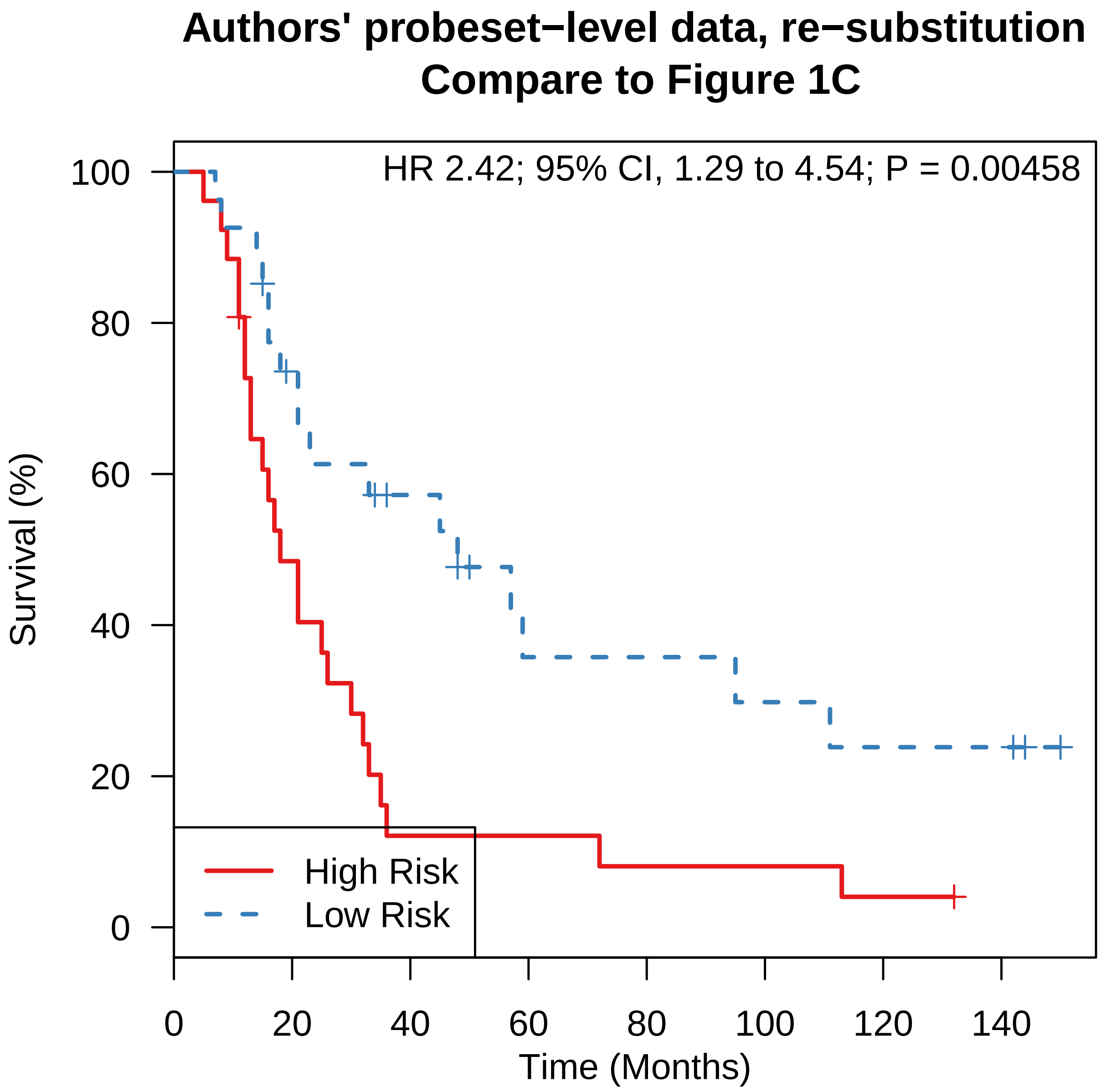
Using expression data summarized at the gene level from curatedOvarianData:

```
> risk.genes <- sapply(1:ncol(Mok2009.c.exp), getRisk,
+                       surv.obj=Mok2009.survival, exp.obj=Mok2009.c.exp)
```

Make Kaplan-Meier plots, stratifying at median risk score, to compare to Figure 1C.

Save the model (Official and COD version):

```
> save(model.official, model.cod, file=model_file)
> plotKMStratifyBy("median",
+                  y=Mok2009.survival,
+                  linearriskscore=predict(model.official, newdata=t(Mok2009.c.exp), type="lp")@lp,
+                  main="Authors' probeset-level data, re-substitution\n Compare to Figure 1C")
```



No. At Risk								
High Risk	26	12	3	3	2	2	1	0
Low Risk	27	18	12	6	6	5	4	4

Figure 33: Kaplan-Meier plot using model trained on GEO data, with re-substitution predictions (not cross-validated) produces very similar results to Figure 1C (compare to reported P = 0.0029).

```
> plotKMStratifyBy("median", y=Mok2009.survival, linearriskscore=risk.probes,
+                    main="Authors' probeset-level data, cross-validation\n
+                          Compare to Figure 1C")
```

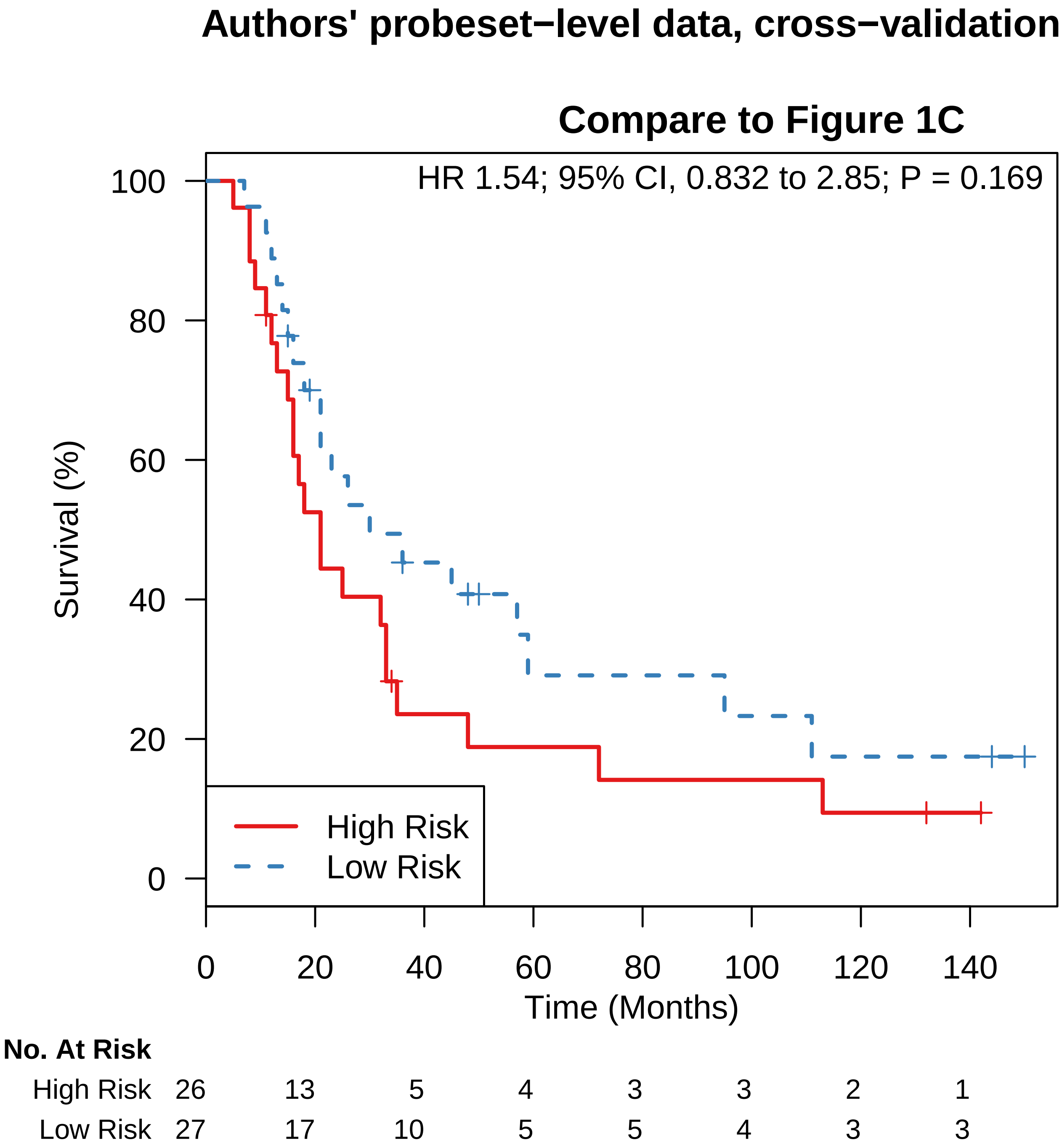


Figure 34: Kaplan-Meier plot using model trained on GEO data, with cross-validated predictions produces somewhat lower accuracy and higher p-value than Figure 1C (compare to reported P = 0.0029).

Crijns 2009

Crijns, A. P., R. S. Fehrmann, et al. (2009). ”Survival-related profile, pathways, and transcription factors in ovarian cancer.” *PLoS Medicine* 6(2): e24.

Implemented by Ben Haibe-Kains and Levi Waldron.

This is only an approximate reproduction of the model, since risk score coefficients are not available, and univariate Cox coefficients reported in Table 2 can only be approximately reproduced - with high correlation, but lower magnitude.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/GSE13876_Crijns09_table2.txt"
[2] "19192944-table2.RData"
```

Load required libraries:

```
> library(survHD)
> library(curatedOvarianData)
> library(HGNChelper)
> library(affy)
```

The *curatedOvarianData* version of the GSE13876 series has measurements from replicate arrays avaraged, although the paper does not say explicitly how replicates were handled. There are 415 microarrays from 157 patients in total.

```
> data(GSE13876_eset)
> GSE13876_eset$y <- Surv(GSE13876_eset$days_to_death / 30,
+                         GSE13876_eset$vital_status == "deceased")
```

Note that in the curatedOvarianData package, probes for this non-commercial Operon array were mapped to genes using a Blast to the human genome, and not all of this 86-gene signature could be mapped.

```
> source.data <- read.delim(input_file, as.is=TRUE)
> summary(rownames(source.data) %in% featureNames(GSE13876_eset))
```

```
   Mode   FALSE    TRUE    NA's
logical      10      76       0
```

```
> rownames(source.data)[!rownames(source.data) %in% featureNames(GSE13876_eset)]
```

```
 [1] "AGPAT7"    "C10orf80"  "C14orf121" "C1orf151"  "C20orf32"  "LIN28"
 [7] "OR9G9"     "TMEM150"   "TUBB4"     "ZBTB8"
```

We do much better after mapping unofficial symbols to approved HGNC symbols:

```
> fix.symbols <- checkGeneSymbols(rownames(source.data))
> fix.symbols[!fix.symbols$Approved, ]
```

```
           x Approved Suggested.Symbol
3     AGPAT7    FALSE           LPCAT4
8   C10orf80    FALSE          CCDC147
9  C14orf121    FALSE          LRRC16B
10  C1orf151    FALSE           MINOS1
15  C20orf32    FALSE            CASS4
42     LIN28    FALSE           LIN28A
76   TMEM150    FALSE         TMEM150A
80     TUBB4    FALSE           TUBB4A
83     ZBTB8    FALSE           ZBTB8A
```

```
> summary(fix.symbols$Suggested.Symbol %in% featureNames(GSE13876_eset))
```

```
   Mode   FALSE    TRUE    NA's
logical       1      85       0
```

```
> rownames(source.data) <- fix.symbols$Suggested.Symbol
```

Then keep only those genes which were mapped:

```
> source.data.matched <- source.data[rownames(source.data) %in% featureNames(GSE13876_eset), ]
```

Create an ExpressionSet with features matched to the paper's Table 2:

```
> eset.matched <- GSE13876_eset[match(rownames(source.data.matched),
+                                    featureNames(GSE13876_eset)), ]
> all.equal(featureNames(eset.matched), rownames(source.data.matched))
```

```
[1] TRUE
```

Check Cox coefficients against those reported in the table. The slope is different than one, likely due to possible scaling in the authors' analysis, but they are highly correlated:

As a double-check that our survival times match the paper, Table 1 reports the Median survival time as 21 months, and range as 1-234:

```
> median(GSE13876_eset$days_to_death) / 30
```

```
[1] 21
```

```
> range(GSE13876_eset$days_to_death) / 30
```

```
[1]   1 234
```

As stated in the Statistical Methods section, the first five PCs were used as predictor variables in a Cox proportional hazards model. We see only PC2 having a significant association with survival, but still use all five for consistency with the methods of the paper:

```
> coxresults <- rowCoxTests(exprs(eset.matched), eset.matched$y)
> plot(coxresults$coef ~ log(source.data.matched$Hazard.Ratio),
+      ylab="Univariate Cox regression coefficients fitted from curatedOvarianData",
+      xlab="Univariate Cox regression coefficients reported by Crijns et al.",
+      asp=1)
> legend("topleft", bty='n',
+        legend=paste("Pearson correlation =",
+        round(cor(coxresults$coef, log(source.data.matched$Hazard.Ratio)), 2)))
> abline(h=0, lty=2)
> abline(v=0, lty=2)
```

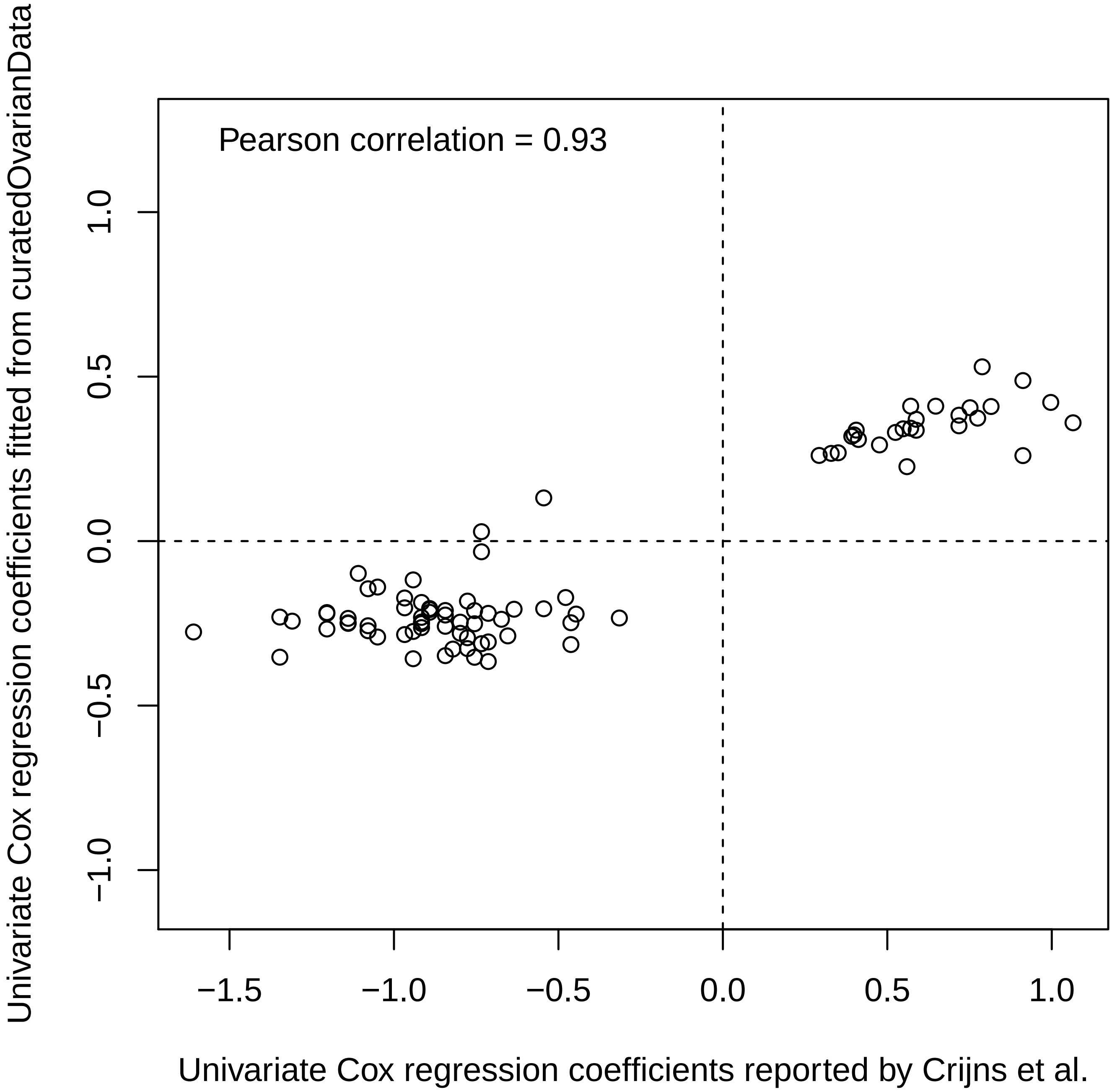


Figure 35: Univariate Cox coefficients are highly correlated with those reported in Table 2, although consistently smaller, possibly due to scaling done in the original analysis.

```
> pca.obj <- prcomp(t(exprs(eset.matched)))
> expr.dat <- data.frame(pca.obj$x[ ,1:5])
> expr.dat$y <- eset.matched$y
> cox.obj <- coxph(y ~ ., data=expr.dat)
> summary(cox.obj)
```

```
Call:
coxph(formula = y ~ ., data = expr.dat)

  n= 157, number of events= 113

         coef exp(coef)  se(coef)      z Pr(>|z|)
PC1 -0.004443  0.995567  0.018972 -0.234    0.815
PC2  0.365763  1.441614  0.049203  7.434 1.06e-13 ***
PC3  0.013382  1.013472  0.060881  0.220    0.826
PC4  0.033188  1.033745  0.066824  0.497    0.619
PC5  0.114152  1.120923  0.089708  1.272    0.203
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

    exp(coef) exp(-coef) lower .95 upper .95
PC1    0.9956     1.0045    0.9592     1.033
PC2    1.4416     0.6937    1.3091     1.588
PC3    1.0135     0.9867    0.8995     1.142
PC4    1.0337     0.9674    0.9068     1.178
PC5    1.1209     0.8921    0.9402     1.336

Concordance= 0.697  (se = 0.031 )
Rsquare= 0.329   (max possible= 0.998 )
Likelihood ratio test= 62.64  on 5 df,   p=3.454e-12
Wald test            = 55.41  on 5 df,   p=1.075e-10
Score (logrank) test = 58.65  on 5 df,   p=2.314e-11
```

Convert this to a ModelLinear object as defined in the survHD package:

```
> centers <- rowMeans(exprs(eset.matched))
> coefs <- pca.obj$rotation[, 1:5] %*% cox.obj$coefficients
> intercept <- -centers %*% pca.obj$rotation[, 1:5] %*% cox.obj$coefficients
> coefs.plus.intercept <- c(intercept, coefs)
> names(coefs.plus.intercept) <- c("(Intercept)", rownames(coefs))
> model.official <- new("ModelLinear",
+                       coefficients=coefs.plus.intercept,
+                       modeltype="plusminus")
```

Confirm that the predictions from the ModelLinear object match those of the coxph object:

```
> preds <- predict(model.official, newdata=t(exprs(eset.matched)), type="lp")@lp
> all.equal(predict(cox.obj), preds)
```

```
[1] TRUE
```

Finally, make a model.cod model object, as an alternative model which uses the full expression dataset for gene selection and model fitting using the authors' methods. We select all genes with Wald Test p-values < 0.001 as stated in the Statistical Methods, using the five PC regression approach. First, we remove probesets matching to multiple genes:

```
> X <- exprs(GSE13876_eset)
> X <- X[!grepl("///", rownames(X)), ]
```

then perform univariate Cox tests. We see only five genes meeting the threshold p<0.001:

```
> cox.results <- rowCoxTests(X, GSE13876_eset$y)
> summary(cox.results$p.value < 0.001)

   Mode   FALSE    TRUE    NA's
logical   16757       5       0
```

so instead we keep the top 86 genes for consistency in length with the reported signature:

```
> X <- X[rank(cox.results$p.value) <= 86, ]
> dim(X)

[1]  86 157
```

Now do the principal components regression using PCs 1-5:

```
> pca.obj <- prcomp(t(X))
> expr.dat <- data.frame(pca.obj$x[ ,1:5])
> expr.dat$y <- eset.matched$y
```

Here we see PC1 strongly associated with survival:

```
> cox.obj <- coxph(y ~ ., data=expr.dat)
> summary(cox.obj)

Call:
coxph(formula = y ~ ., data = expr.dat)

  n= 157, number of events= 113

         coef exp(coef)  se(coef)      z Pr(>|z|)    
PC1  0.090058  1.094237  0.020580  4.376 1.21e-05 ***
PC2 -0.002734  0.997270  0.038481 -0.071   0.9434    
PC3 -0.074036  0.928638  0.040579 -1.824   0.0681 .  
PC4 -0.074329  0.928366  0.061312 -1.212   0.2254    
PC5 -0.059151  0.942564  0.055979 -1.057   0.2907    
---
Signif. codes:  0 âĂŸ***âĂŹ 0.001 âĂŸ**âĂŹ 0.01 âĂŸ*âĂŹ 0.05 âĂŸ.âĂŹ 0.1 âĂŸ âĂŹ 1

    exp(coef) exp(-coef) lower .95 upper .95
PC1    1.0942     0.9139    1.0510     1.139
PC2    0.9973     1.0027    0.9248     1.075
```

```
PC3    0.9286     1.0768    0.8576     1.006
PC4    0.9284     1.0772    0.8232     1.047
PC5    0.9426     1.0609    0.8446     1.052

Concordance= 0.642  (se = 0.031 )
Rsquare= 0.156   (max possible= 0.998 )
Likelihood ratio test= 26.56  on 5 df,   p=6.947e-05
Wald test            = 24.11  on 5 df,   p=0.000207
Score (logrank) test = 24.72  on 5 df,   p=0.0001581
```

Convert this to a ModelLinear object as defined in the survHD package:

```
> centers <- rowMeans(X)
> coefs <- pca.obj$rotation[, 1:5] %*% cox.obj$coefficients
> intercept <- -centers %*% pca.obj$rotation[, 1:5] %*% cox.obj$coefficients
> coefs.plus.intercept <- c(intercept, coefs)
> names(coefs.plus.intercept) <- c("(Intercept)", rownames(coefs))
> model.cod <- new("ModelLinear", coefficients=coefs.plus.intercept, modeltype="plusminus")
```

Look at the overlap of the genes found in each of these models:

```
> length(model.official@coefficients)

[1] 86

> length(model.cod@coefficients)

[1] 87

> length(intersect(names(model.cod@coefficients), names(model.official@coefficients)))

[1] 26
```

Finally, save these models:

```
> save(model.official, model.cod, file=model_file)
```

Denkert 2009

Denkert C, Budczies J, Darb-Esfahani S, Gyorffy B, Sehouli J, Konsgen D, Zeillinger R, Weichert W, Noske A, Buckendahl AC, Muller BM, Dietel M, Lage H. *A prognostic gene expression index in ovarian cancer - validation across different independent data sets.* J Pathol. 2009 Jun;218(2):273-80. PMID: 19294737

Implemented by Jie Ding.

Input/output arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/PATH_2547_sm_supportinginformationst2t.xls"
[2] "19294737-SuppTable2.RData"
```

Load required libraries:

```
> library(Biobase)
> library(GEOquery)
> library(gdata)
> library(survival)
> library(survHD)
> library(curatedOvarianData)
> library(affy)
```

The coefficients of this model are provided by the authors in Supporting Information Table 2, titled

`PATH_2547_sm_supportinginformationst2t.xls}`:

```
> Denkert2009.table <- read.xls(input_file, as.is=TRUE)
> head(Denkert2009.table)
```

```
     ProbeSet
1 213356_x_at
2 211755_s_at
3 207573_x_at
4   200657_at
5 207941_s_at
6   202077_at

1 HNRNPA1 /// HNRPA1L-2 /// HNRPA1P5 /// LOC100128836 /// LOC391670 /// LOC402112 /// LOC440125 /// LOC64281
2
3
4
5
6

1 heterogeneous nuclear ribonucleoprotein A1 /// heterogeneous nuclear ribonucleoprotein A1 pseudogene /// he
2
3
4
```

```
5
6
   GeneBank Mean_Expression P_Logrank Contribution_OPI
1  AL568186           15157    0.0025         0.157231
2  BC005960            7425    0.0190         0.111674
3 NM_006476            6045    0.0200         0.082747
4 NM_001152            5650    0.0032         0.063709
5 NM_004902            2974    0.0170         0.046795
6 NM_005003            3919    0.0060         0.042166
```

```
> dim(Denkert2009.table)
```

```
[1] 22283     7
```

Keep genes and weights:

```
> Denkert2009.table <- Denkert2009.table[,c(1,2,7)]
> names(Denkert2009.table)[2] <- "Gene"
> names(Denkert2009.table)[3] <- "OPI"
> Denkert2009.table <- Denkert2009.table[Denkert2009.table$OPI != 0, ]
> stopifnot(nrow(Denkert2009.table) == 300)   #300-gene signature
```

Re-map probesets to gene symbols:

```
> GPL96 <- getGEO("GPL96")
> GPL96.table <- Table(GPL96)
> Denkert2009.table$Gene <-
+   GPL96.table$"Gene Symbol"[match(Denkert2009.table$ProbeSet,GPL96.table$ID)]
> Denkert2009.table$Gene <- as.character(Denkert2009.table$Gene)
```

There is one gene appeared twice (CYP51A1). Add its two coefs together:

```
> temp <- xtabs(Denkert2009.table$OPI ~ Denkert2009.table$Gene)
> Denkert2009.sig <- data.frame(Gene=names(temp),OPI=as.numeric(temp))
```

And remove the coefficient for unmapped probesets:

```
> Denkert2009.sig <- Denkert2009.sig[Denkert2009.sig$Gene != "", ]
```

Now create a model with probeset coefficients, and a model with gene symbol coefficients which we will consider the "official model". Note that we need to use the *negative* of the coefficients in order to have high values correspond to high risk.

```
> ##probeset version
> coefs.probeset <- structure(Denkert2009.table$OPI, .Names=Denkert2009.table$ProbeSet)
> model.probeset <- new("ModelLinear", coefficients=-coefs.probeset, modeltype="plusminus")
> ##gene version
> coefs.genes <- structure(Denkert2009.sig$OPI, .Names=as.character(Denkert2009.sig$Gene))
> model.official <- new("ModelLinear", coefficients=-coefs.genes, modeltype="plusminus")
```

Obtain data from GEO:

```
> Denkert2009.eset <- getGEO("GSE14764")
> Denkert2009.exp <- exprs(Denkert2009.eset[[1]])
```

Also obtain data from curatedOvarianData package:

```
> data("GSE14764_eset", package="curatedOvarianData")
> GSE14764.exp <- exprs(GSE14764_eset)
```

Note that this linear risk score is developed for unlogged expression values, so we transform them:

```
> Denkert2009.exp <- 2^Denkert2009.exp
> GSE14764.exp <- 2^GSE14764.exp
```

Calculate scores, first using the provided probeset weights on the probeset expression matrix from GEO

```
> Denkert2009.score <- predict(model.probeset, newdata=t(Denkert2009.exp), type="lp")@lp
```

Then using the gene symbol model:

```
> GSE14764.score <- predict(model.official, newdata=t(GSE14764.exp), type="lp")@lp
```

We additionally validate on the Duke study for comparison to Figure 3A, using the gene symbol model and curatedOvarianData package.

Finally, save the model:

```
> save(model.official, file=model_file)
> GEO.time <- as.numeric(sub("overall survival time: ", "", Denkert2009.eset[[1]]$characteristics_ch1.5))
> GEO.cens <- as.numeric(sub("overall survival event: ", "", Denkert2009.eset[[1]]$characteristics_ch1.6))
> GEO.surv <- Surv(time=GEO.time, event=GEO.cens)
> plotKMStratifyBy("median",
+                  y=GEO.surv,
+                  linearriskscore=Denkert2009.score)
```

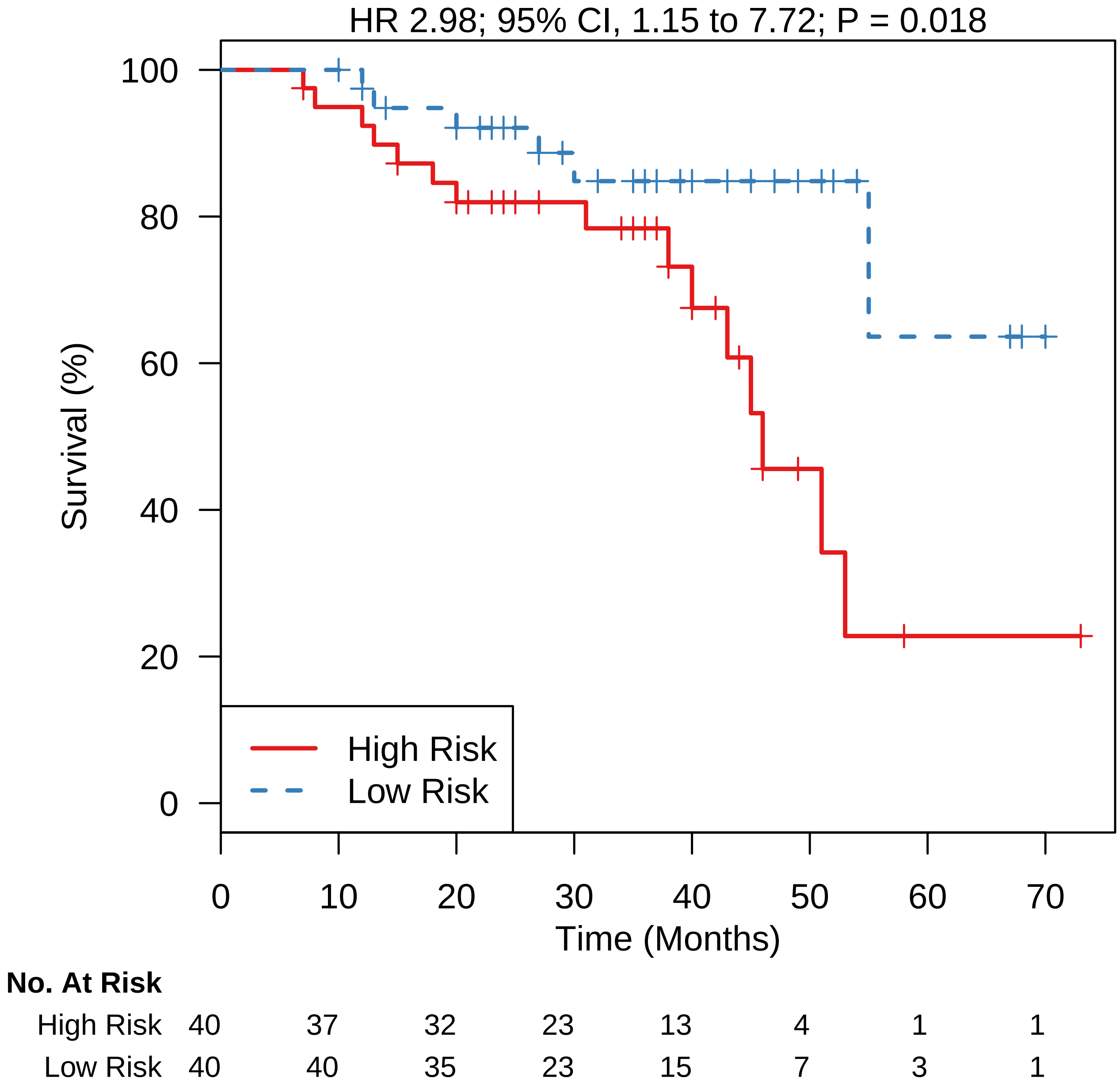
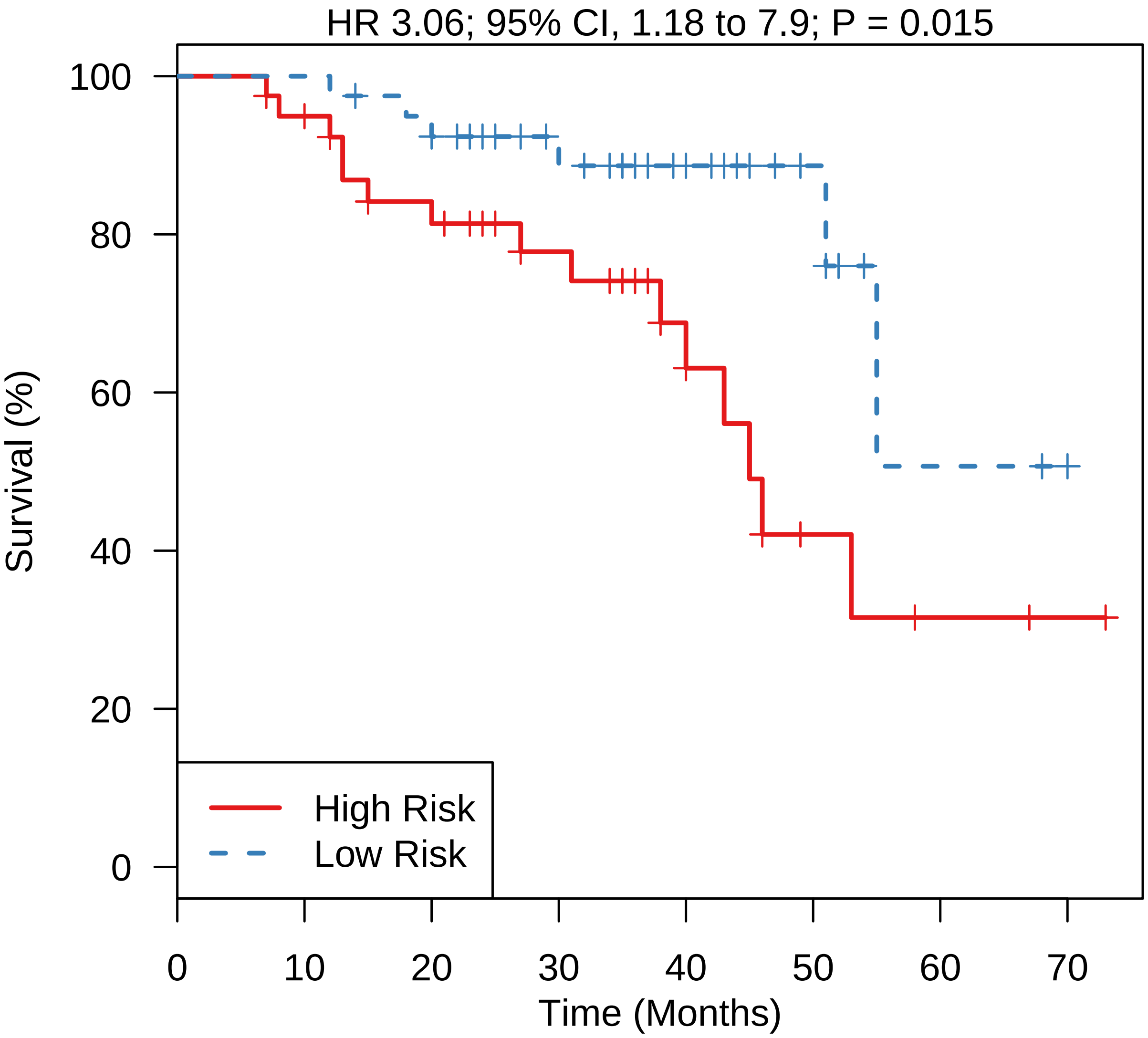


Figure 36: Probe set version of model, using author's data from GEO. Nearly identical to Figure 2A, which reports p=0.019.

```
> COD.surv <- Surv(time=GSE14764_eset$days_to_death / 30, event=GSE14764_eset$vital_status=="deceased")
> plotKMStratifyBy("median",
+                  y=COD.surv,
+                  linearriskscore=GSE14764.score)
```



No. At Risk

High Risk	40	37	30	21	12	4	2	1
Low Risk	40	40	37	25	16	7	2	1

Figure 37: Using a gene symbol version of the model, and data from curatedOvarianData package. Nearly identical to Figure 2A, which reports p=0.019.

```
> data("PMID17290060_eset", package="curatedOvarianData")
> PMID17290060.score <- predict(model.official, newdata=2^t(exprs((PMID17290060_eset))), type="lp")@lp
> PMID17290060.surv <- Surv(time=PMID17290060_eset$days_to_death / 30, event=PMID17290060_eset$vital_status==
> plotKMStratifyBy("median",
+                   y=PMID17290060.surv,
+                   linearriskscore=PMID17290060.score)
```

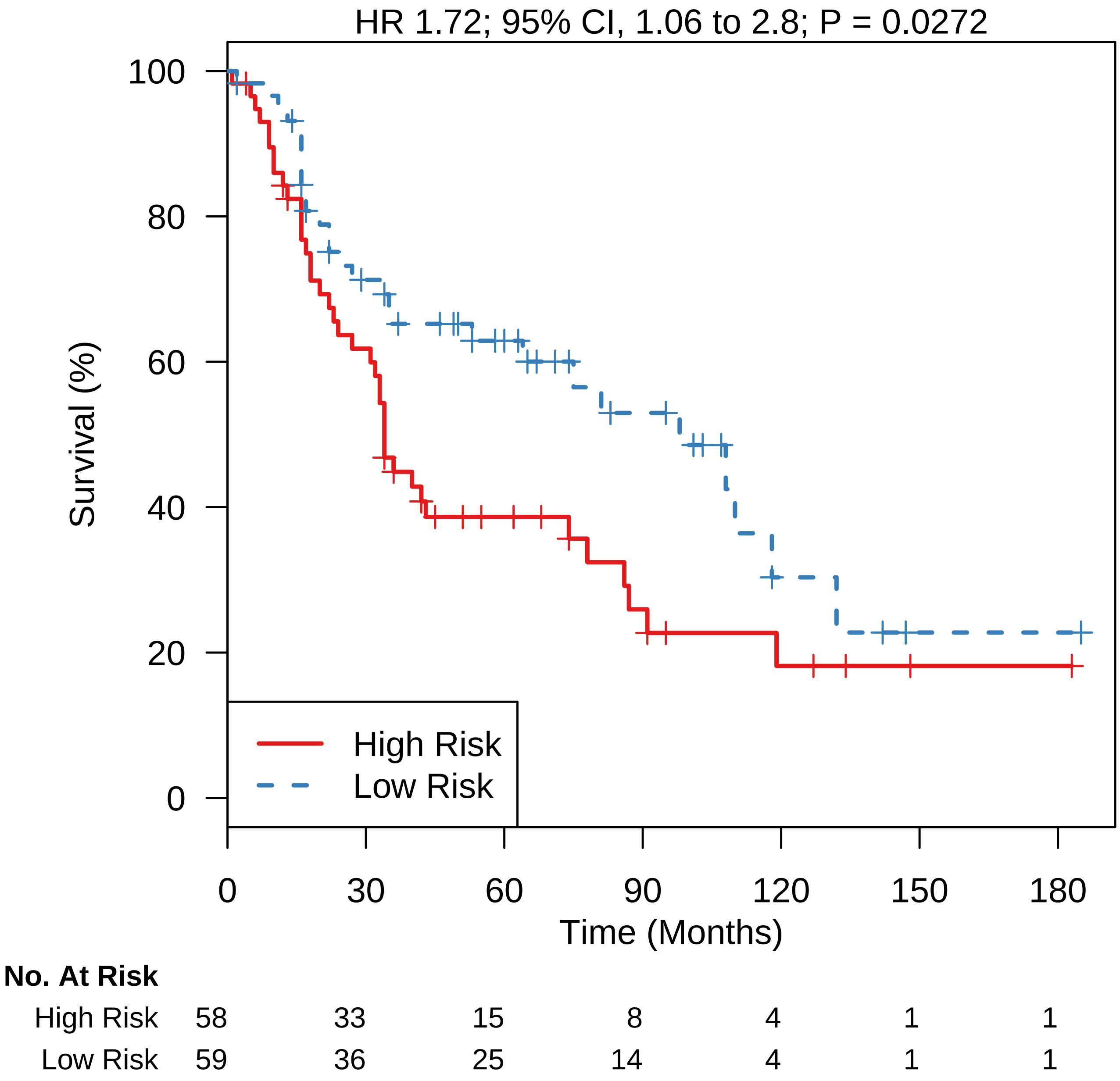


Figure 38: Independent validation on the Dressman (Duke) dataset, using curatedOvarianData package. Nearly identical to Figure 3A, which reports p=0.021.

Bonome 2008, optimally debulked patients

Bonome T, Levine DA, Shih J, Randonovich M, Pise-Masison CA, Bogomolniy F, Ozbun L, Brady J, Barrett JC, Boyd J, Birrer MJ: *A Gene Signature Predicting for Survival in Suboptimally Debulked Patients with Ovarian Cancer.* Cancer Res 2008, 68:5478-5486.

Implemented by Markus Riester and Levi Waldron.

This paper reports two signatures: for optimally and for suboptimally debulked patients. In this vignette we reproduce the signature for optimally debulked patients, although the paper reports this signature to be of marginal predictive quality.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/PMID18593951-TableS2-optimal.txt"
[2] "18593951-TableS2.RData"
```

Load required libraries:

```
> library(survHD)
> library(curatedOvarianData)
> library(affy)
> library(devtools)
> library(GEOquery)
> library(hgu133a.db)
> library(annotate)
```

We will consider three versions of the dataset. First, curatedOvarianData with HGNC symbols as features. We substitute "-" with "hyphen" in HGNC symbols so they are valid R names:

```
> data(GSE26712_eset, package="curatedOvarianData")
> eset.genes <- GSE26712_eset
```

FULLVcuratedOvarianData version, which uses original probe set identifiers, and is pre-processed using frozen RMA:

```
> data(GSE26712_eset, package="FULLVcuratedOvarianData")
> eset.probes <- GSE26712_eset
```

And using the authors' own version from GEO, processed by RMA:

```
> set.seed(1)
> eset.geo <- getGEO("GSE26712")[[1]]
```

Use optimally debulked patients only, and make sure these are the same sample IDs in each dataset:

```
> eset.genes <- eset.genes[ ,!is.na(eset.genes$debulking)]
> eset.genes <- eset.genes[ ,eset.genes$debulking == "optimal"]
> eset.probes <- eset.probes[ ,!is.na(eset.probes$debulking)]
> eset.probes <- eset.probes[ ,eset.probes$debulking == "optimal"]
> eset.geo <- eset.geo[, eset.geo$source_name_ch1 == "Ovarian tumor"]
> eset.geo <- eset.geo[, grep("Optimal", eset.geo$characteristics_ch1.1)]
> identical( sampleNames(eset.geo), sampleNames(eset.probes) )
```

```
[1] TRUE
```

```
> identical( sampleNames(eset.geo), sampleNames(eset.genes) )
```

```
[1] TRUE
```

Prepare the Surv objects (overall survival) for each ExpressionSet:

```
> eset.genes$y <- Surv(eset.genes$days_to_death / 30, eset.genes$vital_status == "deceased")
> eset.probes$y <- Surv(eset.probes$days_to_death / 30, eset.probes$vital_status == "deceased")
> table( eset.geo$characteristics_ch1.2 )  ##all deaths are labelled DOD
```

```
                                         status: AWD (alive with disease) 
                                   0                                   16 
       status: DOD (dead of disease) status: NED (no evidence of disease) 
                                  51                                   23 
```

```
> eset.geo$y <- Surv(as.numeric(sub("survival years: ", "", eset.geo$characteristics_ch1.3)) * 365,
+                     eset.geo$characteristics_ch1.2 == "status: DOD (dead of disease)")
```

Load the Supplemental Table S2 for optimally debulked patients:

```
> signature.table <- read.delim(input_file, as.is=TRUE)
> nrow(signature.table)
```

```
[1] 263
```

```
> geneset.probes <- signature.table$Probe.Set
```

Use current Bioconductor mapping to gene symbols:

```
> geneset.genes <- getSYMBOL(geneset.probes, data="hgu133a.db")
> geneset.genes[is.na(geneset.genes)]  ##probesets that could not be mapped
```

```
  219156_at   222303_at   214967_at   221939_at   210365_at 210676_x_at 
         NA          NA          NA          NA          NA          NA 
209919_x_at   215183_at 217191_x_at   209905_at 
         NA          NA          NA          NA 
```

All of the probesets in this table should be found in the full versions of the ExpressionSets:

```
> summary(geneset.probes %in% featureNames(eset.probes))
```

```
   Mode    TRUE    NA's 
logical     263       0 
```

```
> summary(geneset.probes %in% featureNames(eset.geo))
```

```
   Mode    TRUE    NA's
logical     263       0
```

curatedOvarianData uses mappings from biomaRt, and some additional probesets are lost in the curatedOvarianData ExpressionSet. Those shown below as NA are missing in both biomaRt and Bioconductor, those showing gene symbols are present in Bioconductor but not Biomart:

```
> summary(geneset.genes %in% featureNames(eset.genes))
```

```
   Mode   FALSE    TRUE    NA's
logical      32     231       0
```

```
> geneset.genes[!geneset.genes %in% featureNames(eset.genes)]
```

```
219972_s_at   205380_at   208659_at  39817_s_at   218117_at   201319_at
   "PCNXL4"     "PDZK1"     "CLIC1"     "DNPH1"      "RBX1"    "MYL12A"
  219156_at 204238_s_at 211325_x_at   215513_at 221613_s_at    36019_at
         NA     "DNPH1"    "DSTNP2"     "HYMAI"    "ZFAND6"     "STK19"
  222303_at 209569_x_at   214967_at   207219_at   210496_at   221939_at
         NA      "NSG1"          NA    "ZFP69B"     "NAG18"          NA
  210365_at   212056_at   215307_at 207134_x_at 210676_x_at   209111_at
         NA      "GSE1"    "ZNF529"     "TPSB2"          NA      "RNF5"
  218338_at 209919_x_at   215183_at 217191_x_at   201526_at   217592_at
     "PHC1"          NA          NA          NA      "ARF5"    "ZSWIM1"
  209905_at   220173_at
         NA   "CCDC176"
```

```
> ##keep the ones present in the data:
> geneset.genes <- geneset.genes[ geneset.genes %in% featureNames(eset.genes) ]
```

This reported model uses the univariate Cox regression coefficients as weights for the linear risk score, so can we reproduce those given in Supplemental Table S2?

Now we reproduce Figure 2A using each version of the data. Define a function to do the leave-one-out as described. In each iteration select genes whose univariate Cox regression Wald Test p-value is less than 0.01, and define a linear risk score whose coefficients are equal to the univariate Cox regression coefficient. Call the left-out patient “Poor prognosis” if its prediction is above the median risk score. Then use this to calculate patient predictions for each of the three datasets:

```
> getRisk <- function(i, eset) {
+     featureNames(eset) <- make.names(featureNames(eset))
+     Xlearn <- t(exprs(eset))[-i, ]
+     Ylearn <- eset$y[-i]
+     model = plusMinusSurv(Xlearn, Ylearn,
+                           modeltype="compoundcovariate",
+                           tuningpar="pval", lambda=0.01)
+     ret = predict(model, newdata=t(exprs(eset)), type="lp")
+     ifelse(ret@lp[i] > median(ret@lp), "Poor prognosis", "Good prognosis")
+ }
> risk.genes <- sapply(1:ncol(eset.genes), getRisk, eset.genes)
> risk.probes <- sapply(1:ncol(eset.probes), getRisk, eset.probes)
> risk.geo <- sapply(1:ncol(eset.geo), getRisk, eset.geo)
> rowcox.probes <- rowCoxTests(X=exprs(eset.probes[geneset.probes,]), y=eset.probes$y)
> rowcox.geo <- rowCoxTests(X=exprs(eset.geo[geneset.probes,]), y=eset.geo$y)
> coxc <- data.frame(published=log(signature.table$Cox.Hazard.Ratio),
+                    GEO=rowcox.geo$coef,
+                    curatedOvarianData=rowcox.probes$coef,
+                    row.names=rownames(rowcox.probes))
> pairs(coxc)
```

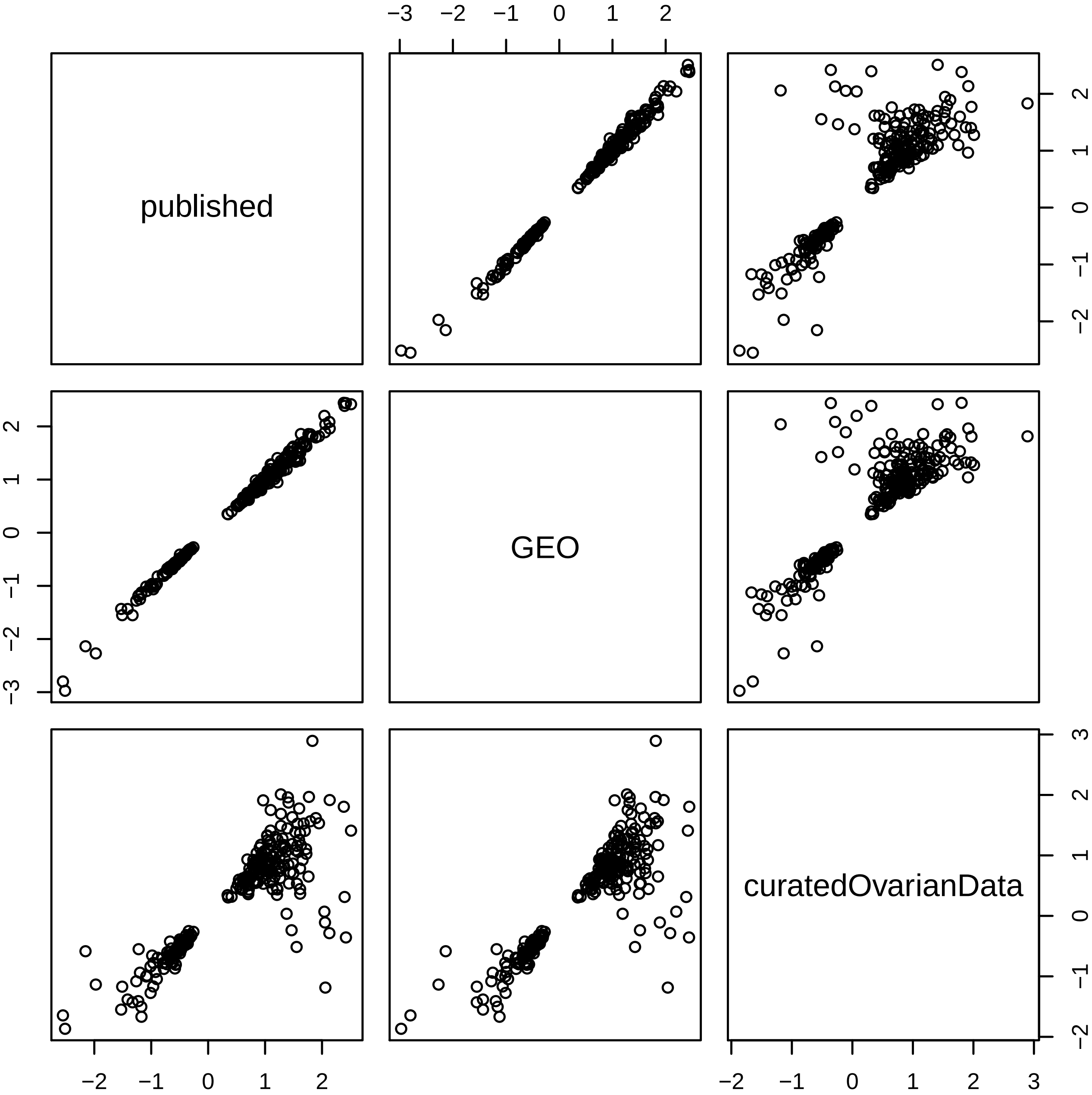
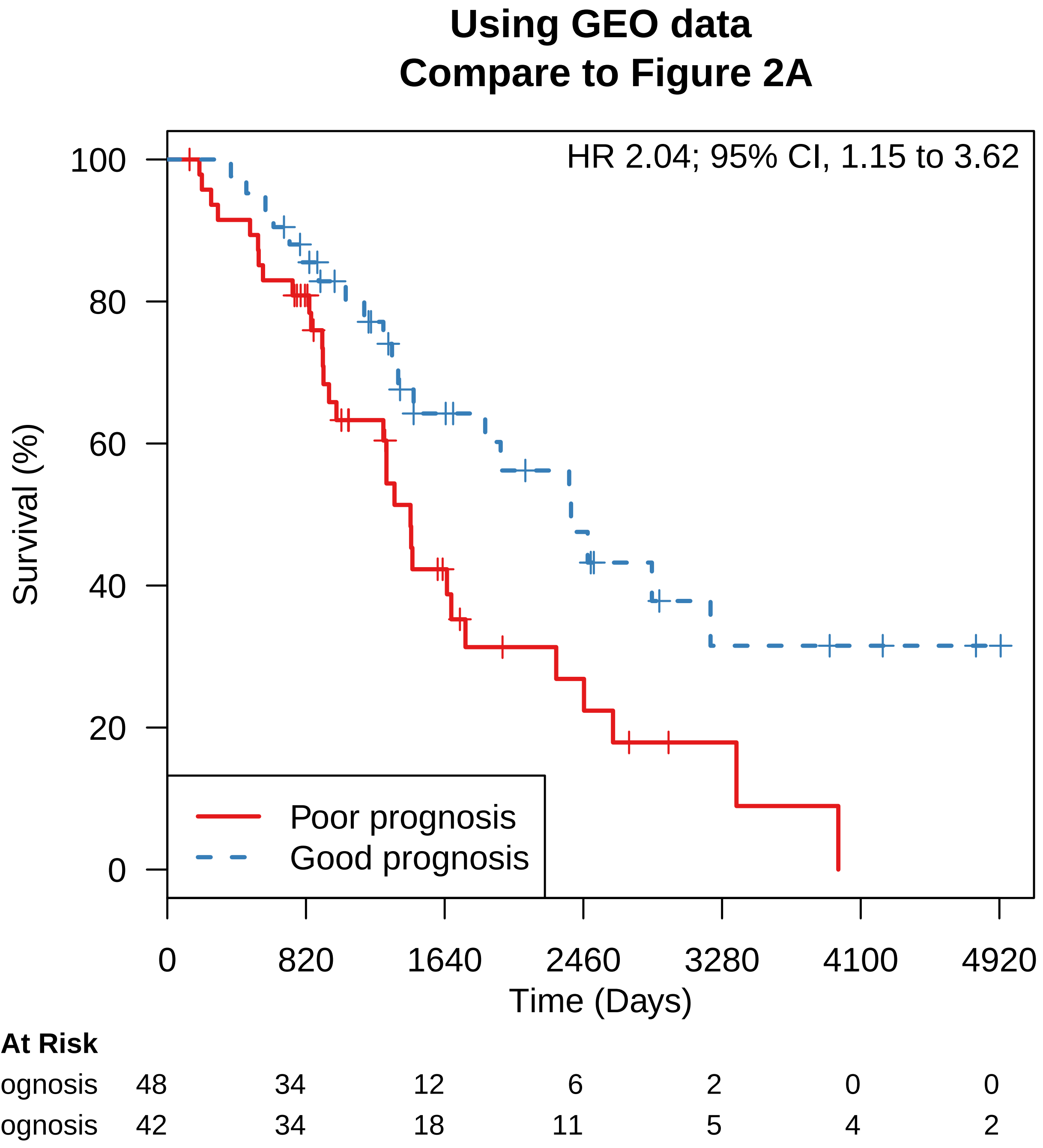


Figure 39: Cox coefficients: as published in Supplemental Table S2, calculated using the GEO dataset, and calculated using the FULLVcuratedOvarianData package.

```
> ##plotKM is from the survHD package
> plotKM(y=eset.geo$y,
+        strata=factor(risk.geo, levels=c("Poor prognosis", "Good prognosis")),
+        show.PV=FALSE,
+        main="Using GEO data\n Compare to Figure 2A")
```



No. At Risk							
Poor prognosis	48	34	12	6	2	0	0
Good prognosis	42	34	18	11	5	4	2

Figure 40: Using the authors' own expression data from GEO looks exactly like Figure 2A, although the numbers of patients in high and low risk groups do not match the legend in the paper.

```
> plotKM(y=eset.probes$y,
+            strata=factor(risk.probes, levels=c("Poor prognosis", "Good prognosis")),
+            show.PV=FALSE,
+            main="Using FULLVcuratedOvarianData\n Compare to Figure 2A")
```

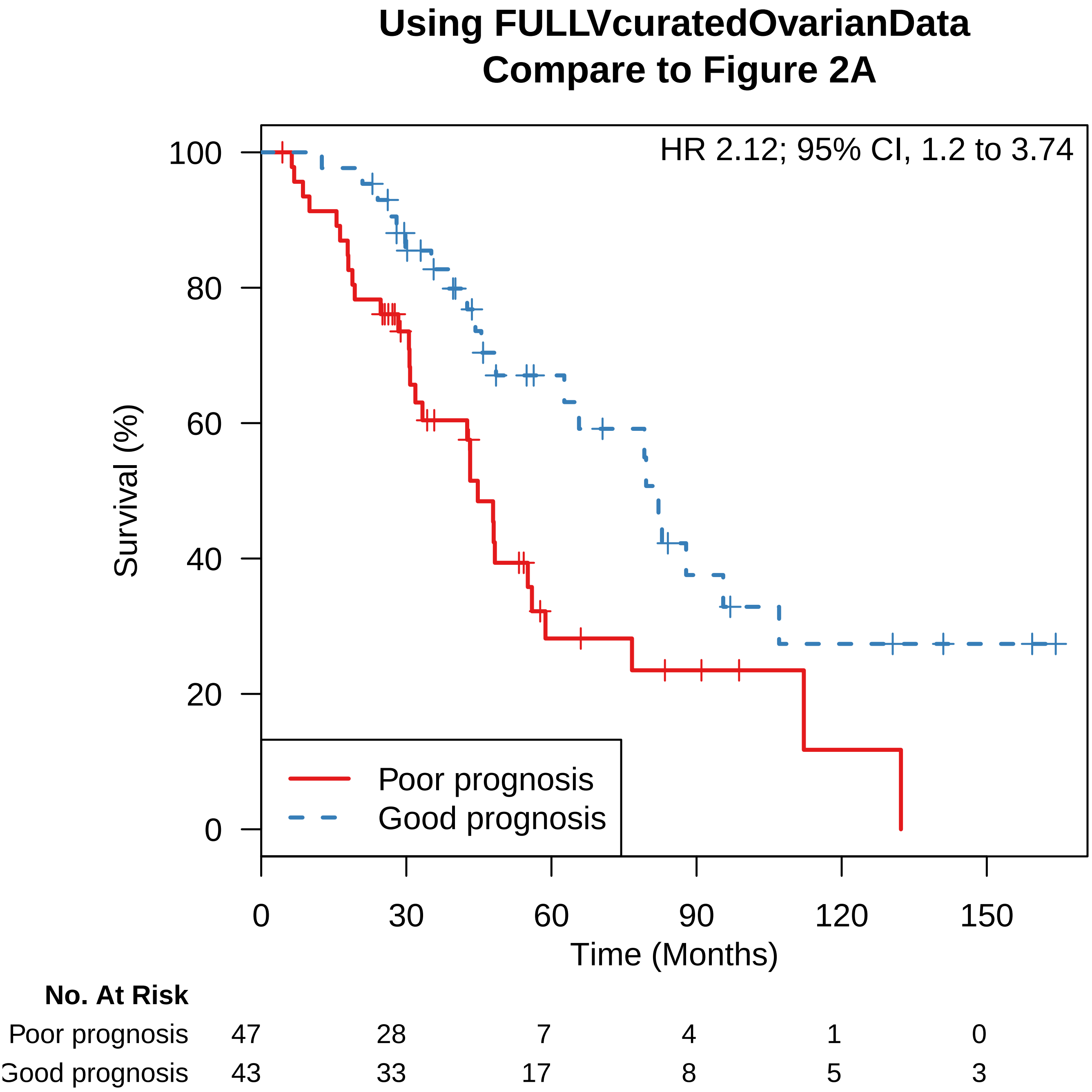


Figure 41: Using FULLVcuratedOvarianData (preprocessed by frozen RMA), it is still extremely close to Figure 2A.

```
> plotKM(y=eset.genes$y,
+            strata=factor(risk.genes, levels=c("Poor prognosis", "Good prognosis")),
+            show.PV=FALSE,
+            main="Using curatedOvarianData\n Compare to Figure 2A")
```

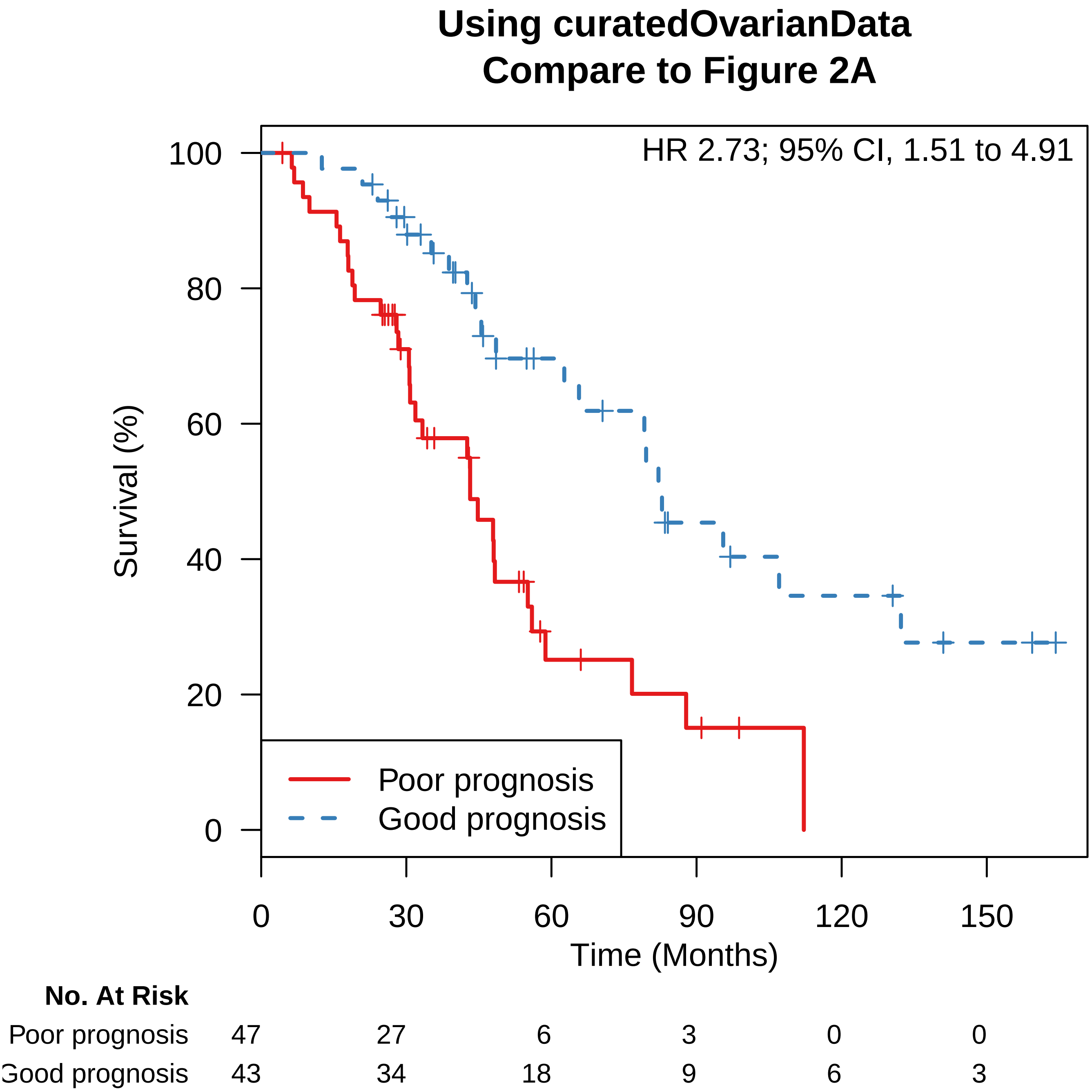


Figure 42: And using curatedOvarianData, where probesets are already collapsed to HGNC symbols.

Finally, we create the model as it will be used for meta-analysis. For the official model, we use the coefficients provided by the authors, map these to HGNC symbols, averaging coefficients for probesets which map to the same gene:

```
> coxc$hgnc <- getSYMBOL(geneset.probes, data="hgu133a.db")
> coefs <- aggregate(coxc$GEO, list(coxc$hgnc), mean)
> coefs <- structure(coefs[ ,2], .Names=coefs[ ,1])
> model.official <- new("ModelLinear", coefficients=coefs, modeltype="compoundcovariate")
```

We also build a model by reproducing the authors' methods, but using the already mapped curatedOvarianData package.

```
> Xlearn <- t(exprs(eset.genes))
> Xlearn <- Xlearn[ ,!grepl("///", colnames(Xlearn))]  ##remove probes which mapped to multiple genes
> Ylearn <- eset.genes$y
> model.cod = plusMinusSurv(Xlearn, Ylearn,
+                           modeltype="compoundcovariate",
+                           tuningpar="pval", lambda=0.01)@model
> ##prune zero coefficients:
> model.cod@coefficients <- model.cod@coefficients[abs(model.cod@coefficients) > 0]
```

Look at the overlap of the genes found in each of these models:

```
> length(model.official@coefficients)

[1] 241

> length(model.cod@coefficients)

[1] 187

> length(intersect(names(model.cod@coefficients), names(model.official@coefficients)))

[1] 94
```

Finally, save these models:

```
> save(model.official, model.cod, file=model_file)
```

Bonome 2008, suboptimally debulked patients

Bonome T, Levine DA, Shih J, Randonovich M, Pise-Masison CA, Bogomolniy F, Ozbun L, Brady J, Barrett JC, Boyd J, Birrer MJ: *A Gene Signature Predicting for Survival in Suboptimally Debulked Patients with Ovarian Cancer.* Cancer Res 2008, 68:5478-5486.

Implemented by Markus Riester and Levi Waldron.

This paper reports two signatures: for optimally and for suboptimally debulked patients. In this vignette we reproduce the signature for suboptimally debulked patients.

Input arguments:

```
> print(c(input_file, model_file))
```

```
[1] "../../input/official_models/PMID18593951-TableS3-suboptimal.txt"
[2] "18593951-TableS3.RData"
```

Load required libraries:

```
> library(survHD)
> library(curatedOvarianData)
> library(affy)
> library(devtools)
> library(GEOquery)
> library(hgu133a.db)
> library(annotate)
```

We will consider three versions of the dataset. First, curatedOvarianData with HGNC symbols as features. We substitute "-" with "hyphen" in HGNC symbols so they are valid R names:

```
> data(GSE26712_eset, package="curatedOvarianData")
> eset.genes <- GSE26712_eset
```

FULLVcuratedOvarianData version, which uses original probe set identifiers, and is pre-processed using frozen RMA:

```
> data(GSE26712_eset, package="FULLVcuratedOvarianData")
> eset.probes <- GSE26712_eset
```

And using the authors' own version from GEO, processed by RMA:

```
> set.seed(1)
> eset.geo <- getGEO("GSE26712")[[1]]
```

Use suboptimally debulked patients only, and make sure these are the same sample IDs in each dataset:

```
> eset.genes <- eset.genes[ ,!is.na(eset.genes$debulking)]
> eset.genes <- eset.genes[ ,eset.genes$debulking == "suboptimal"]
> eset.probes <- eset.probes[ ,!is.na(eset.probes$debulking)]
> eset.probes <- eset.probes[ ,eset.probes$debulking == "suboptimal"]
> eset.geo <- eset.geo[, eset.geo$source_name_ch1 == "Ovarian tumor"]
> eset.geo <- eset.geo[, grep("Suboptimal", eset.geo$characteristics_ch1.1)]
> identical( sampleNames(eset.geo), sampleNames(eset.probes) )
```

```
[1] TRUE
```

```
> identical( sampleNames(eset.geo), sampleNames(eset.genes) )
```

```
[1] TRUE
```

Prepare the Surv objects (overall survival) for each ExpressionSet:

```
> eset.genes$y <- Surv(eset.genes$days_to_death / 30, eset.genes$vital_status == "deceased")
> eset.probes$y <- Surv(eset.probes$days_to_death / 30, eset.probes$vital_status == "deceased")
> table( eset.geo$characteristics_ch1.2 )  ##all deaths are labelled DOD
```

```
                                         status: AWD (alive with disease)
                                   0                                    8
       status: DOD (dead of disease) status: NED (no evidence of disease)
                                  78                                    9
```

```
> eset.geo$y <- Surv(as.numeric(sub("survival years: ", "", eset.geo$characteristics_ch1.3)) * 365,
+                     eset.geo$characteristics_ch1.2 == "status: DOD (dead of disease)")
```

Load the Supplemental Table S3 for suboptimally debulked patients:

```
> signature.table <- read.delim(input_file, as.is=TRUE)
> nrow(signature.table)
```

```
[1] 572
```

```
> geneset.probes <- signature.table$Probe.Set
```

Use current Bioconductor mapping to gene symbols:

```
> geneset.genes <- getSYMBOL(geneset.probes, data="hgu133a.db")
> geneset.genes[is.na(geneset.genes)]  ##probesets that could not be mapped
```

```
  214152_at   215869_at   216821_at 221649_s_at 214151_s_at 215016_x_at
         NA          NA          NA          NA          NA          NA
221511_x_at 214487_s_at 213397_x_at   202409_at 217322_x_at 217399_s_at
         NA          NA          NA          NA          NA          NA
  217350_at 208523_x_at   214658_at   215474_at   201380_at   203799_at
         NA          NA          NA          NA          NA          NA
208527_x_at   210491_at 205905_s_at   216405_at   217469_at   216805_at
         NA          NA          NA          NA          NA          NA
  214935_at   215401_at 208490_x_at 216212_s_at 212254_s_at   217451_at
         NA          NA          NA          NA          NA          NA
215329_s_at   222381_at   207188_at 201639_s_at   212243_at   215488_at
         NA          NA          NA          NA          NA          NA
   37005_at   213013_at 221406_s_at   206659_at 220705_s_at   217034_at
         NA          NA          NA          NA          NA          NA
  209505_at 217326_x_at   217132_at 207891_s_at
         NA          NA          NA          NA
```

All of the probesets in this table should be found in the full versions of the ExpressionSets:

```
> summary(geneset.probes %in% featureNames(eset.probes))

   Mode    TRUE    NA's 
logical     572       0 

> summary(geneset.probes %in% featureNames(eset.geo))

   Mode    TRUE    NA's 
logical     572       0 
```

curatedOvarianData uses mappings from biomaRt, and some additional probesets are lost in the curatedOvarianData ExpressionSet. Those shown below as NA are missing in both biomaRt and Bioconductor, those showing gene symbols are present in Bioconductor but not Biomart:

```
> summary(geneset.genes %in% featureNames(eset.genes))

   Mode   FALSE    TRUE    NA's 
logical      60     512       0 

> geneset.genes[!geneset.genes %in% featureNames(eset.genes)]

  214152_at   219021_at   215869_at   216821_at 218473_s_at 221649_s_at 
         NA    "RNF121"          NA          NA  "COLGALT1"          NA 
214151_s_at 215016_x_at 221511_x_at 214487_s_at 213397_x_at   202409_at 
         NA          NA          NA          NA          NA          NA 
217322_x_at 217399_s_at   217350_at 208523_x_at 216940_x_at   214658_at 
         NA          NA          NA          NA      "YBX1"          NA 
  215474_at   201380_at   203799_at 208527_x_at   210491_at 205905_s_at 
         NA          NA          NA          NA          NA          NA 
  216405_at   217469_at   216805_at   214935_at   217462_at   215401_at 
         NA          NA          NA          NA      "MYRF"          NA 
208490_x_at 216212_s_at 209379_s_at   204726_at 212254_s_at   217451_at 
         NA          NA    "CCSER2"     "CDH13"          NA          NA 
211403_x_at   206964_at 215329_s_at   222381_at   207188_at 201639_s_at 
     "VCX2"     "NAT8B"          NA          NA          NA          NA 
  217069_at   212243_at   215488_at    37005_at   213013_at 221406_s_at 
     "MLL4"          NA          NA          NA          NA          NA 
  222196_at   222289_at   210711_at   206659_at 220705_s_at   217034_at 
"LOC389906"     "KCNC2" "LINC00260"          NA          NA          NA 
  209505_at 213510_x_at 217326_x_at   221797_at   217132_at 207891_s_at 
         NA   "USP32P2"          NA     "OXLD1"          NA          NA 

> ##keep the ones present in the data:
> geneset.genes <- geneset.genes[ geneset.genes %in% featureNames(eset.genes) ]
```

This reported model uses the univariate Cox regression coefficients as weights for the linear risk score, so can we reproduce those given in Supplemental Table S3?

```
> rowcox.probes <- rowCoxTests(X=exprs(eset.probes[geneset.probes,]), y=eset.probes$y)
> rowcox.geo <- rowCoxTests(X=exprs(eset.geo[geneset.probes,]), y=eset.geo$y)
> coxc <- data.frame(published=log(signature.table$Cox.Hazard.Ratio),
+                    GEO=rowcox.geo$coef,
+                    curatedOvarianData=rowcox.probes$coef,
+                    row.names=rownames(rowcox.probes))
> pairs(coxc)
```

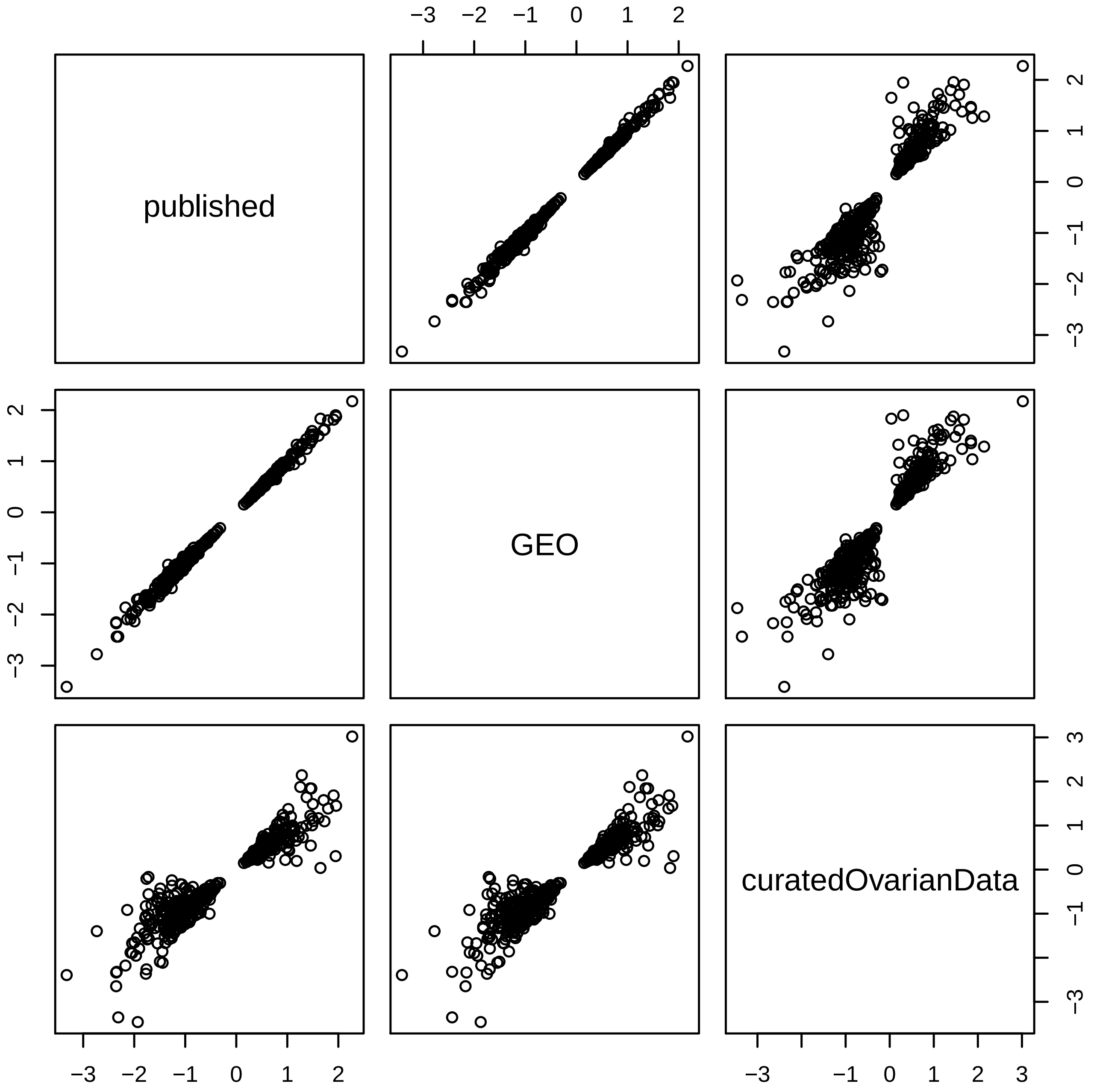


Figure 43: Cox coefficients: as published in Supplemental Table S3, calculated using the GEO dataset, and calculated using the FULLVcuratedOvarianData package.

Now we reproduce Figure 2B using each version of the data. Define a function to do the leave-one-out as described. In each iteration select genes whose univariate Cox regression Wald Test p-value is less than 0.01, and define a linear risk score whose coefficients are equal to the univariate Cox regression coefficient. Call the left-out patient "Poor prognosis" if its prediction is above the median risk score. Then use this to calculate patient predictions for each of the three datasets:

```
> getRisk <- function(i, eset) {
+     featureNames(eset) <- make.names(featureNames(eset))
+     Xlearn <- t(exprs(eset))[-i, ]
+     Ylearn <- eset$y[-i]
+     model = plusMinusSurv(Xlearn, Ylearn, modeltype="compoundcovariate", tuningpar="pval", lambda=0.01)
+     ret = predict(model, newdata=t(exprs(eset)), type="lp")
+     ifelse(ret@lp[i] > median(ret@lp), "Poor prognosis", "Good prognosis")
+ }
> risk.genes <- sapply(1:ncol(eset.genes), getRisk, eset.genes)
> risk.probes <- sapply(1:ncol(eset.probes), getRisk, eset.probes)
> risk.geo <- sapply(1:ncol(eset.geo), getRisk, eset.geo)
```

Finally, we create the model as it will be used for meta-analysis. For the official model, we use the coefficients provided by the authors, map these to HGNC symbols, averaging coefficients for probesets which map to the same gene:

```
> coxc$hgnc <- getSYMBOL(geneset.probes, data="hgu133a.db")
> coefs <- aggregate(coxc$GEO, list(coxc$hgnc), mean)
> coefs <- structure(coefs[ ,2], .Names=coefs[ ,1])
> model.official <- new("ModelLinear", coefficients=coefs, modeltype="compoundcovariate")
```

We also build a model by reproducing the authors' methods, but using the already mapped curatedOvarianData package.

```
> Xlearn <- t(exprs(eset.genes))
> Xlearn <- Xlearn[ ,!grepl("///", colnames(Xlearn))]  ##remove probes which mapped to multiple genes
> Ylearn <- eset.genes$y
> model.cod = plusMinusSurv(Xlearn, Ylearn,
+                           modeltype="compoundcovariate",
+                           tuningpar="pval", lambda=0.01)@model
> ##prune zero coefficients:
> model.cod@coefficients <- model.cod@coefficients[abs(model.cod@coefficients) > 0]
```

Look at the overlap of the genes found in each of these models:

```
> length(model.official@coefficients)

[1] 491

> length(model.cod@coefficients)

[1] 337

> length(intersect(names(model.cod@coefficients), names(model.official@coefficients)))

[1] 206
```

Finally, save these models:

```
> save(model.official, model.cod, file=model_file)
> ##plotKM is from the survHD package
> plotKM(y=eset.geo$y,
+            strata=factor(risk.geo, levels=c("Poor prognosis", "Good prognosis")),
+            show.PV=FALSE,
+            main="Using GEO data\n Compare to Figure 2B")
```

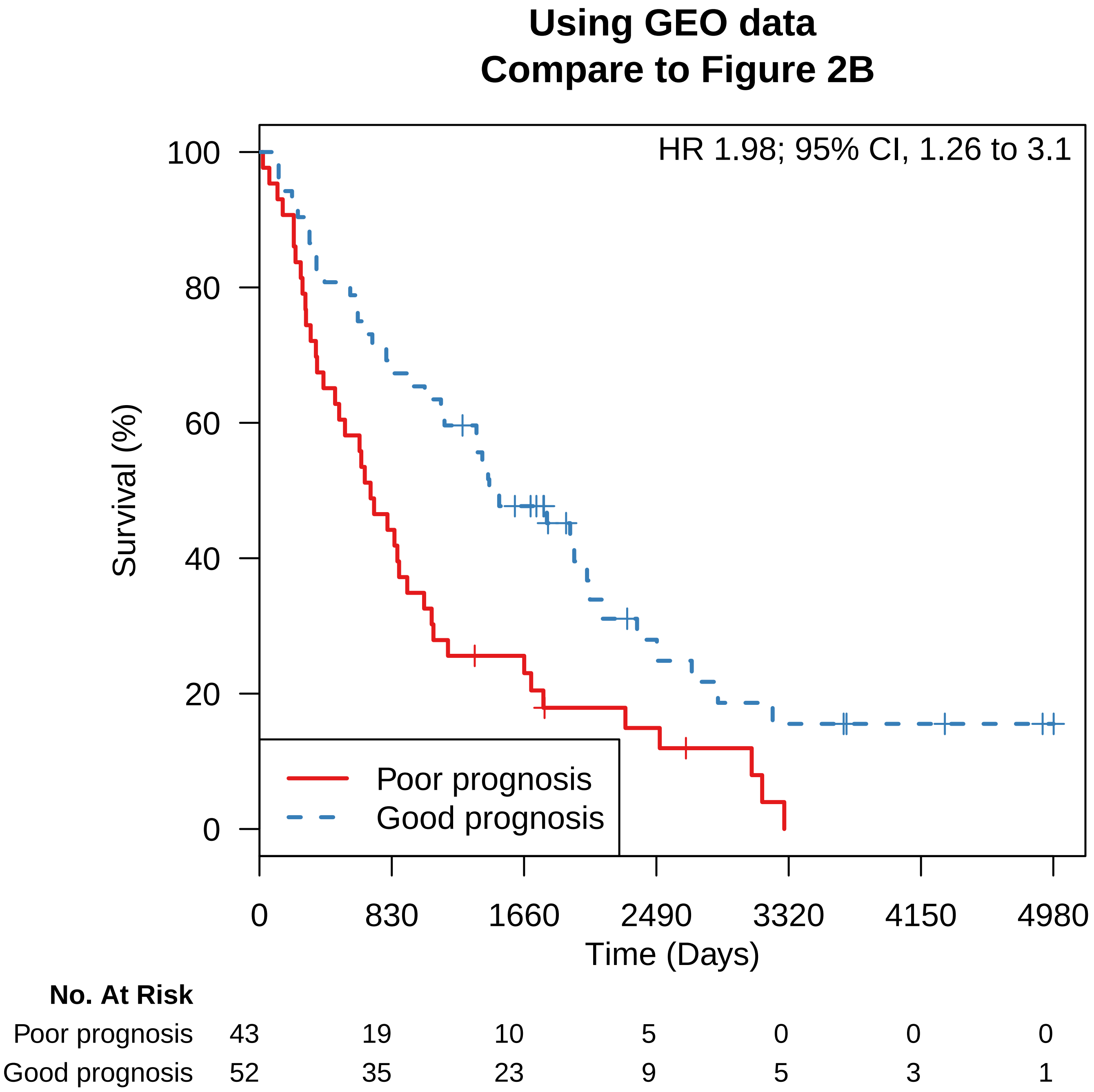


Figure 44: Using the authors' own expression data from GEO looks exactly like Figure 2B, although the numbers of patients in high and low risk groups do not match the legend in the paper.

```
> plotKM(y=eset.probes$y,
+            strata=factor(risk.probes, levels=c("Poor prognosis", "Good prognosis")),
+            show.PV=FALSE,
+            main="Using FULLVcuratedOvarianData\n Compare to Figure 2B")
```

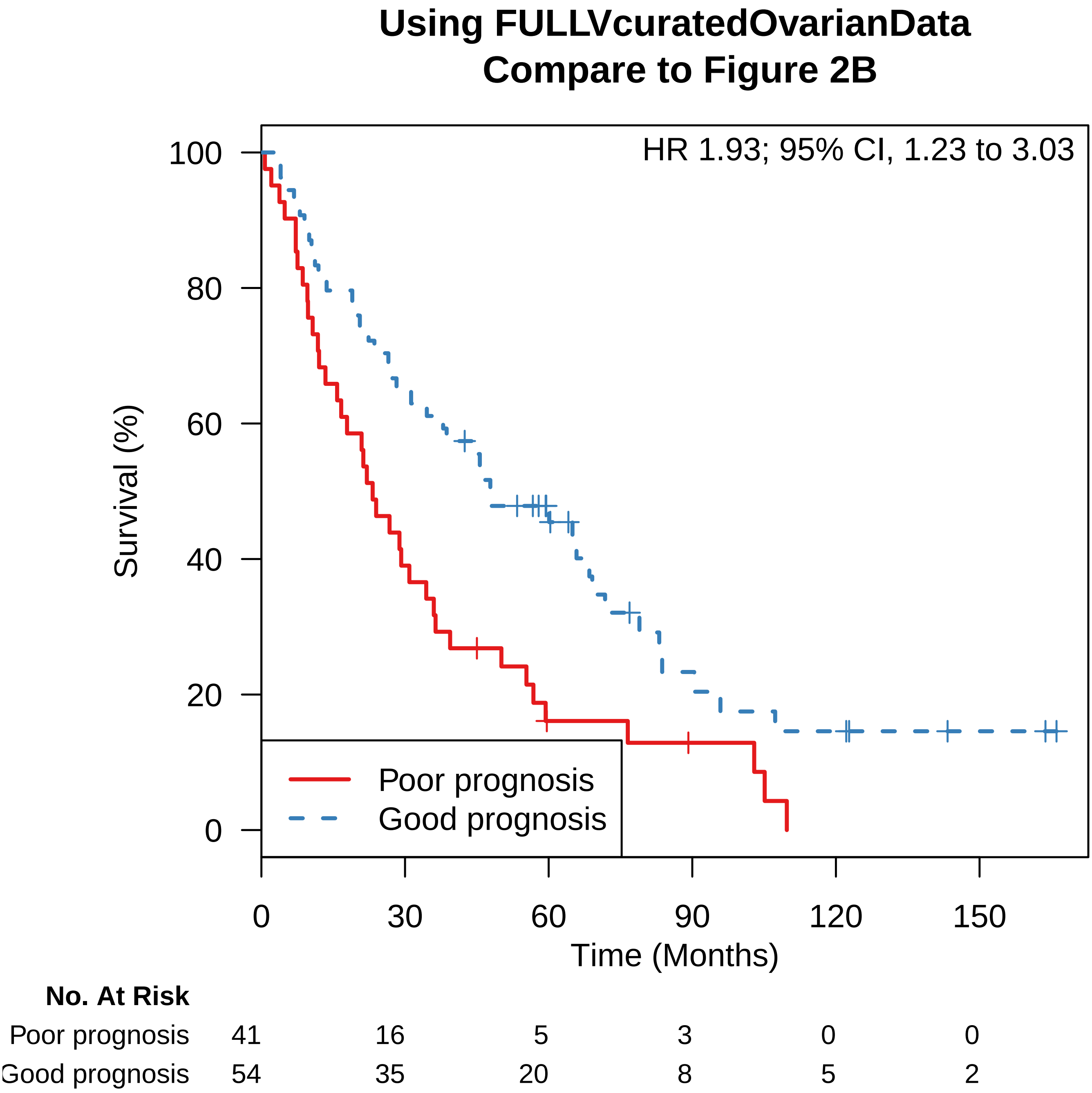


Figure 45: Using FULLVcuratedOvarianData (preprocessed by frozen RMA), it is still extremely close to Figure 2B.

```
> plotKM(y=eset.genes$y,
+            strata=factor(risk.genes, levels=c("Poor prognosis", "Good prognosis")),
+            show.PV=FALSE,
+            main="Using curatedOvarianData\n Compare to Figure 2B")
```

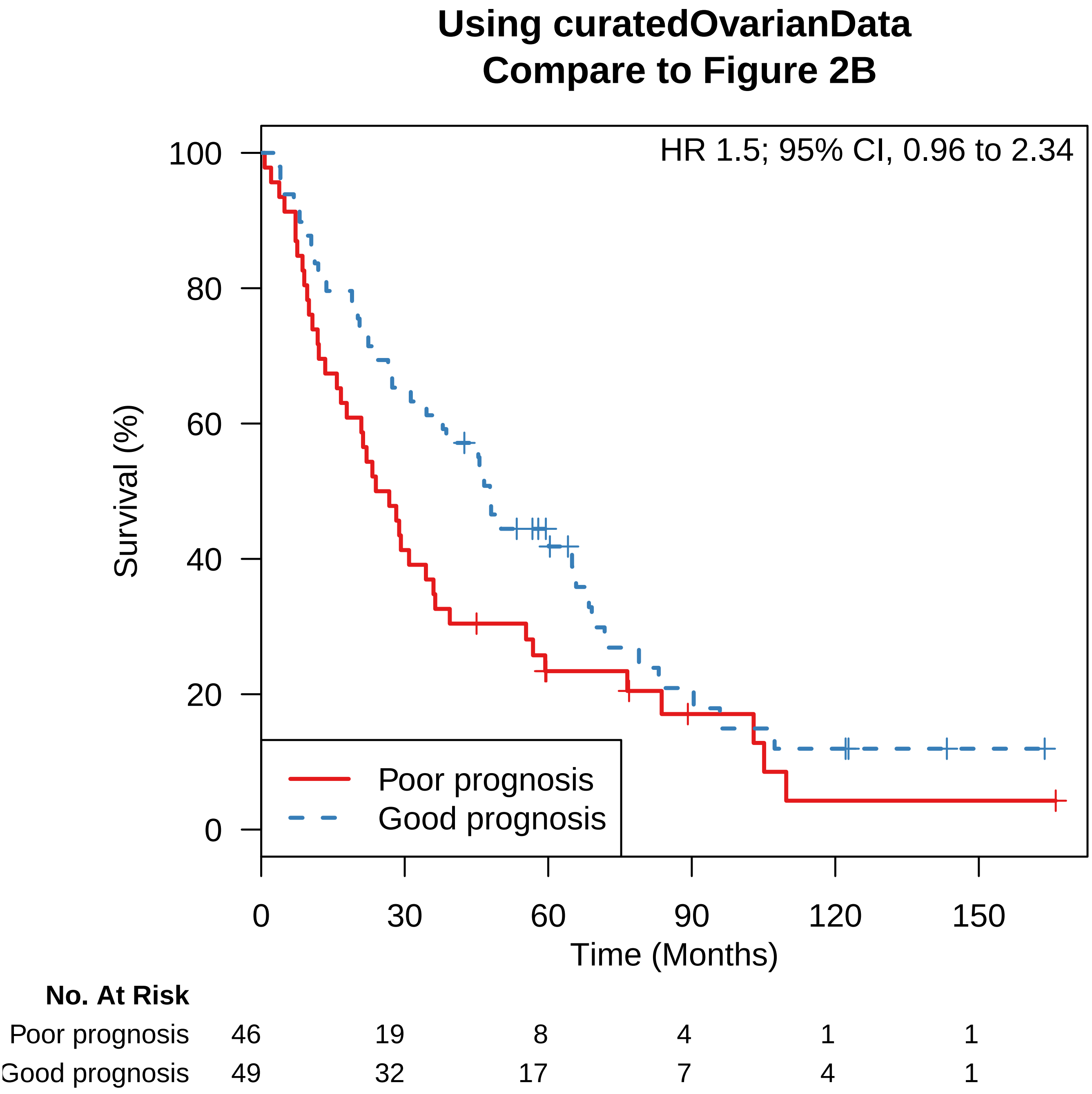


Figure 46: And using curatedOvarianData, where probesets are already collapsed to HGNC symbols.

Session Info

- R version 3.0.1 (2013-05-16), `x86_64-unknown-linux-gnu`
- Locale: `LC_CTYPE=en_US.UTF-8`, `LC_NUMERIC=C`, `LC_TIME=en_US.UTF-8`, `LC_COLLATE=en_US.UTF-8`, `LC_MONETARY=en_US.UTF-8`, `LC_MESSAGES=en_US.UTF-8`, `LC_PAPER=C`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.UTF-8`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, parallel, splines, stats, utils
- Other packages: affy 1.38.1, annotate 1.38.0, AnnotationDbi 1.22.6, Biobase 2.20.0, BiocGenerics 0.6.0, curatedOvarianData 1.0.1, DBI 0.2-7, devtools 1.2, GEOquery 2.26.1, hgu133a.db 2.9.0, Hmisc 3.10-1.1, org.Hs.eg.db 2.9.0, penalized 0.9-42, prodlim 1.3.7, RColorBrewer 1.0-5, RSQLite 0.11.4, survC1 1.0-2, survcomp 1.10.0, survHD 0.99.1, survival 2.37-4
- Loaded via a namespace (and not attached): affyio 1.28.0, BiocInstaller 1.10.1, bootstrap 2012.04-0, cluster 1.14.4, digest 0.6.3, evaluate 0.4.3, grid 3.0.1, httr 0.2, IRanges 1.18.1, KernSmooth 2.23-10, lattice 0.20-15, memoise 0.1, preprocessCore 1.22.0, RCurl 1.95-4.1, rmeta 2.16, stats4 3.0.1, stringr 0.6.2, SuppDists 1.1-9, survivalROC 1.0.3, tools 3.0.1, whisker 0.3-2, XML 3.96-1.1, xtable 1.7-1, zlibbioc 1.6.0