Correlating Survival by Pathways

Keith A. Baggerly, E. Shannon Neeley and Kevin R. Coombes

October 19, 2007

1 Introduction

Dressman et al found that overall survival in NR patients was clearly stratified by the dichotomized (high/low) scores for several pathways. In particular, they note significant differences using Src, E2F3, and Myc, while focusing most attention on the first two due to lack of a therapeutic option to address the third. Here, we check how well survival is split by our scores.

2 Options and Libraries

```
> options(width = 80)

> library(survival)

> load(file.path("RDataObjects", "clinicalInfo.Rda"))
> load(file.path("RDataObjects", "pathwayScores.Rda"))
```

3 Build Survival Objects

```
> survDressman <- with(clinicalInfo, Surv(SurvMonths, Censoring ==
+     "Dead"))
> survBild <- with(clinicalInfo, Surv(SurvMonths, CensoringBild ==
+     "Dead"))
> rownames(survDressman) <- rownames(clinicalInfo)
> rownames(survBild) <- rownames(clinicalInfo)
```

We will be performing quite a few tests here, and most of them will involve using survdiff or survfit in very specific ways. To save ourselves some typing, we first build some wrapper functions.

```
> pathTest <- function(surv, clin, score, resp = "All") {
+     if (resp == "All") {
+         ourTest <- survdiff(surv ~ (score > median(score)), data = cbind(clin,
+             score[rownames(clin)]))
+     }
+     else {
+         ourTest <- survdiff(surv[Response == resp] ~ (score[Response ==
+             resp] > median(score)), data = cbind(clin, score[rownames(clin)]))
```

```
+     }
+     return(ourTest)
+ }
> pathKM <- function(surv, clin, score, resp = "All") {
+     if (resp == "All") {
+         ourTest <- survfit(surv ~ (score > median(score)), data = cbind(clin,
+             score[rownames(clin)]))
+     }
+     else {
+         ourTest <- survfit(surv[Response == resp] ~ (score[Response ==
+             resp] > median(score)), data = cbind(clin, score[rownames(clin)]))
+     }
+     return(ourTest)
+ }
```

Having defined the helper functions, we now assess significance and construct KM-plots for all pathway/pathwayScore/patientSubgroup combinations.

```
> pathways <- names(pathwayScoresCEL)
> pathways

[1] "Bcat" "E2F3" "Myc"  "Ras"  "Src"

> pathwayScores <- c("pathwayScoresCEL", "pathwayScoresCELBild",
+     "pathwayScoresCELBatch", "pathwayScoresCELBildBatch", "pathwayScoresXLS",
+     "pathwayScoresXLSBild", "pathwayScoresXLSBatch", "pathwayScoresXLSBildBatch")
> patientSubgroups <- c("NR", "CR", "All")
> pathwaySurvTests <- vector("list", length(pathways))
> names(pathwaySurvTests) <- pathways
> for (pathway in pathways) {
+     pathwaySurvTests[[pathway]] <- vector("list", length(pathwayScores))
+     names(pathwaySurvTests[[pathway]]) <- pathwayScores
+     for (pathwayScore in pathwayScores) {
+         pathwaySurvTests[[pathway]][[pathwayScore]] <- vector("list",
+             length(patientSubgroups))
+         names(pathwaySurvTests[[pathway]][[pathwayScore]]) <- patientSubgroups
+     }
+ }
> pathwayKMFits <- pathwaySurvTests
> pathwaySurvTestsBild <- pathwaySurvTests
> pathwayKMFitsBild <- pathwaySurvTests
> for (pathway in pathways) {
+     for (pathwayScore in pathwayScores) {
+         for (patientSubgroup in patientSubgroups) {
+             pathwaySurvTests[[pathway]][[pathwayScore]][[patientSubgroup]] <- pathTest(survDressman,
+                 clinicalInfo, get(pathwayScore)[[pathway]], patientSubgroup)
+             pathwayKMFits[[pathway]][[pathwayScore]][[patientSubgroup]] <- pathKM(survDressman,
+                 clinicalInfo, get(pathwayScore)[[pathway]], patientSubgroup)
+             pathwaySurvTestsBild[[pathway]][[pathwayScore]][[patientSubgroup]] <- pathTest(survBild,
```

```
+                 clinicalInfo, get(pathwayScore)[[pathway]], patientSubgroup)
+             pathwayKMFitsBild[[pathway]][[pathwayScore]][[patientSubgroup]] <- pathKM(survBild,
+                 clinicalInfo, get(pathwayScore)[[pathway]], patientSubgroup)
+         }
+     }
+ }
```

Given the survival tests, let's extract the p-values in a more usable form.

```
> pathwaySurvPVals <- vector("list", length(pathways))
> names(pathwaySurvPVals) <- pathways
> pathwaySurvBildPVals <- pathwaySurvPVals
> for (pathway in pathways) {
+     temp <- matrix(0, length(pathwayScores), length(patientSubgroups))
+     rownames(temp) <- pathwayScores
+     colnames(temp) <- patientSubgroups
+     tempBild <- temp
+     for (pathwayScore in pathwayScores) {
+         for (patientSubgroup in patientSubgroups) {
+             temp[pathwayScore, patientSubgroup] <- 1 - pchisq(pathwaySurvTests[[pathway]][[pathwayScor
+                 1)
+             tempBild[pathwayScore, patientSubgroup] <- 1 - pchisq(pathwaySurvTestsBild[[pathway]][[pat
+                 1)
+         }
+     }
+     pathwaySurvPVals[[pathway]] <- temp
+     pathwaySurvBildPVals[[pathway]] <- tempBild
+ }
```

Now that we have p-values, we'd like to couple these with KM-plots. Before we produce these plots, let's check the order of the arguments returned.

```
> names(pathwayKMFits$Src$pathwayScoresCEL$NR)

 [1] "n"             "time"          "n.risk"        "n.event"
 [5] "surv"          "type"          "ntimes.strata" "strata"
 [9] "strata.all"    "std.err"       "upper"         "lower"
[13] "conf.type"     "conf.int"      "call"

> pathwayKMFits$Src$pathwayScoresCEL$NR$strata.all

score[Response == resp] > median(score)=FALSE
                                           13
 score[Response == resp] > median(score)=TRUE
                                           21
```

The first curve returned corresponds to cases where the pathway score is not above the median (ie, it's closer to the normal state), and the second to the case where the pathway activation level is high.

Let's assemble a wrapper script for producing quartets of KM-plots.

```
> par(mfrow = c(2, 2))
> for (pathwayScore in scoreList[1:4]) {
+     plot(pathwayKMFits[[pathway]][[pathwayScore]][[patientSubgroup]],
+         col = c("blue", "red"), lty = 1, lwd = 1, xlab = "Surv in Months",
+         ylab = "Propn Alive", main = paste(pathway, pathwayScore,
+             patientSubgroup))
+     legend(x = 0.6 * par()$usr[2], y = 0.95 * par()$usr[4], legend = c(paste(c("Low",
+         "High"), pathway)), col = c("blue", "red"), lty = 1,
+         cex = 0.75)
+     text(x = 0.05 * par()$usr[2], y = 0.05 * par()$usr[4], paste("p =",
+         format(1 - pchisq(pathwaySurvTests[[pathway]][[pathwayScore]][[patientSubgroup]]$chisq,
+             1), digits = 4)), cex = 0.75, adj = c(0, 0))
+ }
> par(mfrow = c(1, 1))
```

4 Src

```
> pathway <- "Src"
> pathwaySurvPVals[[pathway]]
```

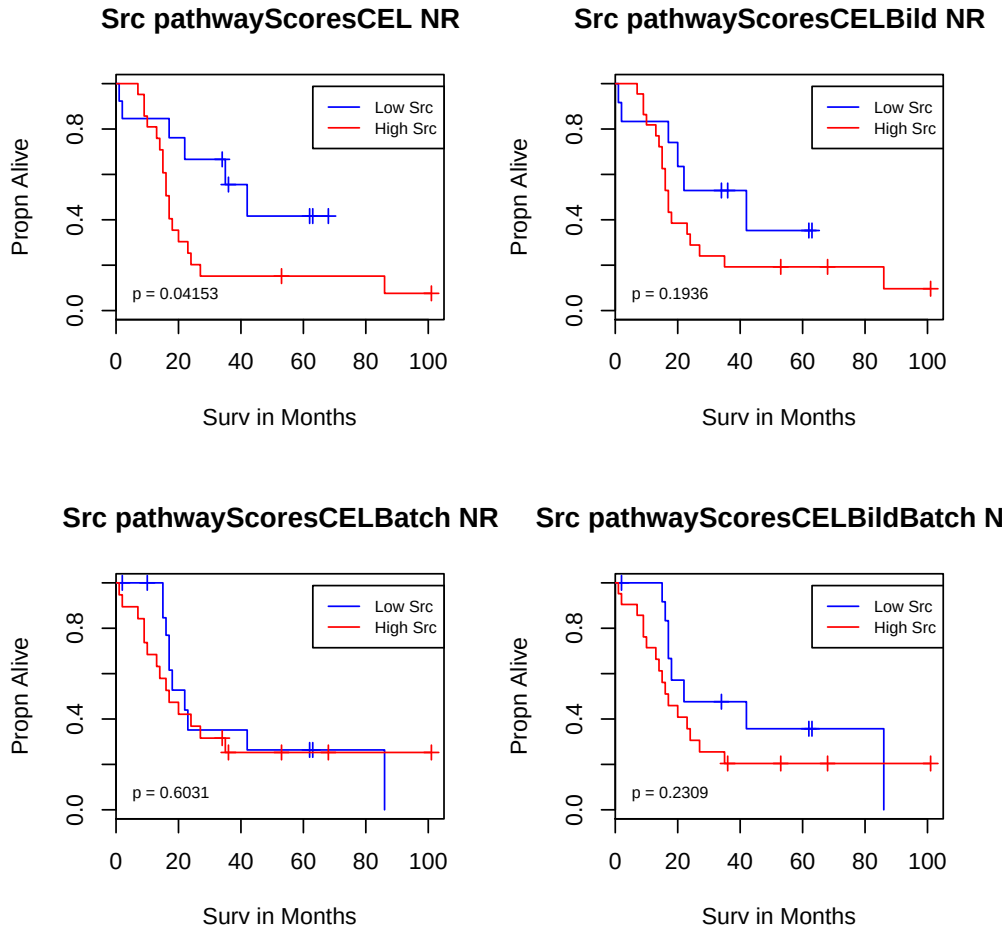
```
                                NR        CR        All
pathwayScoresCEL          0.0415277 0.1997895 0.01313789
pathwayScoresCELBild      0.1936404 0.1258556 0.01408270
pathwayScoresCELBatch     0.6031257 0.3245034 0.66533870
pathwayScoresCELBildBatch 0.2309310 0.7263481 0.51217863
pathwayScoresXLS          0.3898021 0.3868314 0.92158411
pathwayScoresXLSBild      0.3156260 0.7707464 0.35742769
pathwayScoresXLSBatch     0.5074334 0.3324089 0.75556800
pathwayScoresXLSBildBatch 0.1534417 0.7975712 0.56270788
```

```
> pathwaySurvBildPVals[[pathway]]
```

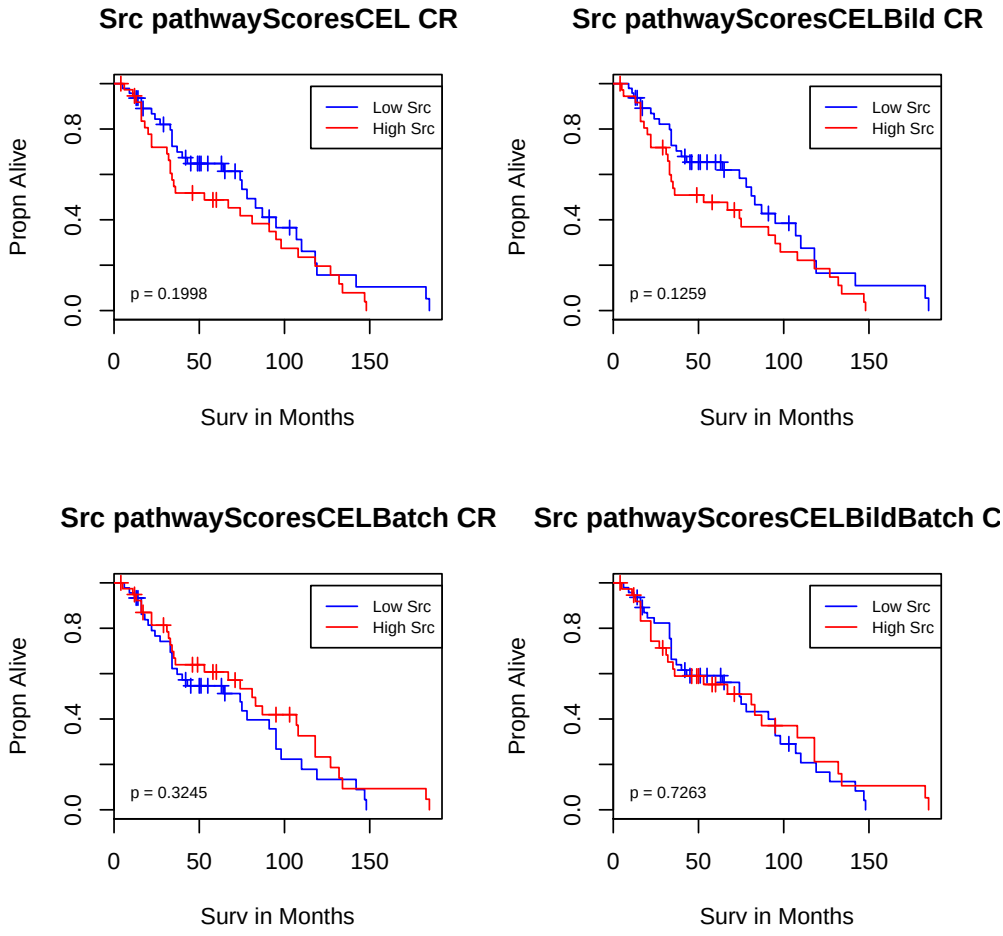
```
                                NR        CR        All
pathwayScoresCEL          0.03153667 0.2901204 0.01227788
pathwayScoresCELBild      0.15763100 0.1981157 0.01410347
pathwayScoresCELBatch     0.76001951 0.3662188 0.68355890
pathwayScoresCELBildBatch 0.17689394 0.6827717 0.13560971
pathwayScoresXLS          0.56422450 0.1490313 0.67892861
pathwayScoresXLSBild      0.45238626 0.8077543 0.69267344
pathwayScoresXLSBatch     0.66988988 0.1213613 0.39238475
pathwayScoresXLSBildBatch 0.23930279 0.6644604 0.71867431
```

In looking at Src, we see some marginally significant differences when we stratify survival of the NRs or All by pathway CEL scores. This is rendered slightly more significant if the Bild survival values are used. However, these differences are largely confounded with run batch, and as such disappear when corrections for batch are employed. There is no real structure when the XLS quantifications are employed.

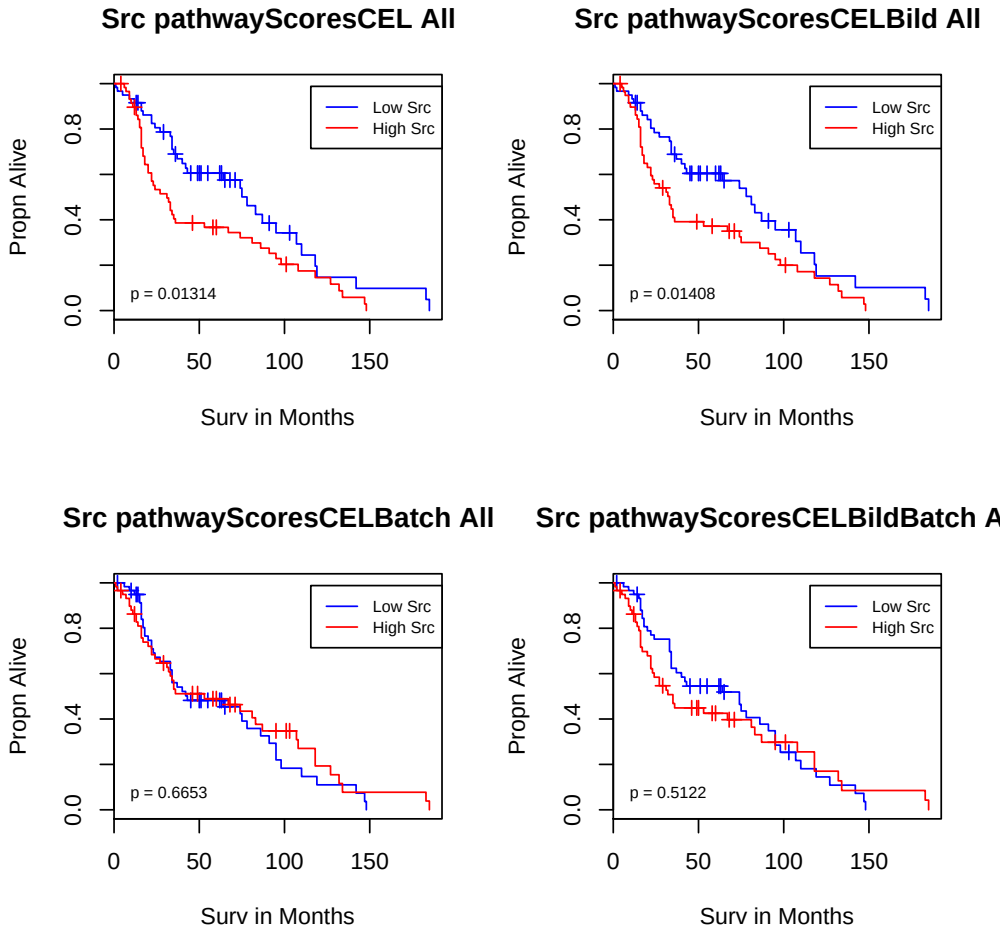
```
> scoreList <- pathwayScores[1:4]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```



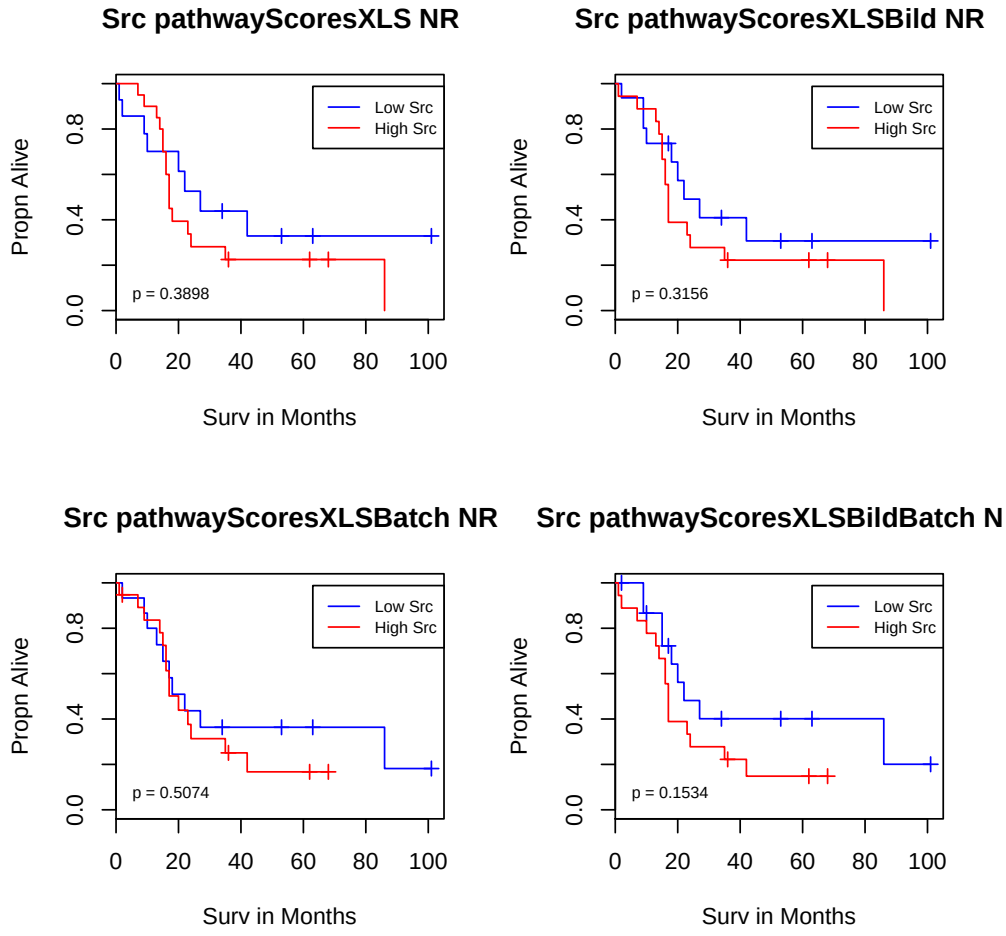

```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```



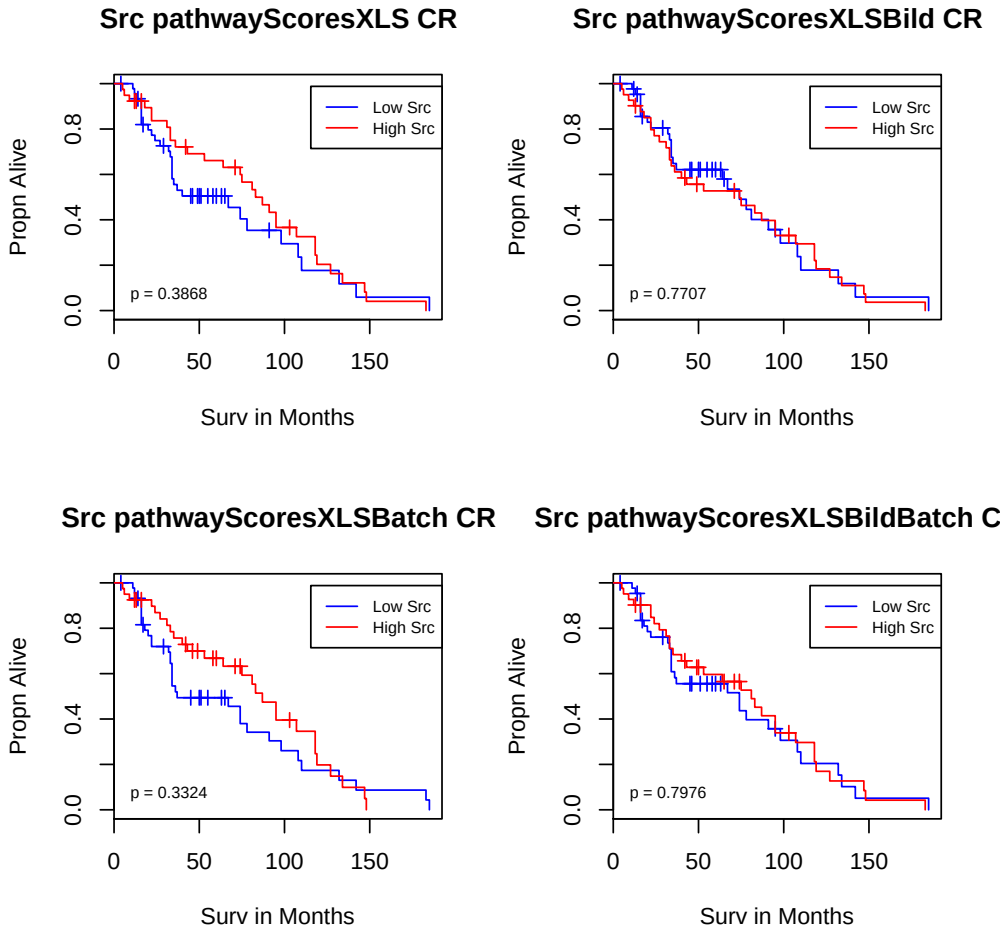

```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



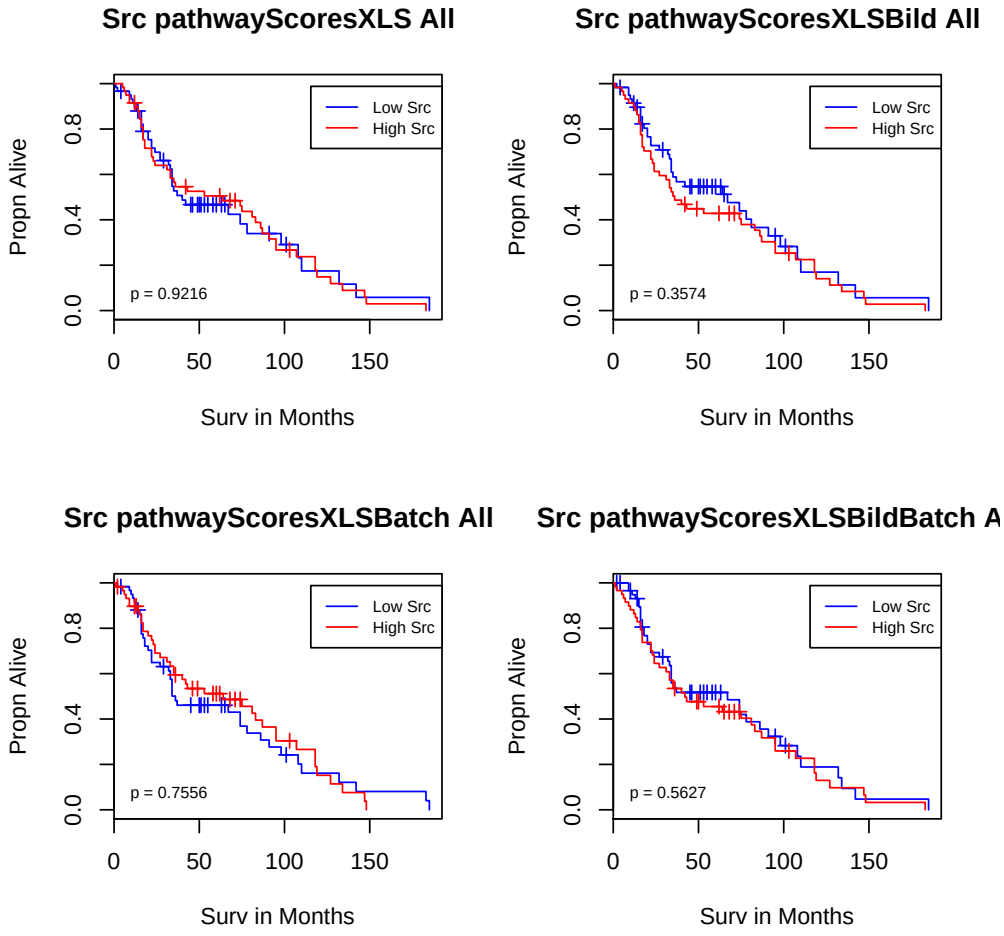

```
> scoreList <- pathwayScores[5:8]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



5 E2F3

```
> pathway <- "E2F3"
> pathwaySurvPVals[[pathway]]
```

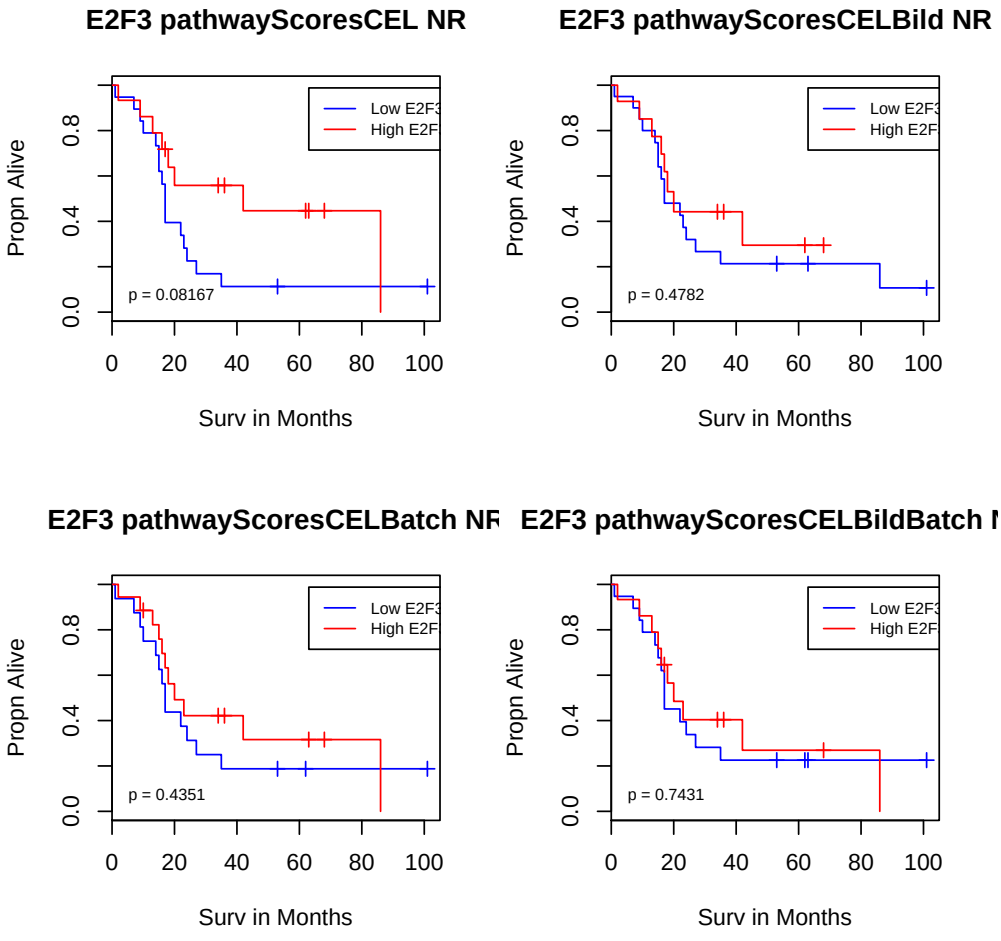
	NR	CR	All
pathwayScoresCEL	0.08167078	0.002441720	0.0003396044
pathwayScoresCELBild	0.47824499	0.001955799	0.0007973347
pathwayScoresCELBatch	0.43508782	0.083082512	0.0828477920
pathwayScoresCELBildBatch	0.74308048	0.056440042	0.0443868904
pathwayScoresXLS	0.66487592	0.379591774	0.5129602084
pathwayScoresXLSBild	0.18090745	0.229973963	0.6407170132
pathwayScoresXLSBatch	0.66487592	0.181881203	0.2946277992
pathwayScoresXLSBildBatch	0.20394044	0.236292230	0.5301186572

```
> pathwaySurvBildPVals[[pathway]]
```

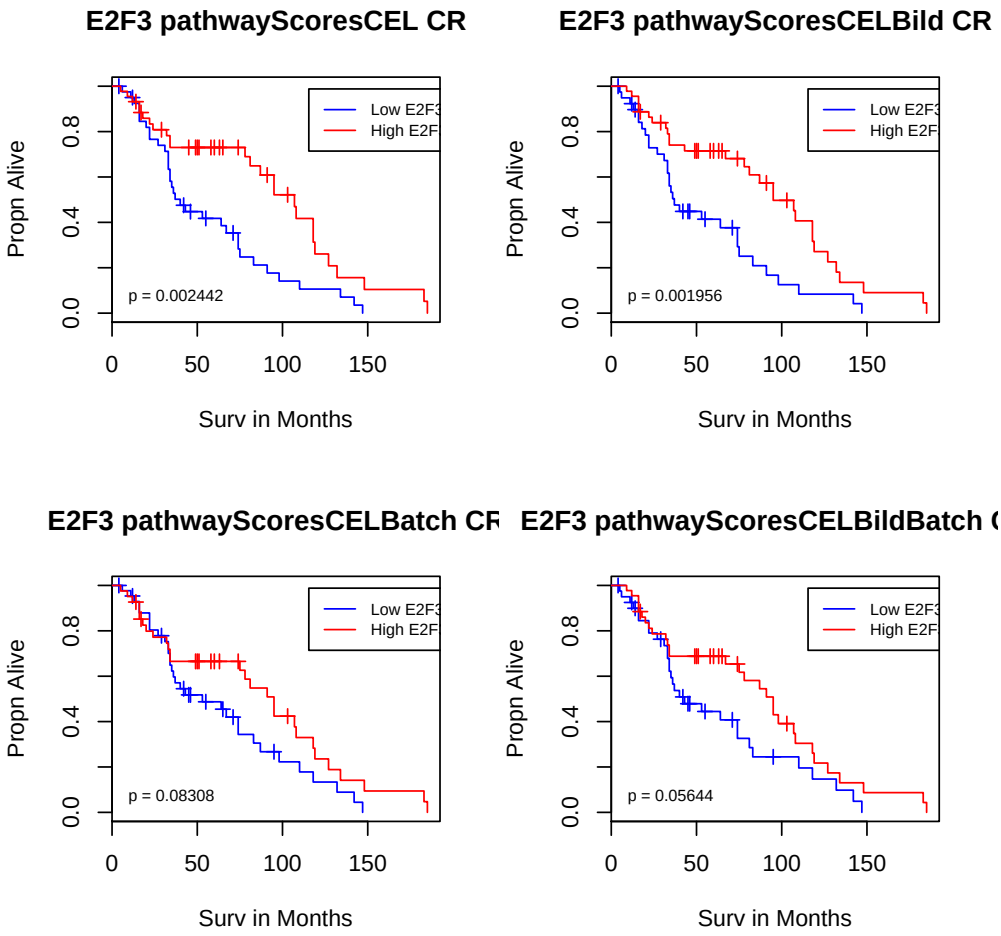
```
                                NR          CR          All
pathwayScoresCEL          0.05965767 0.007762867 0.0006908506
pathwayScoresCELBild      0.38968077 0.001891373 0.0004995484
pathwayScoresCELBatch     0.56411205 0.099778994 0.1288195814
pathwayScoresCELBildBatch 0.61969736 0.039642576 0.0250556442
pathwayScoresXLS          0.80662601 0.277217419 0.3808701744
pathwayScoresXLSBild      0.25705294 0.163875535 0.5017499621
pathwayScoresXLSBatch     0.80662601 0.042942015 0.0997551992
pathwayScoresXLSBildBatch 0.28033304 0.070576835 0.2303580410
```

Looking at E2F3, we see some separation in survival for the CRs and All, but not really for the NRs. The separation becomes less significant when we correct for batch, but does not wholly disappear. The structure present is greatly attenuated when the XLS quantifications are used.

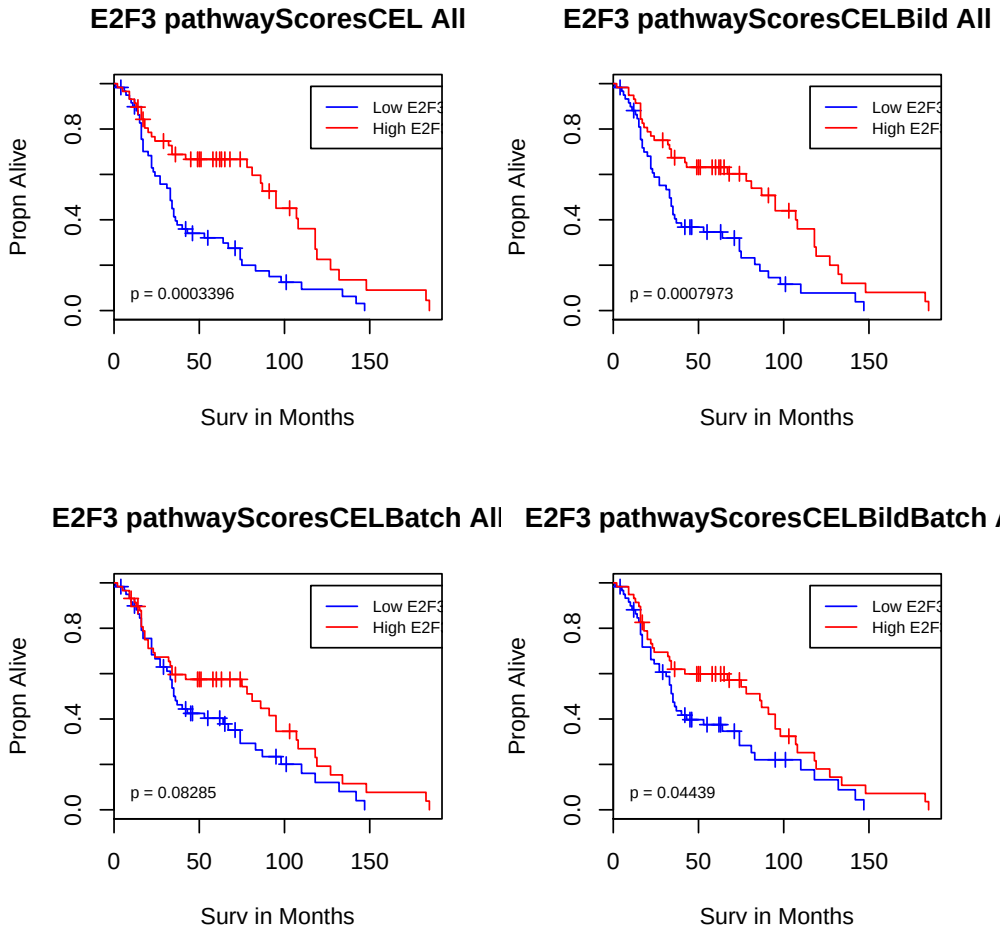
```
> scoreList <- pathwayScores[1:4]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```



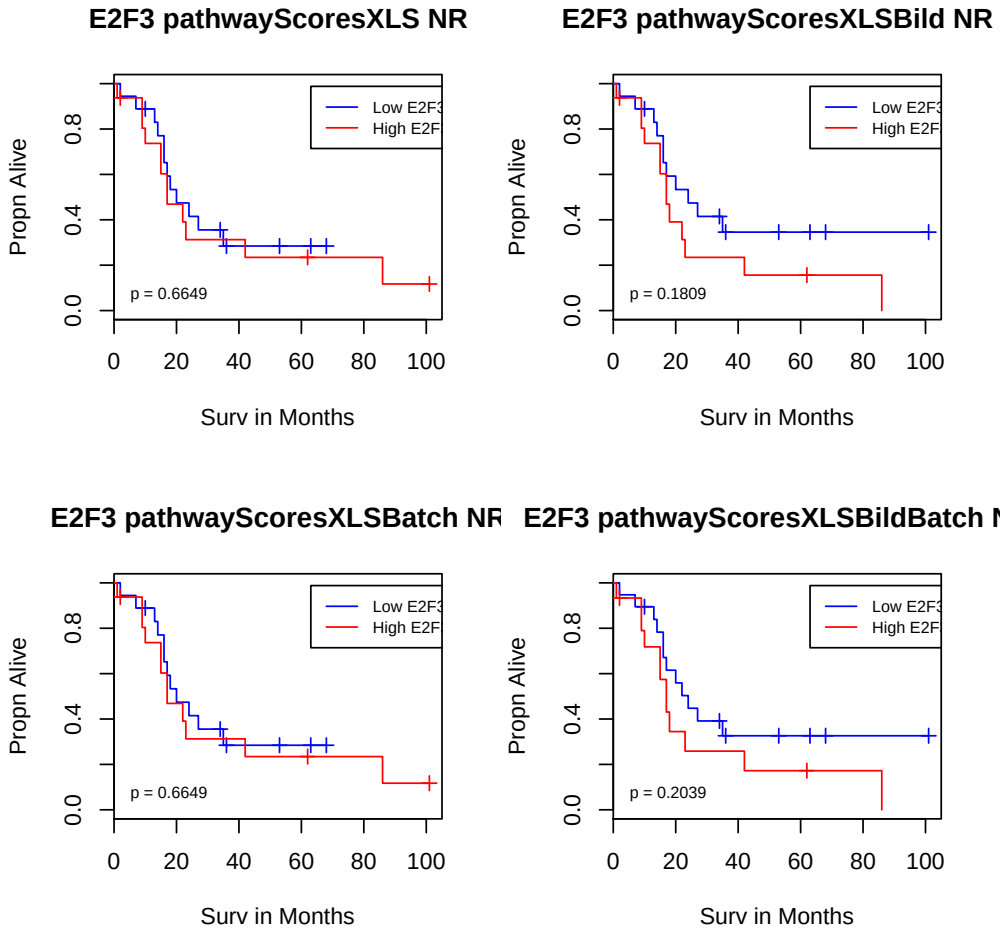
```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```



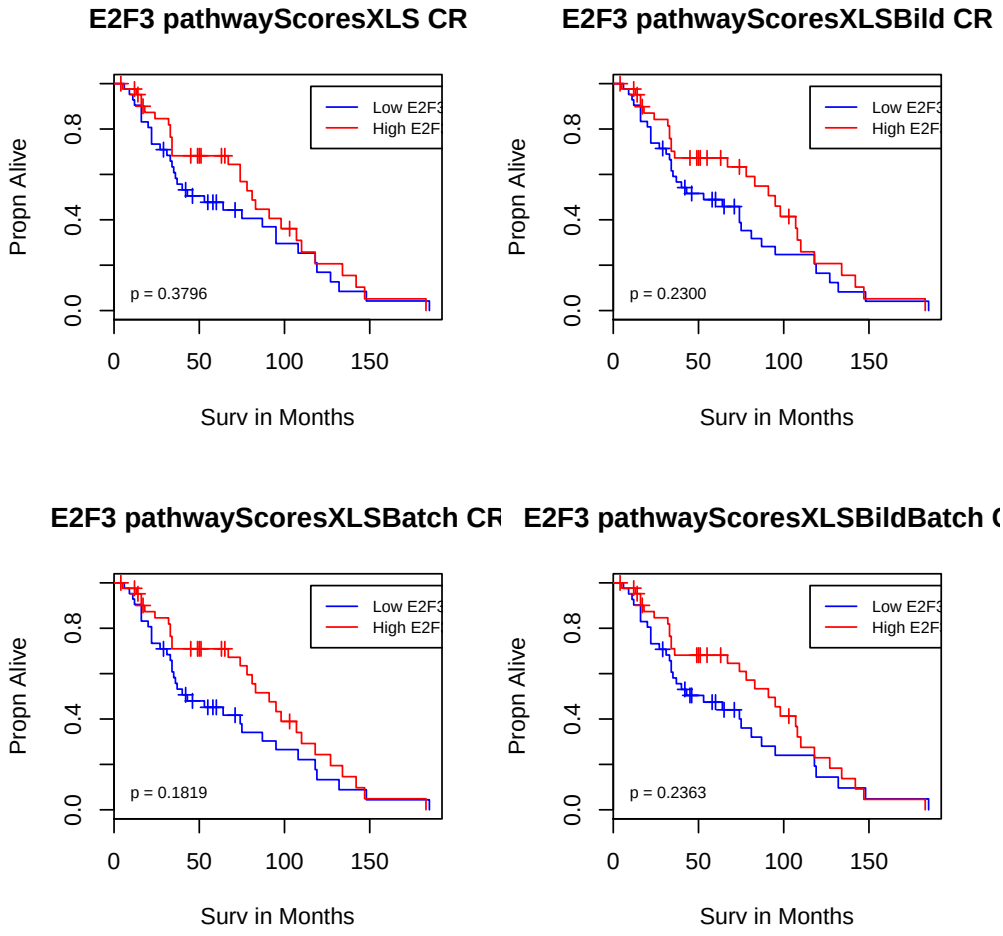

```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



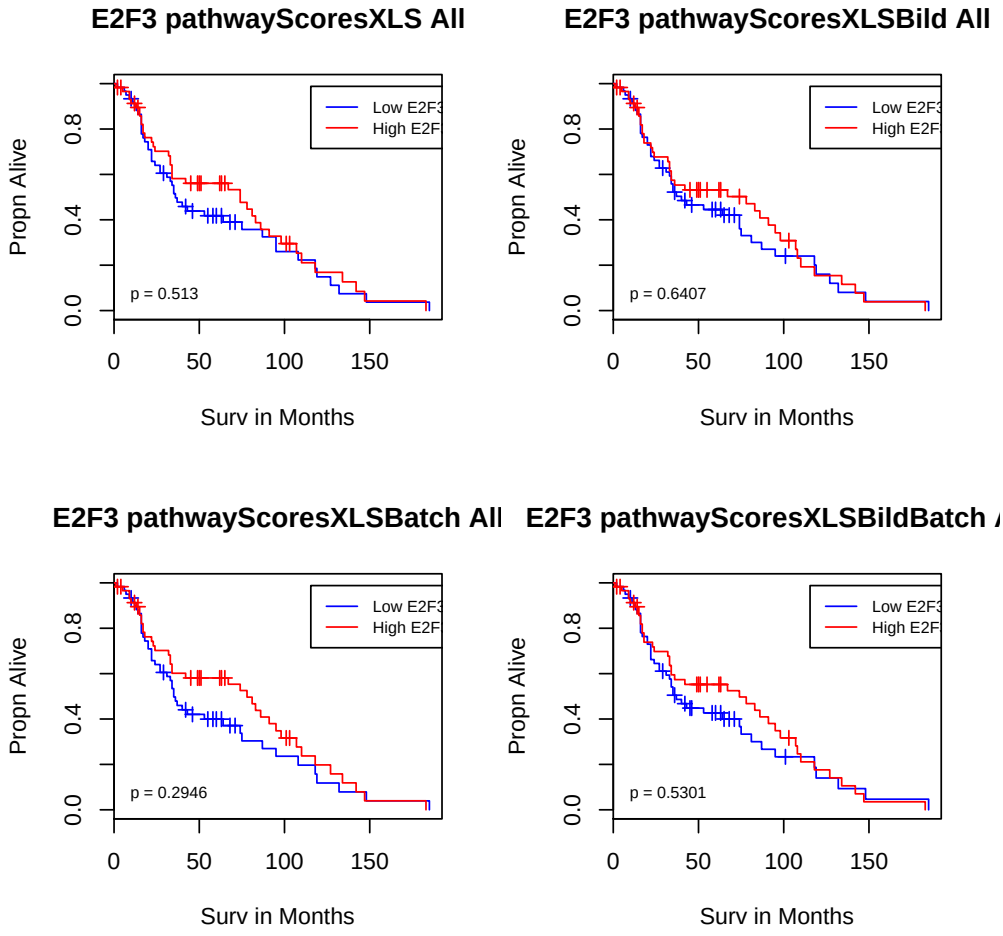

```
> scoreList <- pathwayScores[5:8]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



6 Myc

```
> pathway <- "Myc"
> pathwaySurvPVals[[pathway]]
```

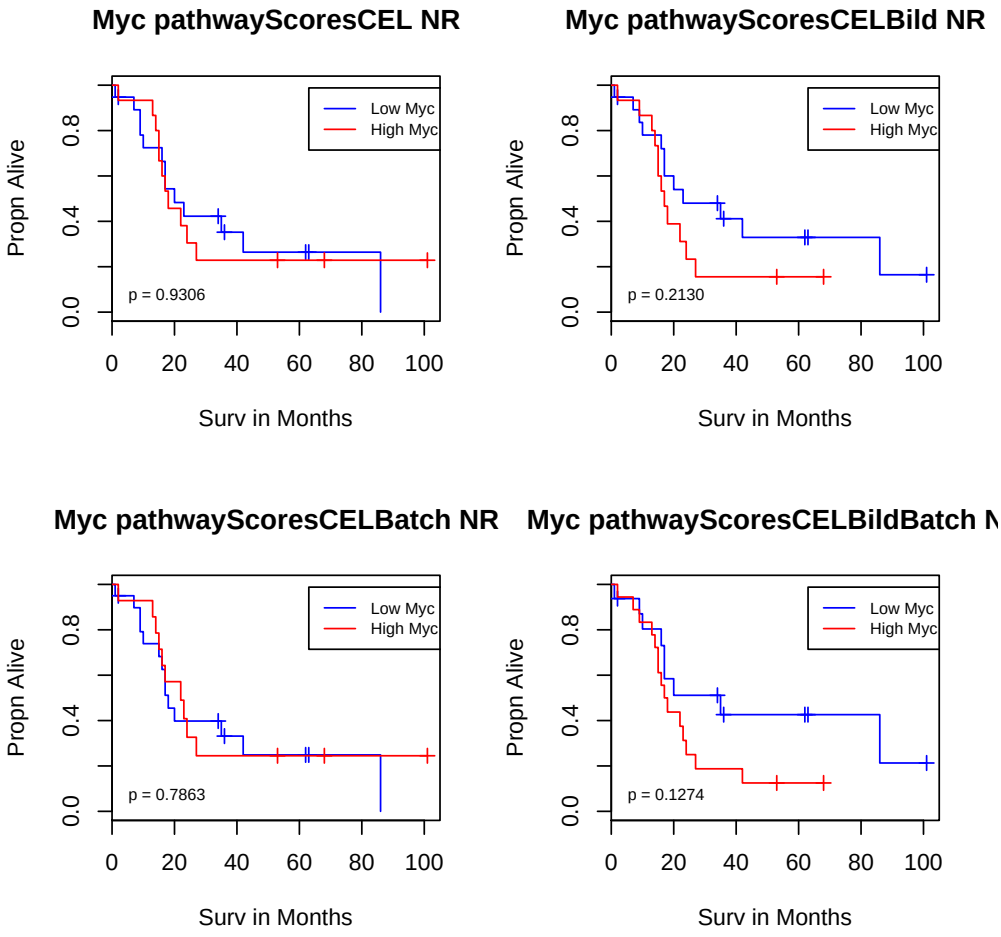
```
                                NR         CR         All
pathwayScoresCEL          0.9305531 0.08988875 0.09431339
pathwayScoresCELBild      0.2129562 0.38414412 0.65096875
pathwayScoresCELBatch     0.7863266 0.08024409 0.05398578
pathwayScoresCELBildBatch 0.1273757 0.21697245 0.80828977
pathwayScoresXLS          0.3866791 0.40510427 0.13208745
pathwayScoresXLSBild      0.5786686 0.09311163 0.05793782
pathwayScoresXLSBatch     0.3866791 0.55695485 0.20164391
pathwayScoresXLSBildBatch 0.4026899 0.22251465 0.08183767
```

```
> pathwaySurvBildPVals[[pathway]]
```

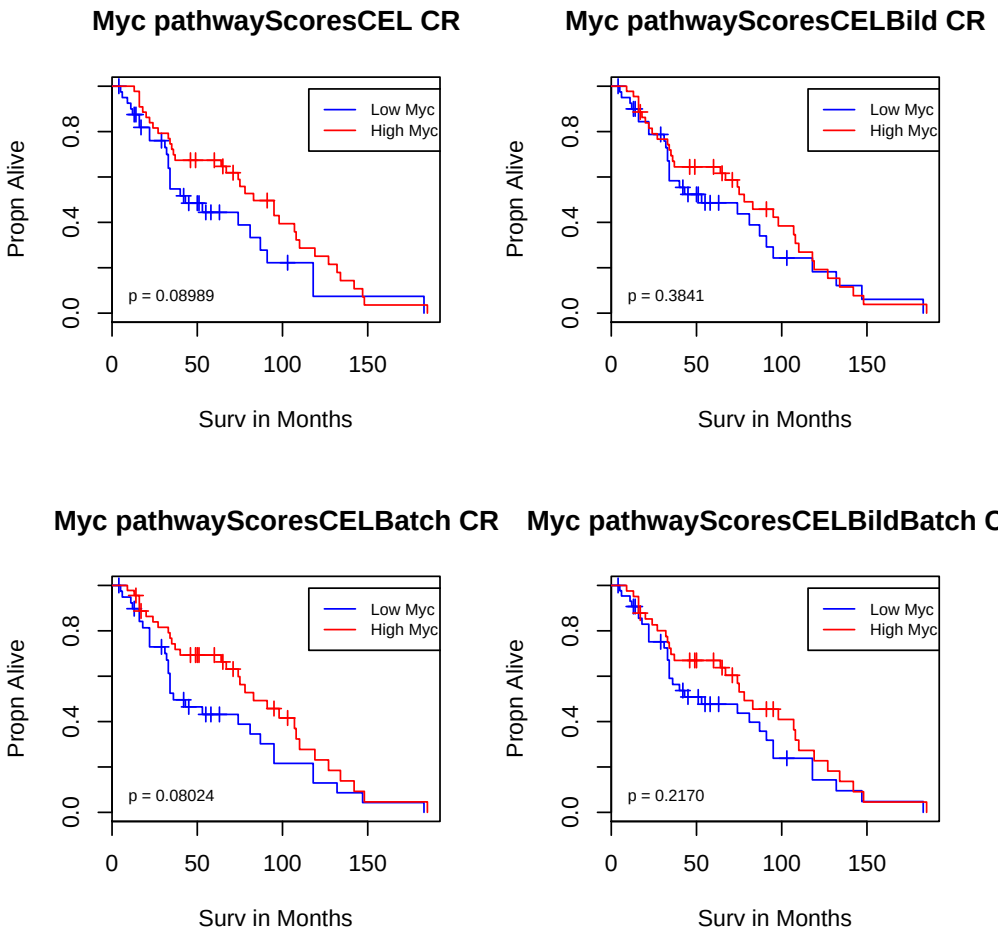
```
                               NR          CR          All
pathwayScoresCEL          0.9052111 0.02024796 0.023708691
pathwayScoresCELBild      0.3074254 0.21991678 0.394984826
pathwayScoresCELBatch     0.6486185 0.04779531 0.027359927
pathwayScoresCELBildBatch 0.2039310 0.13517119 0.618674675
pathwayScoresXLS          0.3169511 0.03968798 0.008815240
pathwayScoresXLSBild      0.4782334 0.00677554 0.005145462
pathwayScoresXLSBatch     0.3169511 0.12388712 0.029596052
pathwayScoresXLSBildBatch 0.3311996 0.09709565 0.027990100
```

For Myc, there is no separation of the NRs. Separation in the other groups is marginal at best, and shows most clearly when the Bild censoring is used. The best separation is seen here when the XLS quantifications are used, but this is not what we want to see.

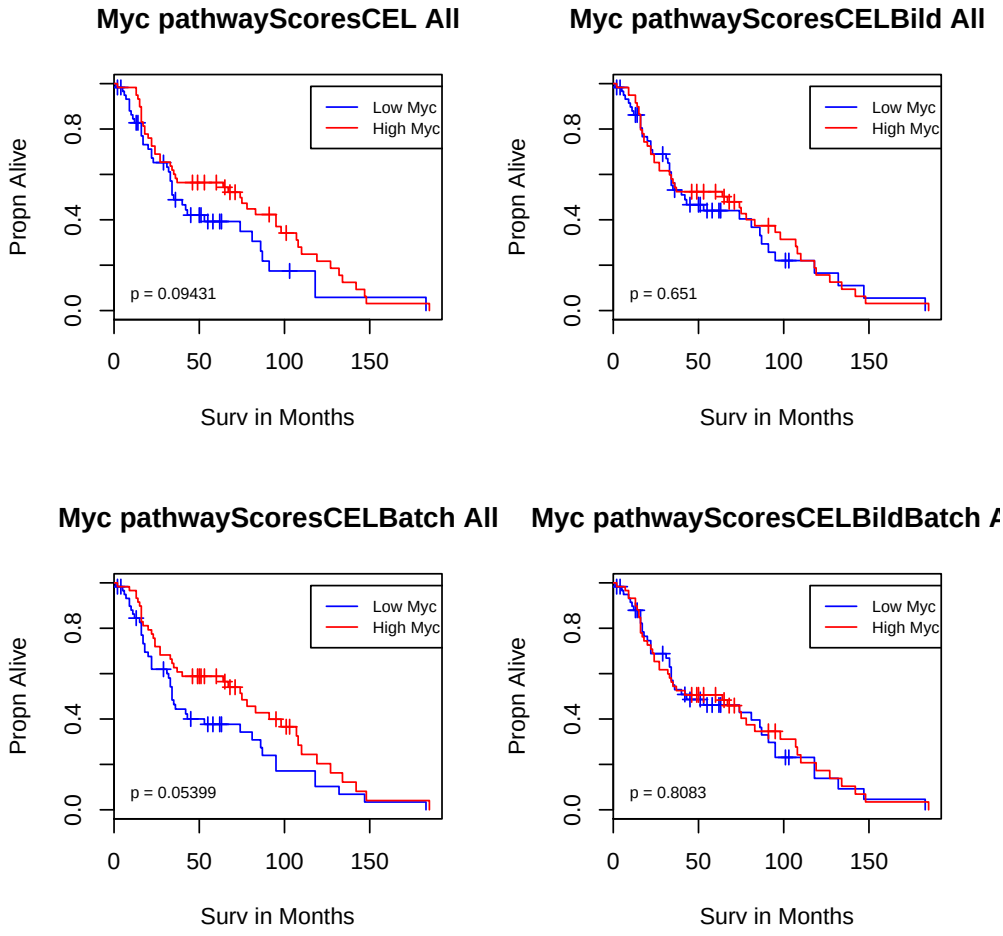
```
> scoreList <- pathwayScores[1:4]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```



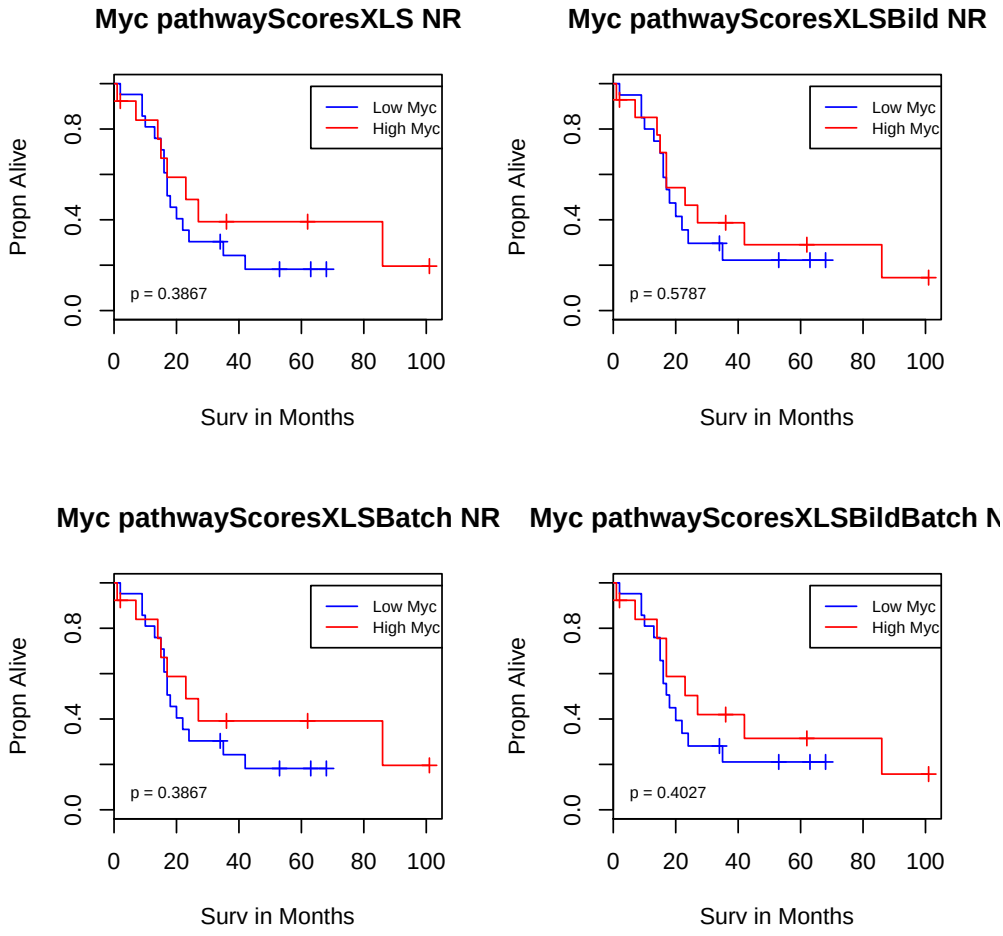
```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```



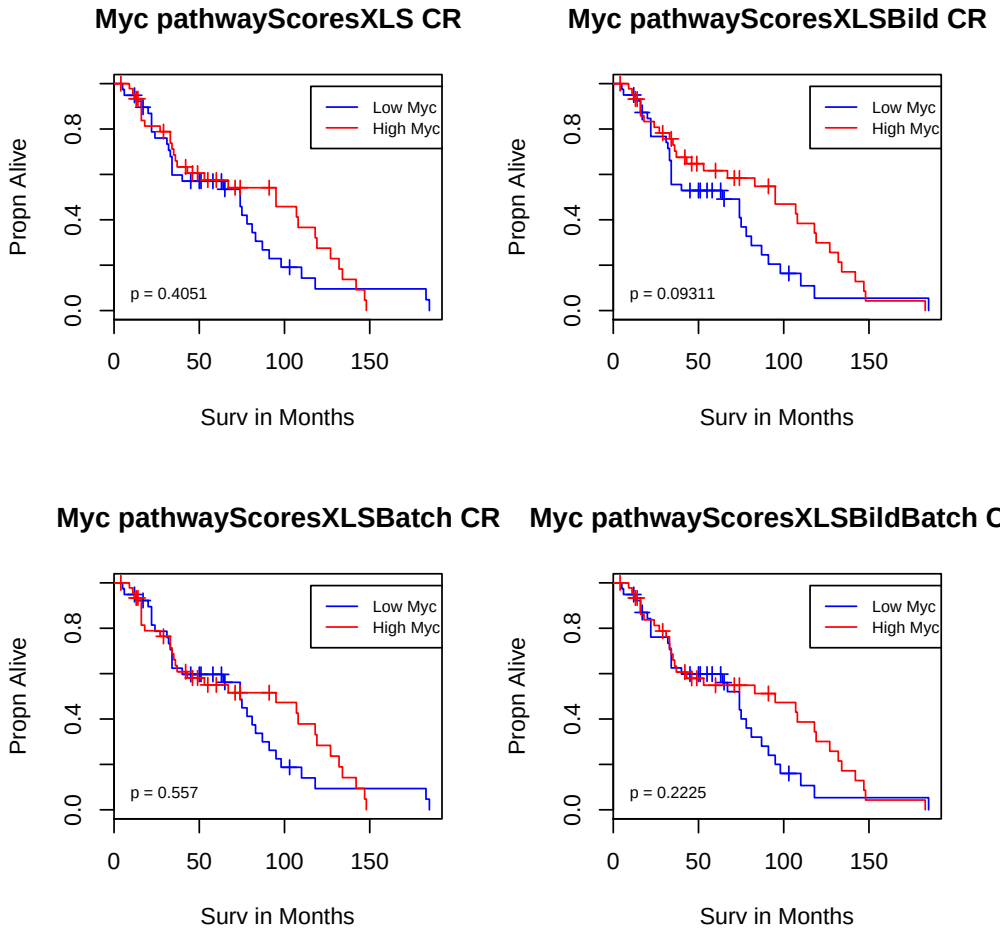

```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



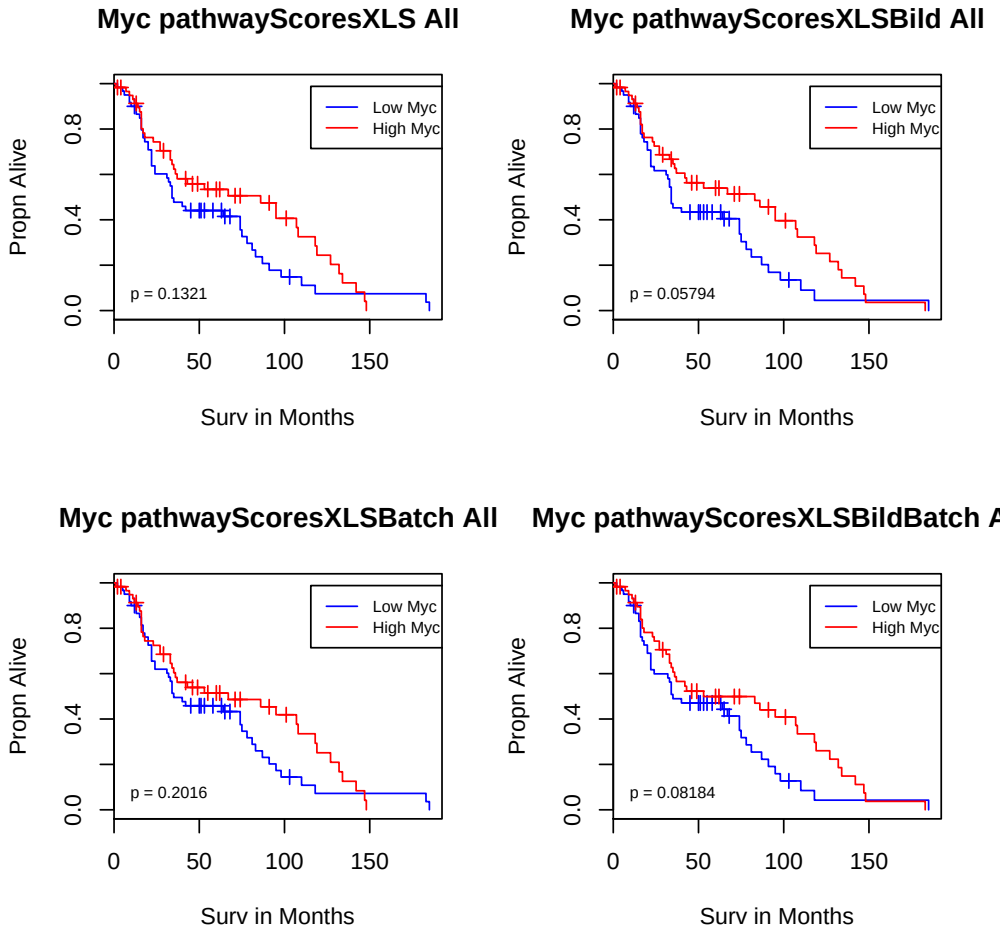

```
> scoreList <- pathwayScores[5:8]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



7 Ras

```
> pathway <- "Ras"
> pathwaySurvPVals[[pathway]]
```

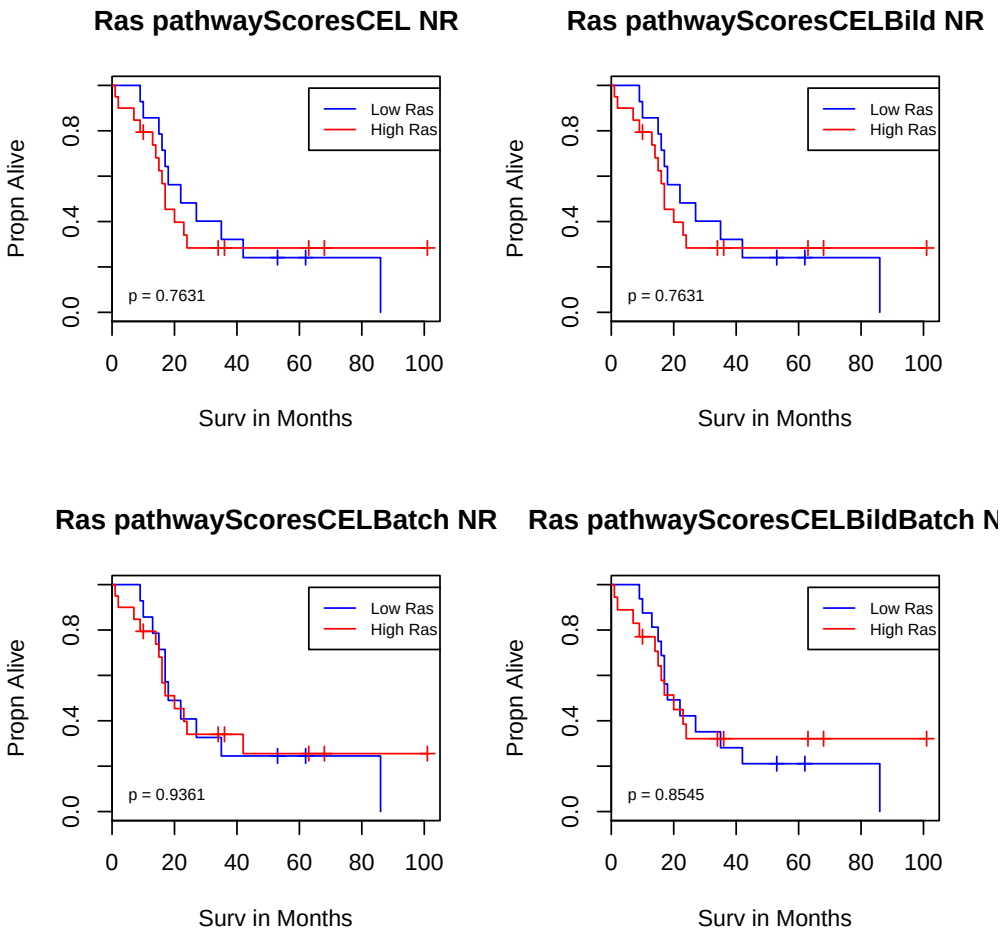
```
                                NR         CR        All
pathwayScoresCEL          0.7630975 0.16574956 0.1425585
pathwayScoresCELBild      0.7630975 0.13725656 0.1204157
pathwayScoresCELBatch     0.9361496 0.39074667 0.3501027
pathwayScoresCELBildBatch 0.8544592 0.05649931 0.1420359
pathwayScoresXLS          0.9248254 0.65236409 0.8836579
pathwayScoresXLSBild      0.9248254 0.41530276 0.6524673
pathwayScoresXLSBatch     0.9569750 0.81258700 0.4390627
pathwayScoresXLSBildBatch 0.9493136 0.68129485 0.9273739
```

```
> pathwaySurvBildPVals[[pathway]]
```

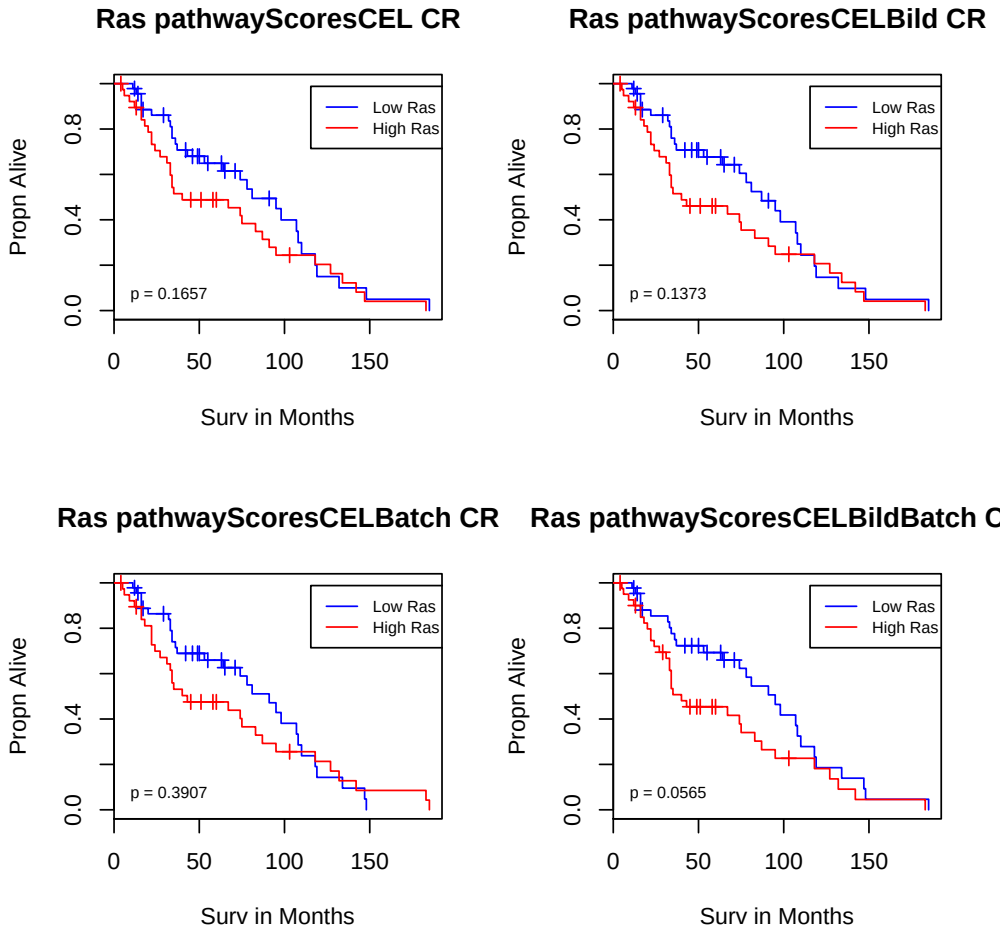
```
                               NR         CR       All
pathwayScoresCEL          0.6281619 0.28558150 0.1824615
pathwayScoresCELBild      0.6281619 0.24604720 0.1580798
pathwayScoresCELBatch     0.9117217 0.21286671 0.1658357
pathwayScoresCELBildBatch 0.9698491 0.06349225 0.1331219
pathwayScoresXLS          0.9418516 0.91386862 0.7872557
pathwayScoresXLSBild      0.9418516 0.60836740 0.9575235
pathwayScoresXLSBatch     0.8469949 0.45417699 0.1817082
pathwayScoresXLSBildBatch 0.8304671 0.92947017 0.6104710
```

We see no real stratification with Ras.

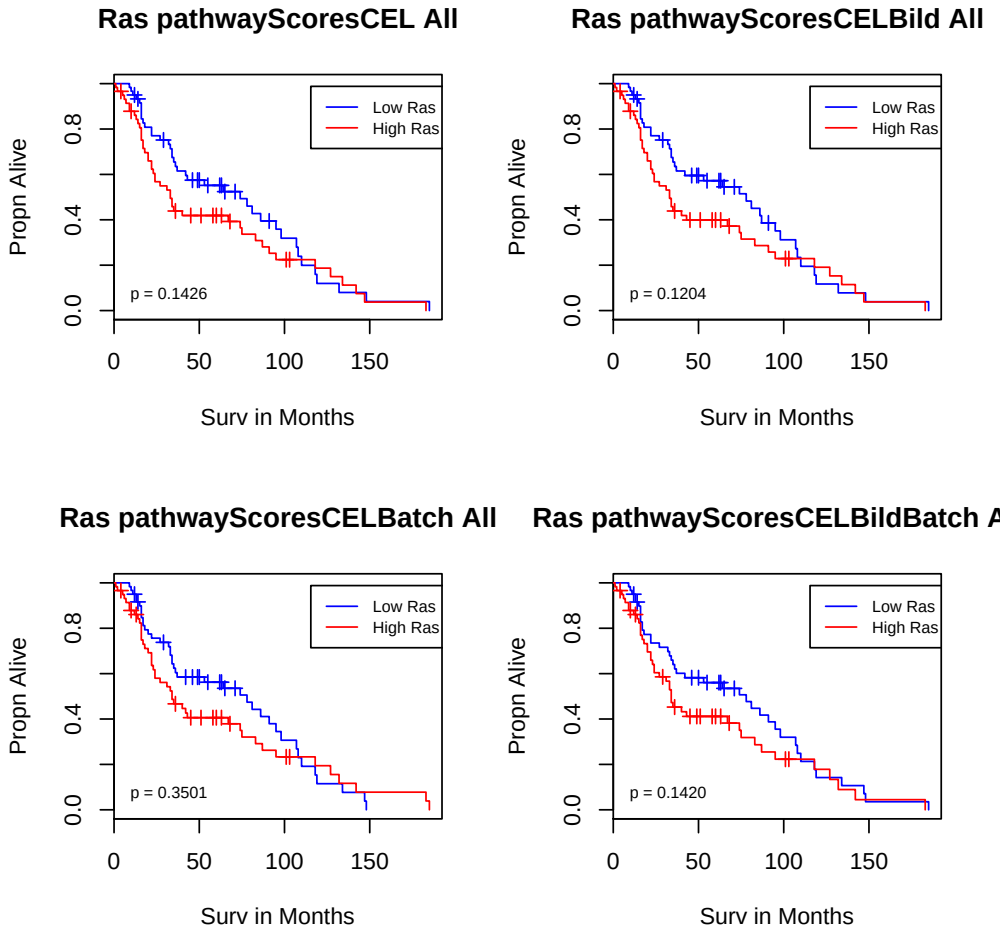
```
> scoreList <- pathwayScores[1:4]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```



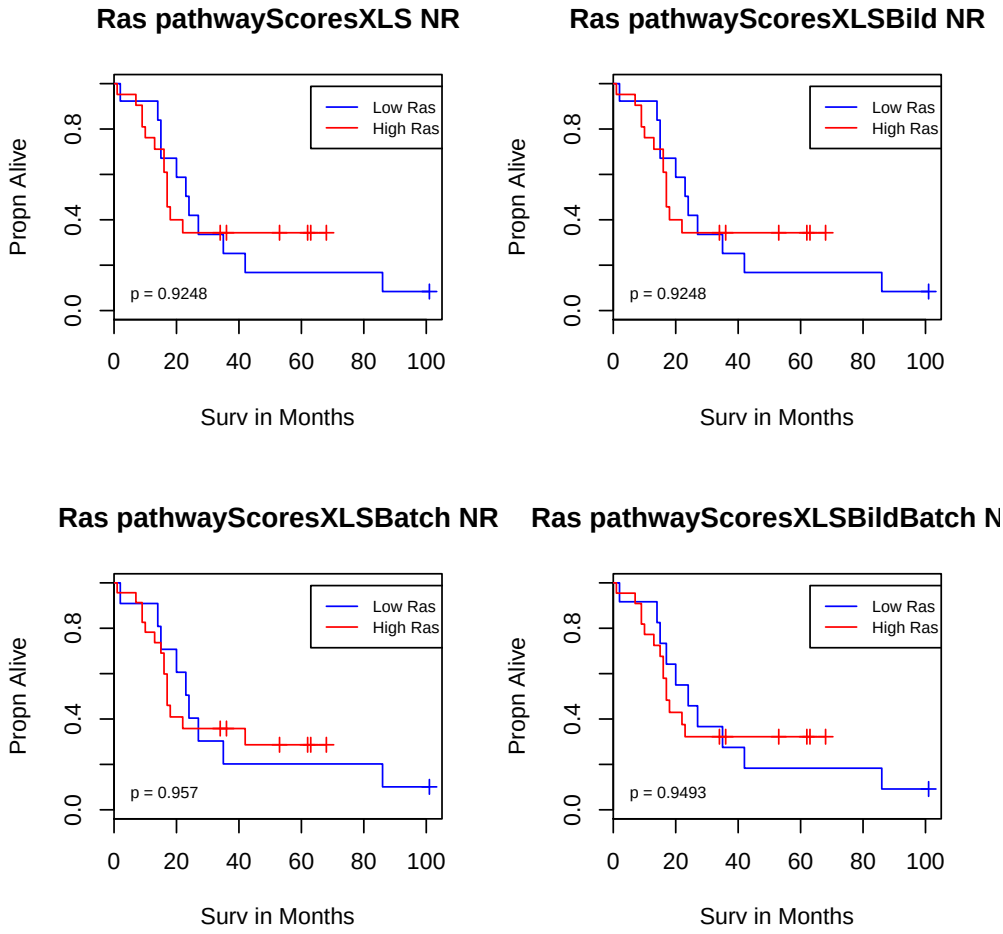

```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```



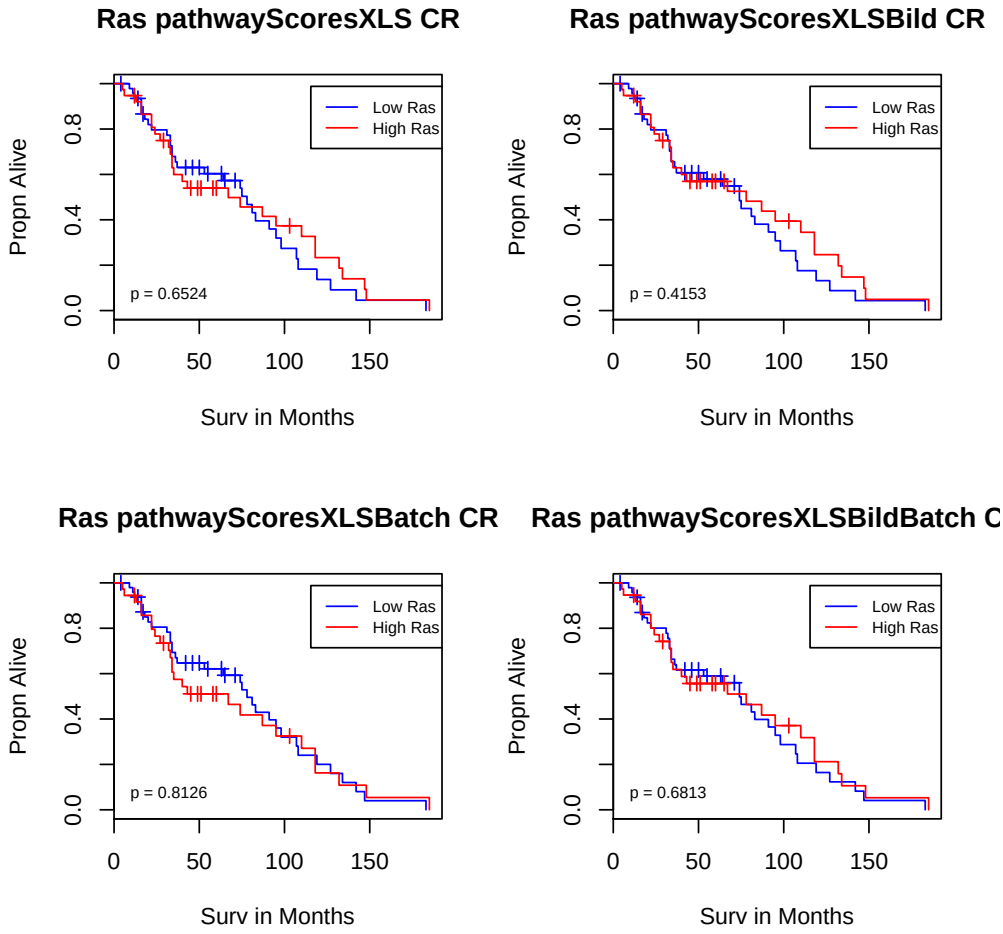

```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



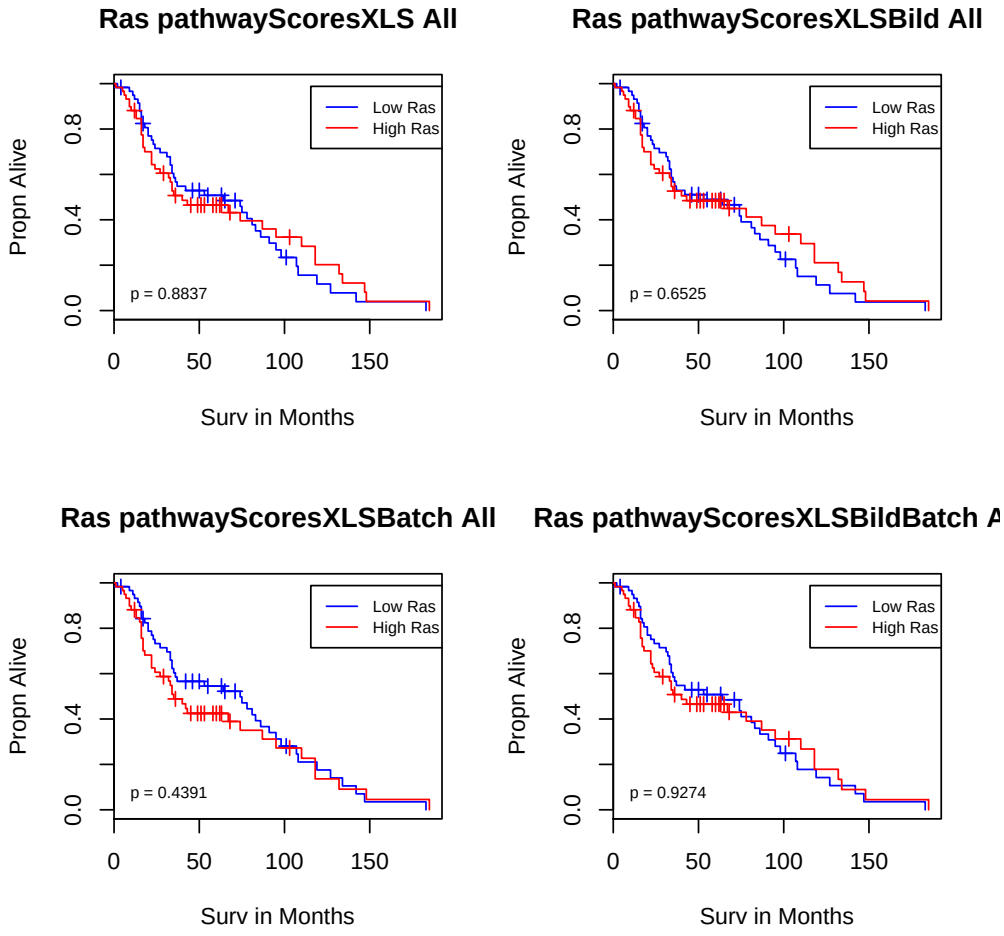

```
> scoreList <- pathwayScores[5:8]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



8 Bcat

```
> pathway <- "Bcat"
> pathwaySurvPVals[[pathway]]
```

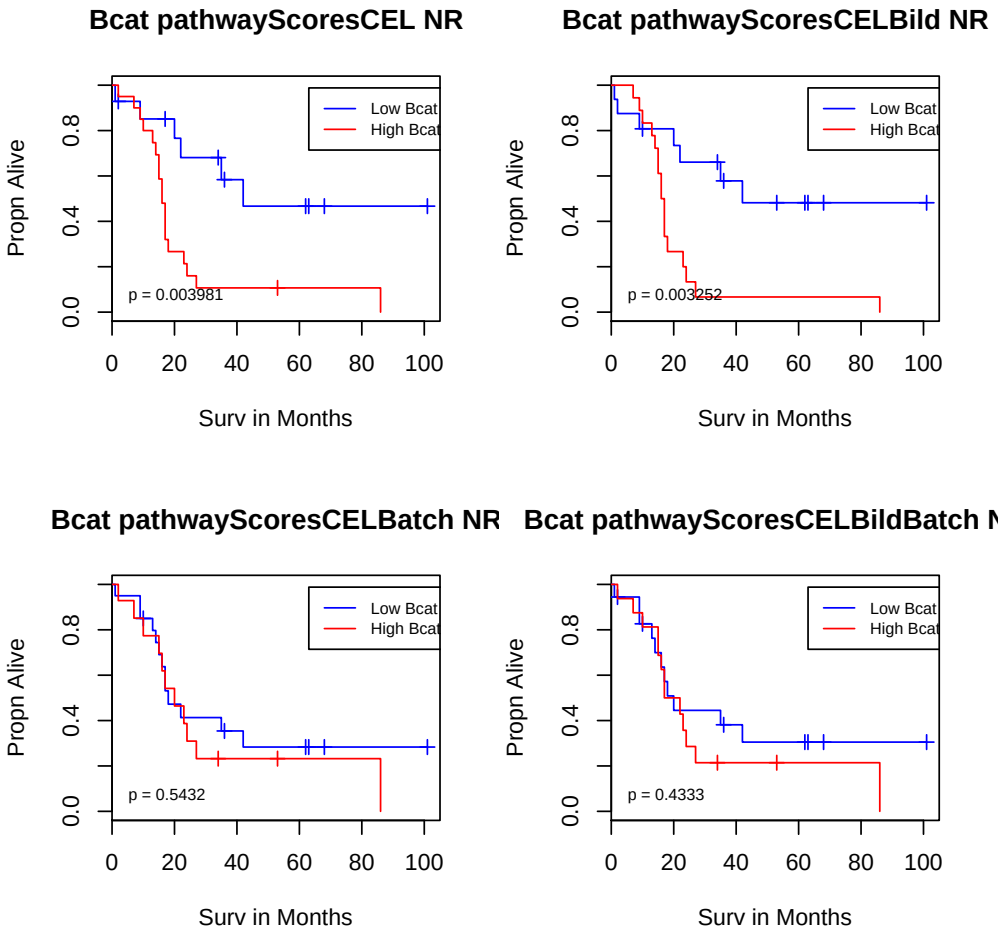
```
                                NR         CR          All
pathwayScoresCEL          0.003980742 0.01542867 0.0001813096
pathwayScoresCELBild      0.003251997 0.06433352 0.0020041972
pathwayScoresCELBatch     0.543153326 0.58358534 0.6104797009
pathwayScoresCELBildBatch 0.433291994 0.60916359 0.3838894922
pathwayScoresXLS          0.168393055 0.27001625 0.1092547645
pathwayScoresXLSBild      0.202316004 0.80452938 0.4698100498
pathwayScoresXLSBatch     0.278177417 0.32475636 0.3017883781
pathwayScoresXLSBildBatch 0.650600928 0.58176264 0.7595134122
```

```
> pathwaySurvBildPVals[[pathway]]
```

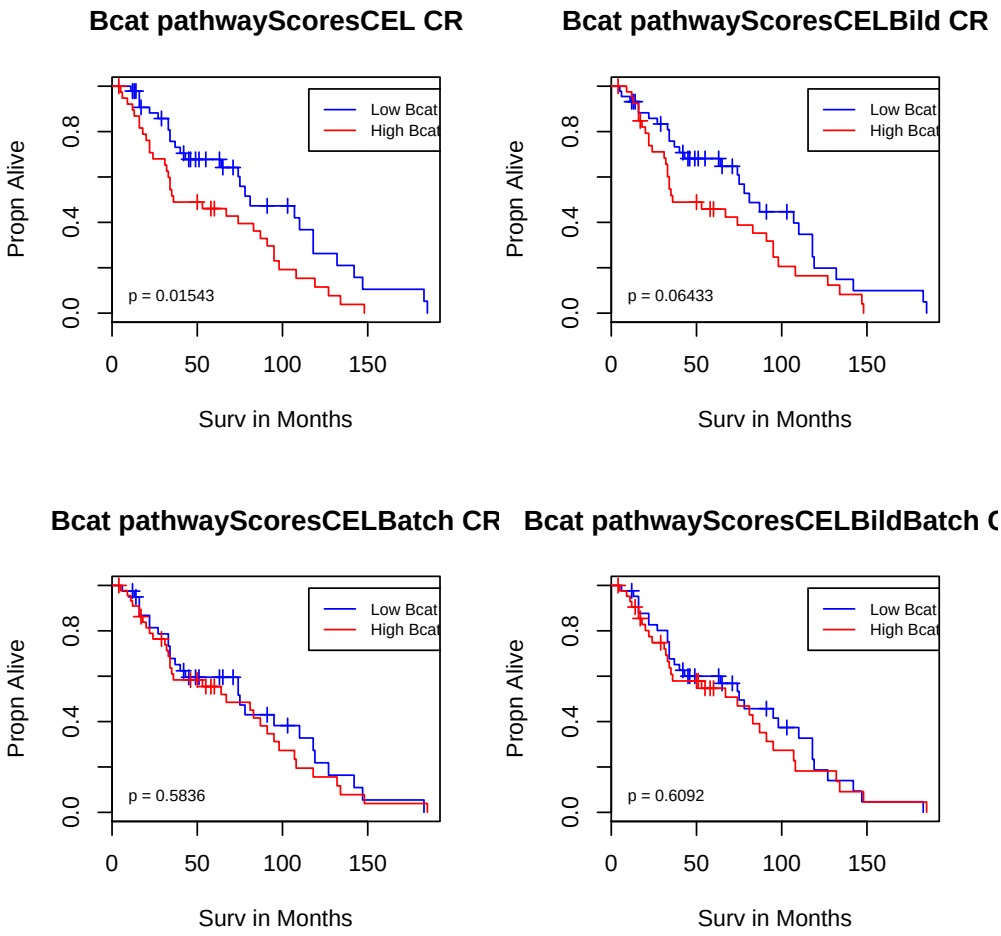
```
                                NR         CR          All
pathwayScoresCEL          0.002809777 0.05425851 0.0004882958
pathwayScoresCELBild      0.008402855 0.23109586 0.0136590912
pathwayScoresCELBatch     0.665222482 0.44093265 0.5933816301
pathwayScoresCELBildBatch 0.573606595 0.55838617 0.4178724976
pathwayScoresXLS          0.254978482 0.84676724 0.4758556767
pathwayScoresXLSBild      0.299593231 0.63599297 0.9988979652
pathwayScoresXLSBatch     0.372168008 0.89862674 0.8689081841
pathwayScoresXLSBildBatch 0.781995419 0.67094882 0.5023623651
```

The story with β-catenin is very similar to the story with Src – we see some separation by score within the NRs and in the group as a whole, but this separation is confounded with run batch and disappears when we correct for the latter.

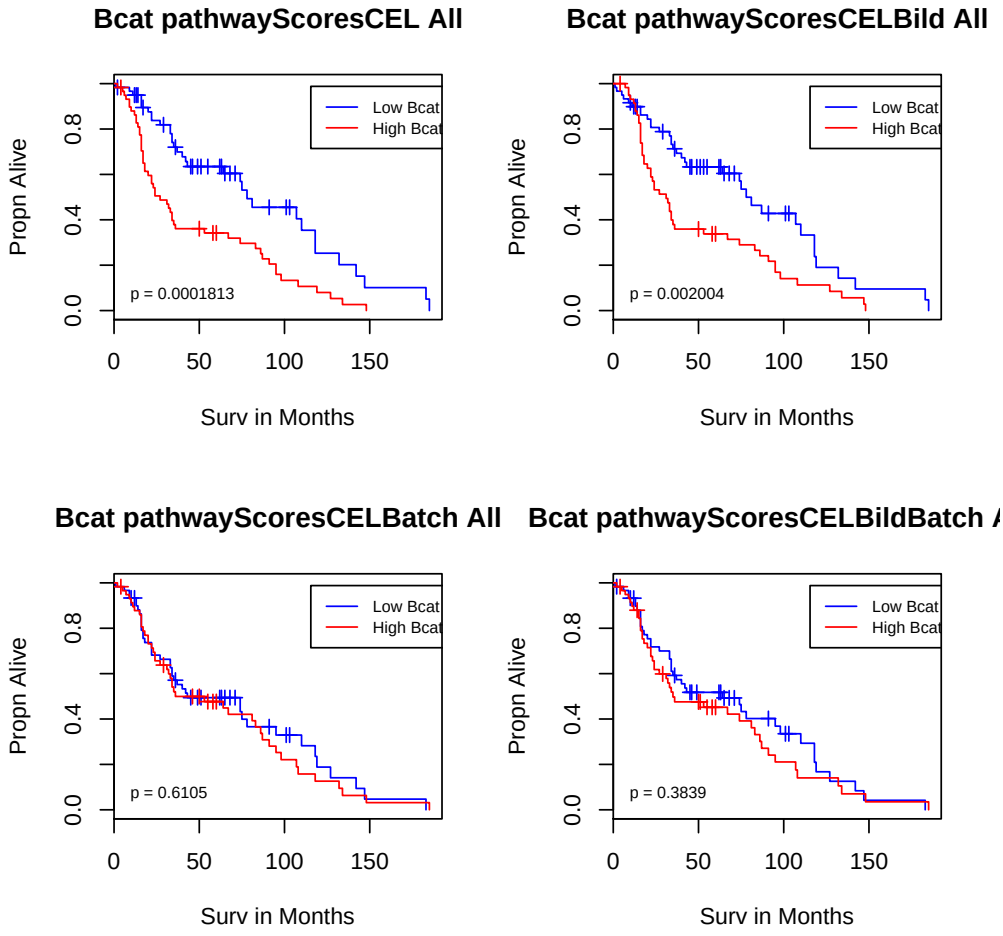
```
> scoreList <- pathwayScores[1:4]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```



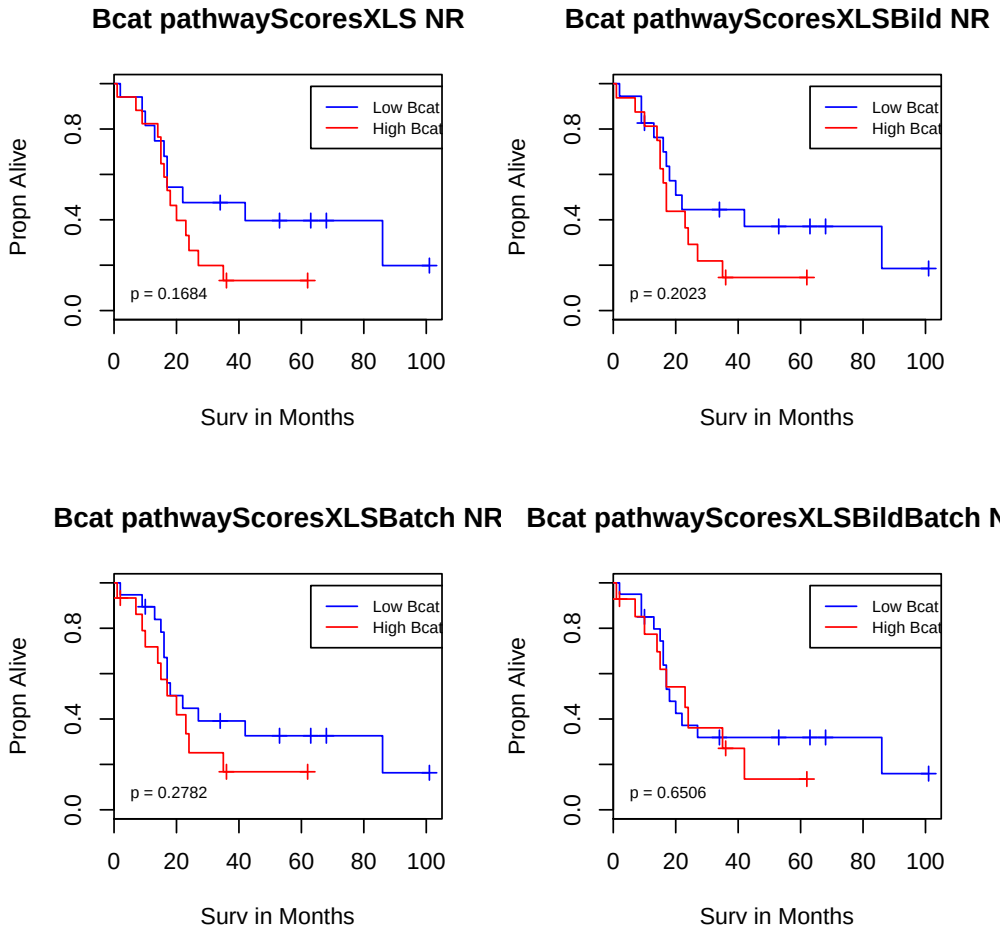
```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```



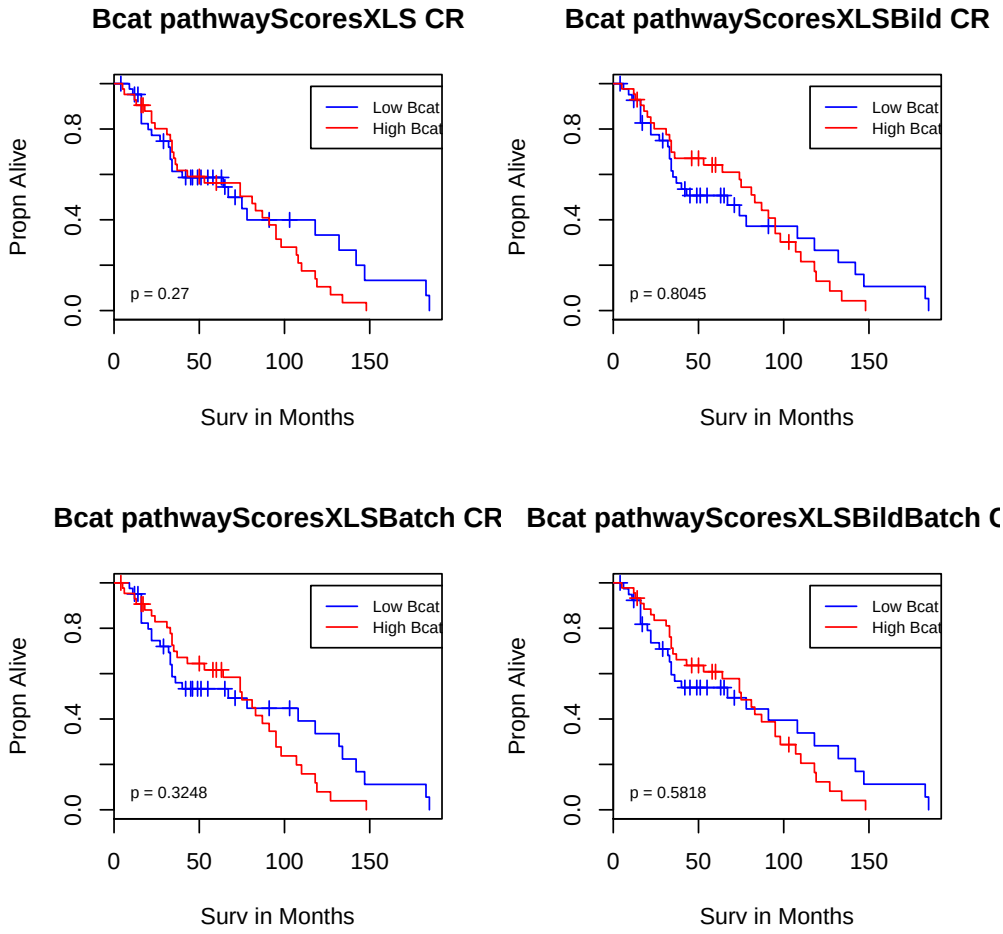

```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



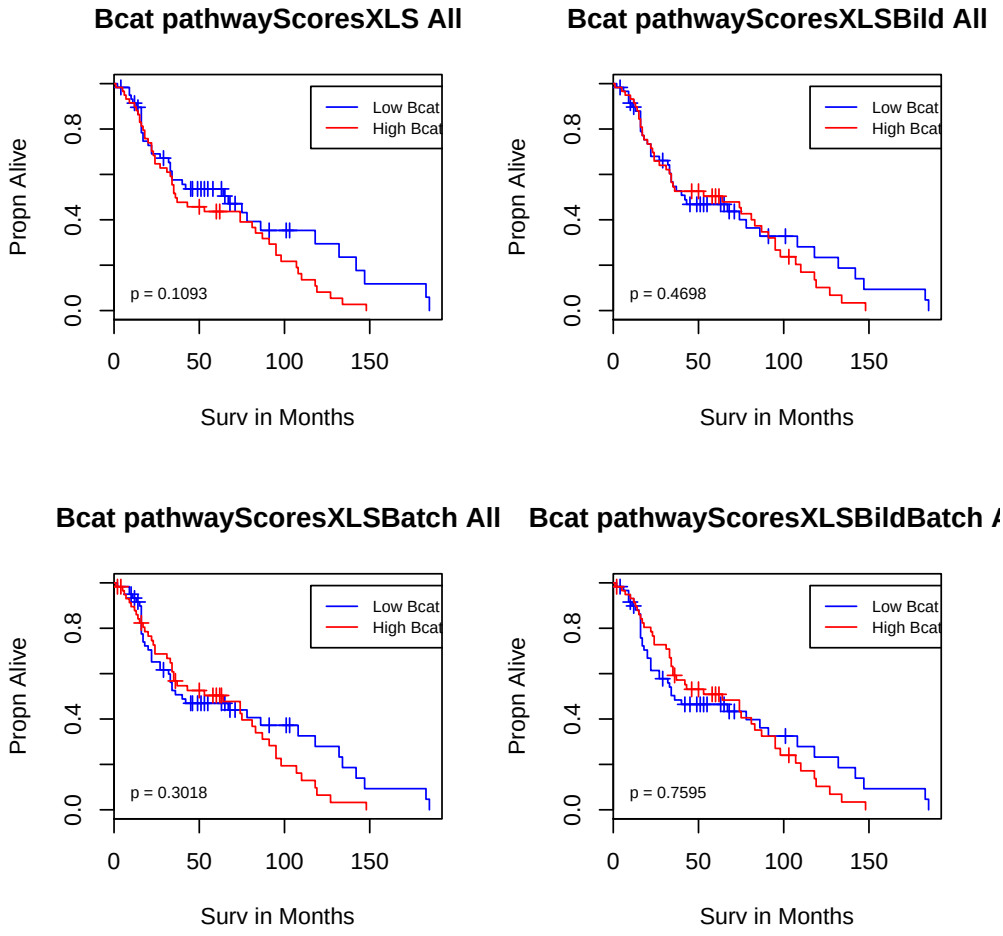

```
> scoreList <- pathwayScores[5:8]
> patientSubgroup <- "NR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "CR"
> <<plotKMQuartet>>
```




```
> patientSubgroup <- "All"
> <<plotKMQuartet>>
```



9 A Closer Look at Src

The KM-plots shown in Dressman et al allow us to infer the high/low status of the Src NR samples. By looking at how well these agree with our own scores, we can assess how well these map. Looking at the KM plots for the CRs clearly shows that the longest times are being treated as deaths, so only the Dressman et al censoring values are used here.

```
> sort(survDressman[clinicalInfo$Response == "NR"])

 [1]   1    2+   2    7    9    9    10+  10   13   14   15   15   16   16   17
[16]  17   17   17+  18   20   22   23   24   27   34+  35   36+  42   53+  62+
[31]  63+  68+  86   101+

> srcHighLowNR <- c("H", "L", "L", "H", "H", "H", "H", "H", "H",
+     "H", "L", "H", "H", "H", "H", "H", "H", "L", "L", "H", "L",
+     "H", "L", "L", "L", "L", "L", "L", "L", "L", "L", "H", "H",
```

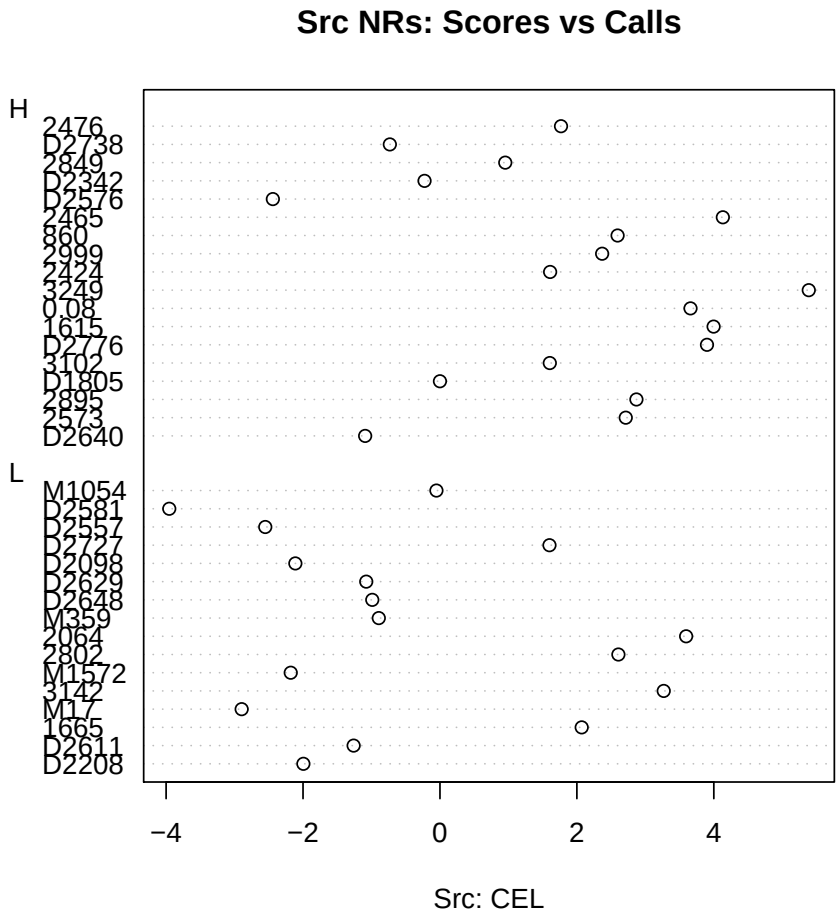
```
+     "L")
> names(srcHighLowNR) <- rownames(sort(survDressman[clinicalInfo$Response ==
+     "NR"]))
> srcHighLowNR

D2640 D2208 D2611  2573  2895 D1805  3102 D2776  1615  0.08  1665  3249  2424
  "H"   "L"   "L"   "H"   "H"   "H"   "H"   "H"   "H"   "H"   "L"   "H"   "H"
 2999   860  2465 D2576   M17  3142 D2342 M1572  2849  2802  2064  M359 D2648
  "H"   "H"   "H"   "H"   "L"   "L"   "H"   "L"   "H"   "L"   "L"   "L"   "L"
D2629 D2098 D2727 D2557 D2581 D2738  2476 M1054
  "L"   "L"   "L"   "L"   "L"   "H"   "H"   "L"
```

There is some uncertainty in the above allotment; one of the 15 month survival times is associated with a low Src score and the other with a high Src score. Thus, the placement of 1665 and 3249 may be reversed.

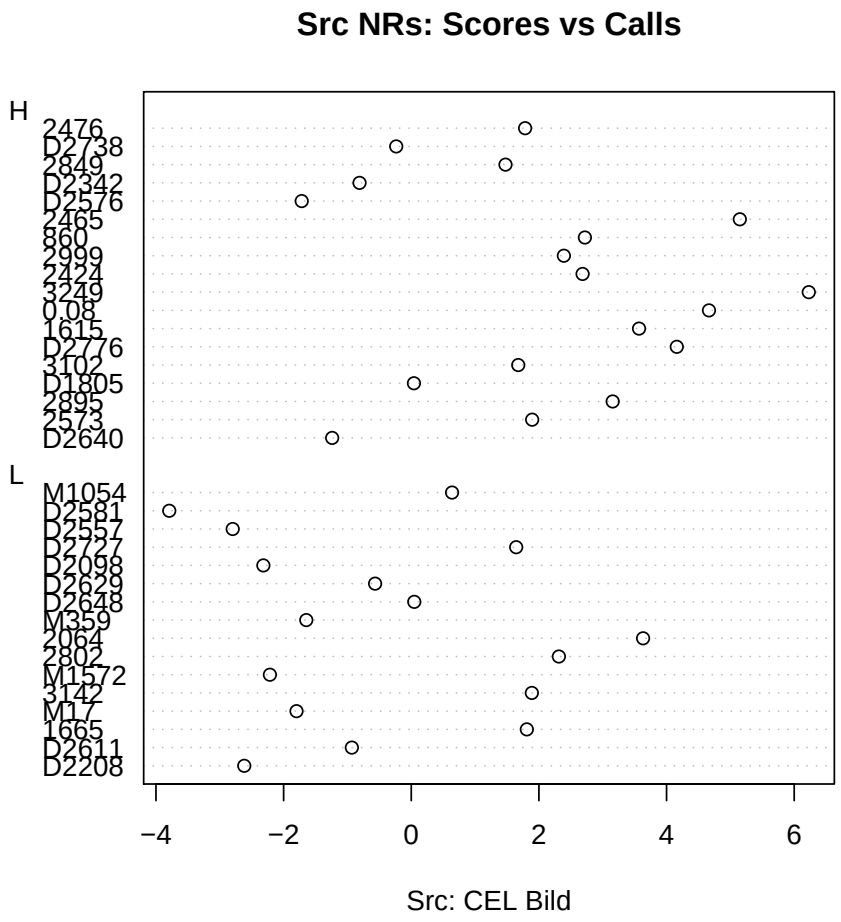
The plots below check how well the classes dictated by our scores agree with those actually used. If they agree well, points in the top half of the plot should be grouped at the right, and those in the bottom half grouped at the left.

This does not appear to hold for any of the scores we compute.

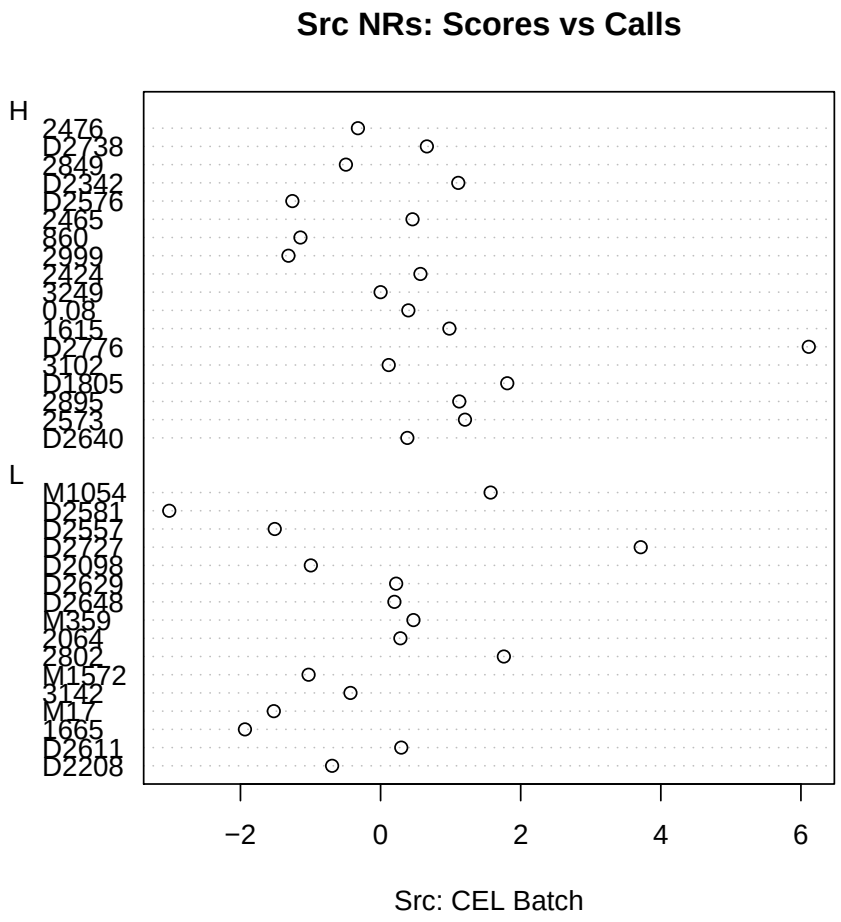
```
> temp <- pathwayScoresCEL$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: CEL")
```



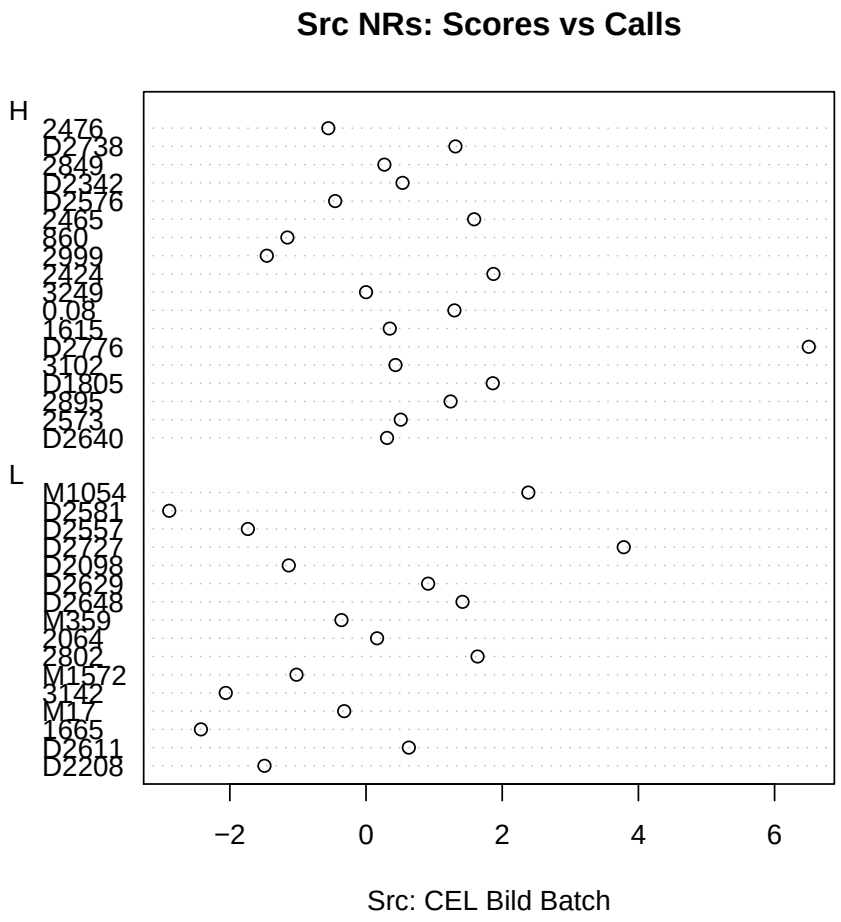

```
> temp <- pathwayScoresCELBild$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: CEL Bild")
```



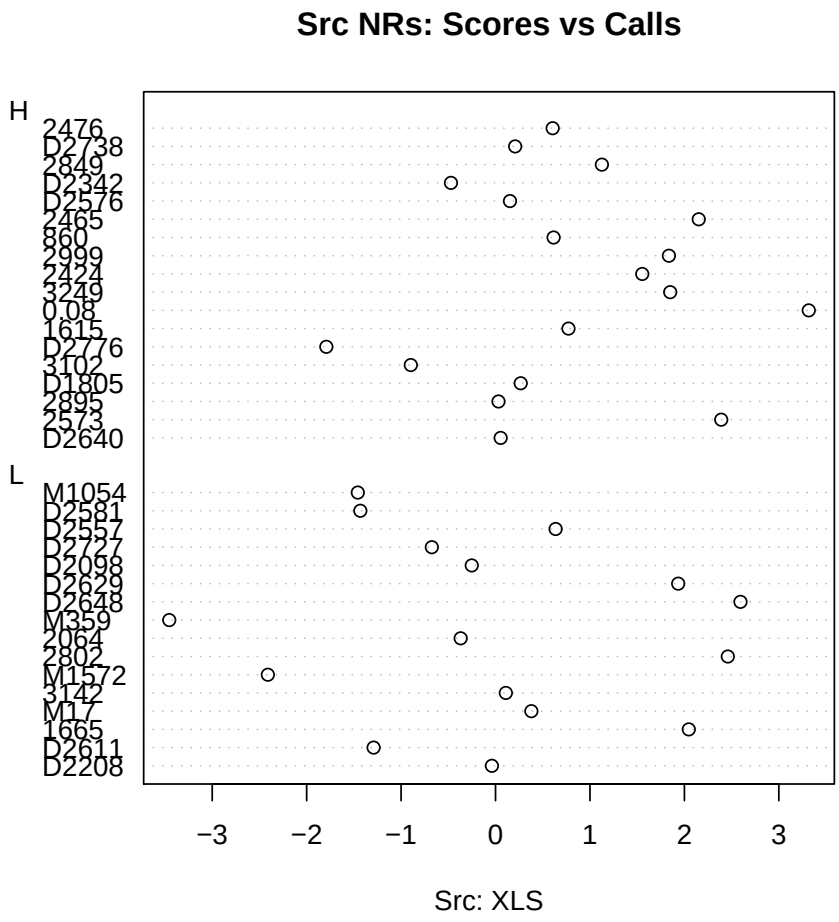

```
> temp <- pathwayScoresCELBatch$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: CEL Batch")
```



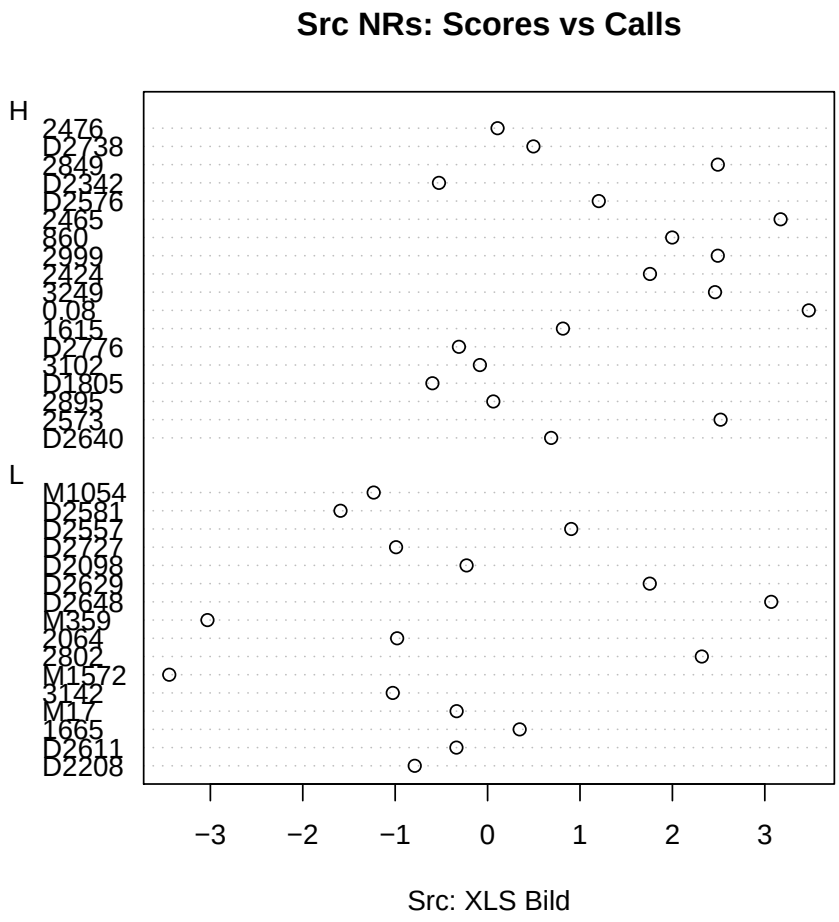

```
> temp <- pathwayScoresCELBildBatch$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: CEL Bild Batch")
```



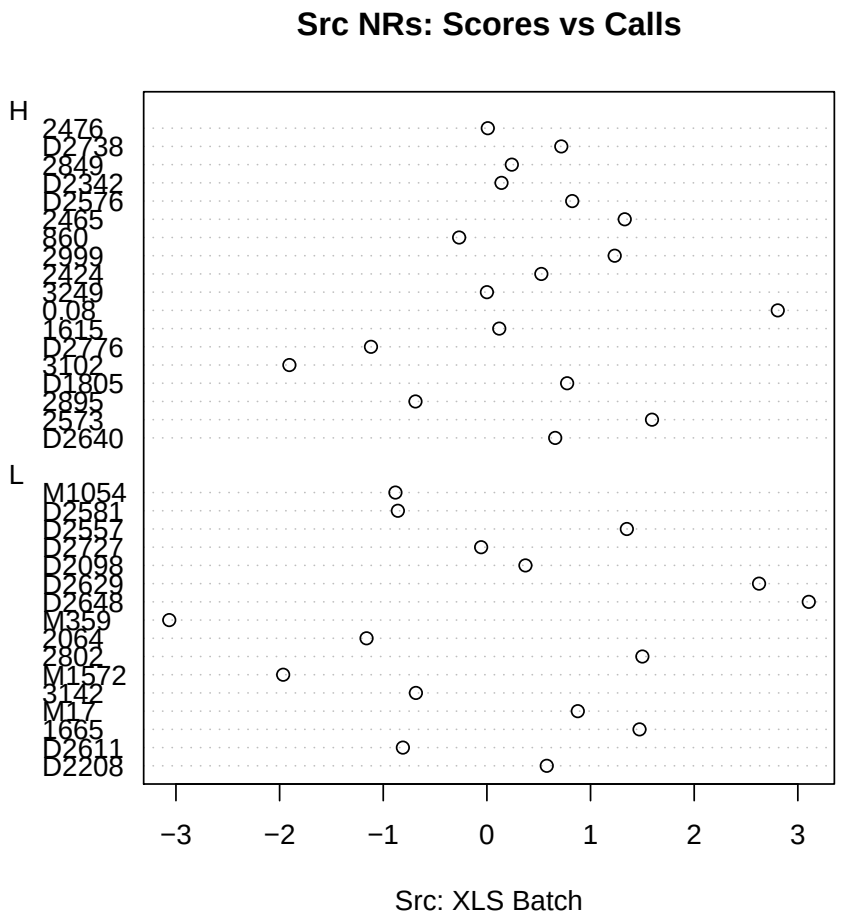

```
> temp <- pathwayScoresXLS$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: XLS")
```



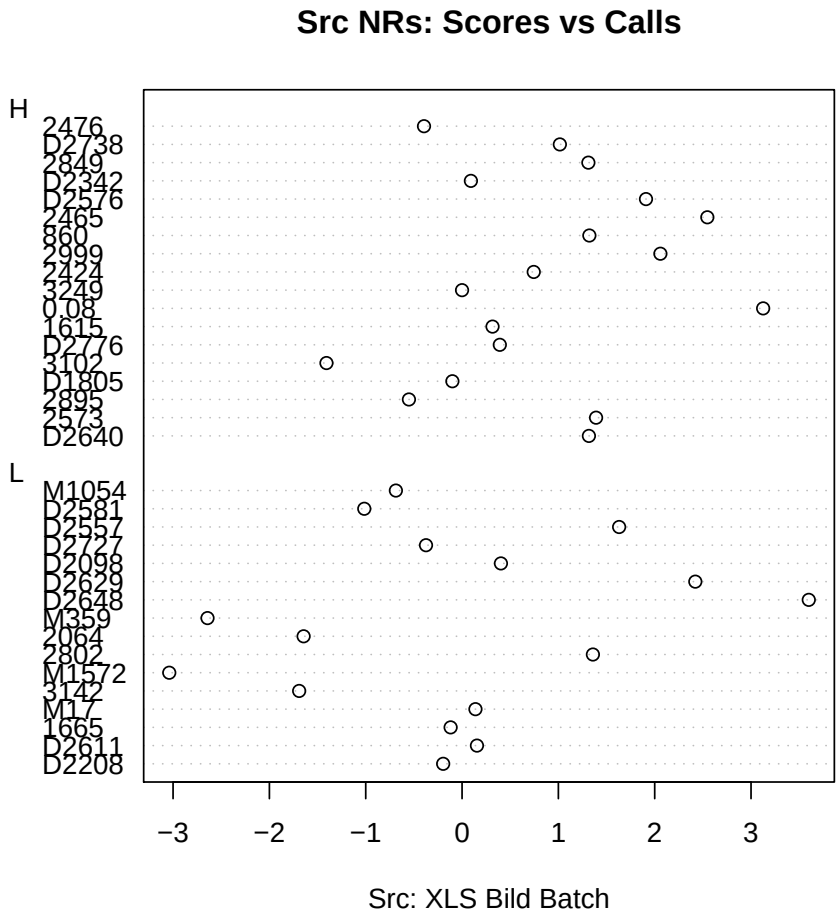

```
> temp <- pathwayScoresXLSBild$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: XLS Bild")
```




```
> temp <- pathwayScoresXLSBatch$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: XLS Batch")
```




```
> temp <- pathwayScoresXLSBildBatch$Src[names(srcHighLowNR)]
> dotchart(temp, groups = as.factor(srcHighLowNR), main = "Src NRs: Scores vs Calls",
+     xlab = "Src: XLS Bild Batch")
```



In all of the above plots, there is substantial overlap between the scores assigned to the NR samples that Dressman et al designate as high and low. Their scores clearly do not match ours.

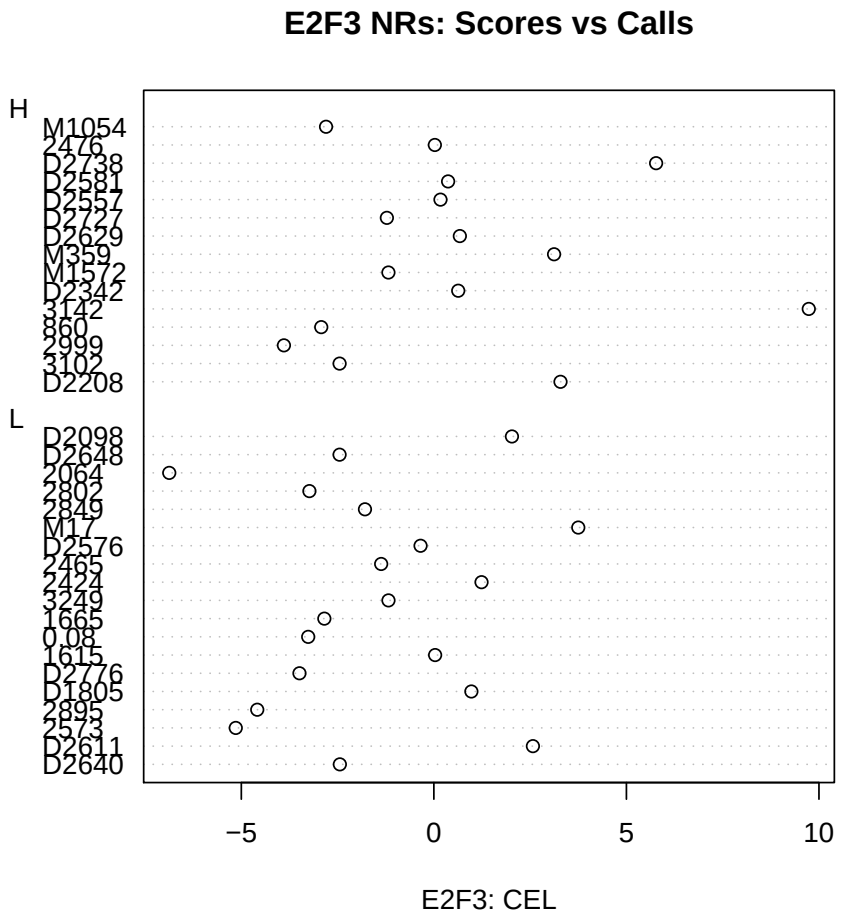
10 A Closer Look at E2F3

```
> sort(survDressman[clinicalInfo$Response == "NR"])

 [1]   1    2+   2    7    9    9   10+  10   13   14   15   15   16   16   17
[16]  17   17   17+  18   20   22   23   24   27   34+  35   36+  42   53+  62+
[31]  63+  68+  86  101+

> e2f3HighLowNR <- c("L", "H", "L", "L", "L", "L", "H", "L", "L",
+     "L", "L", "L", "L", "H", "H", "L", "L", "L", "H", "H", "H",
+     "L", "L", "L", "H", "L", "H", "L", "H", "H", "H", "H", "H",
+     "H")
> names(e2f3HighLowNR) <- rownames(sort(survDressman[clinicalInfo$Response ==
```

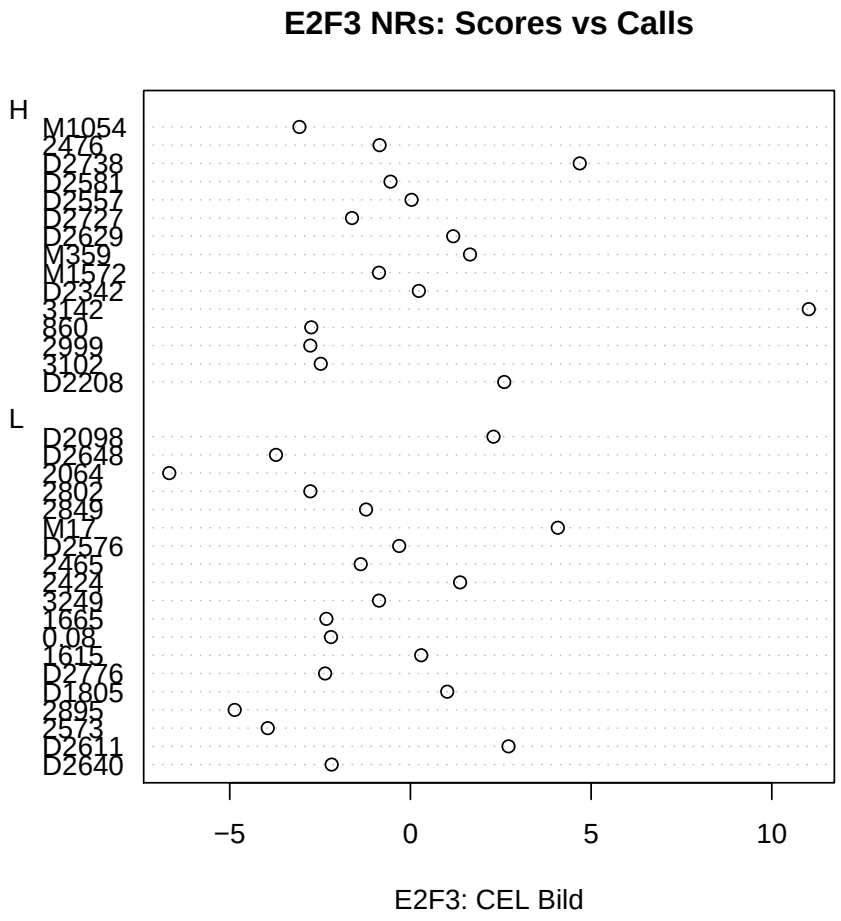
```
+     "NR"]))
> e2f3HighLowNR

D2640 D2208 D2611  2573  2895 D1805  3102 D2776  1615  0.08  1665  3249  2424
  "L"   "H"   "L"   "L"   "L"   "L"   "H"   "L"   "L"   "L"   "L"   "L"   "L"
 2999   860  2465 D2576   M17  3142 D2342 M1572  2849  2802  2064  M359 D2648
  "H"   "H"   "L"   "L"   "L"   "H"   "H"   "H"   "L"   "L"   "L"   "H"   "L"
D2629 D2098 D2727 D2557 D2581 D2738  2476 M1054
  "H"   "L"   "H"   "H"   "H"   "H"   "H"   "H"
```

As with Src, there is some ambiguity: one of the two 16s is High, the other Low, and one of the three 17s is High, the other two low. Just as a check, the table we assembled is given below: 1 L, 2+ H, 2 L, 7 L, 9 L, 9 L, 10+ H, 10 L, 13 L, 14 L, 15 L, 15 L, 16 L –, 16 H –, 17 H –, 17 L –, 17 L –, 17+ L, 18 H, 20 H, 22 H, 23 L, 24 L, 27 L, 34+ H, 35 L, 36+ H, 42 L, 53+ H, 62+ H, 63+ H, 68+ H, 86 H, 101+ H.

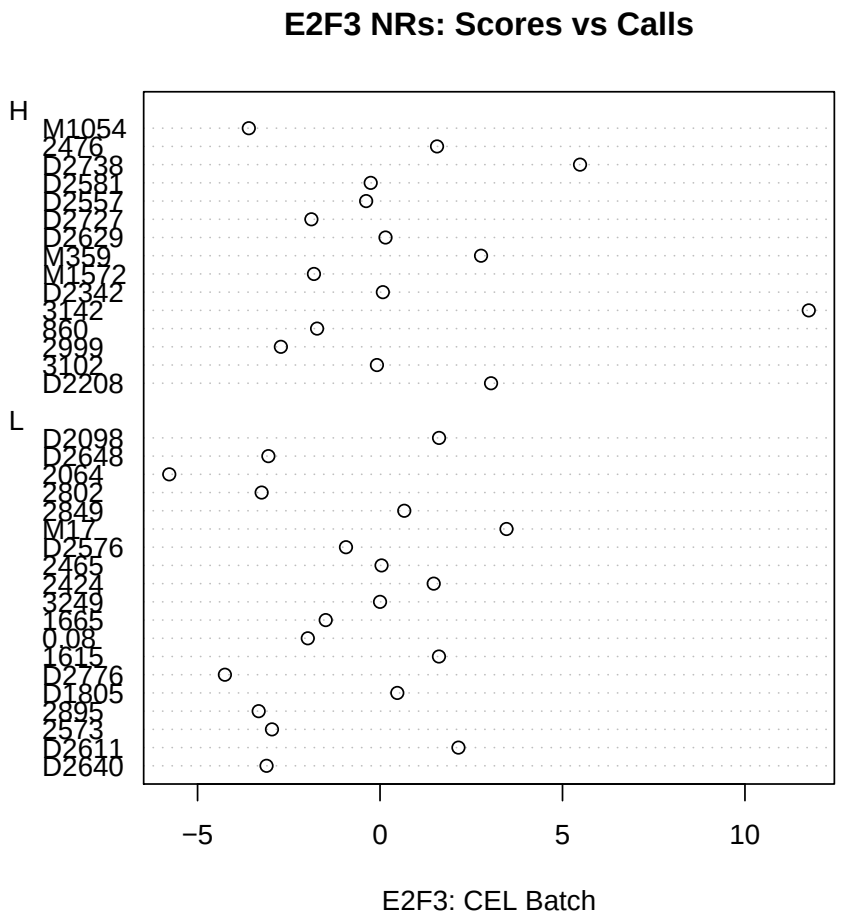
```
> temp <- pathwayScoresCEL$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: CEL")
```



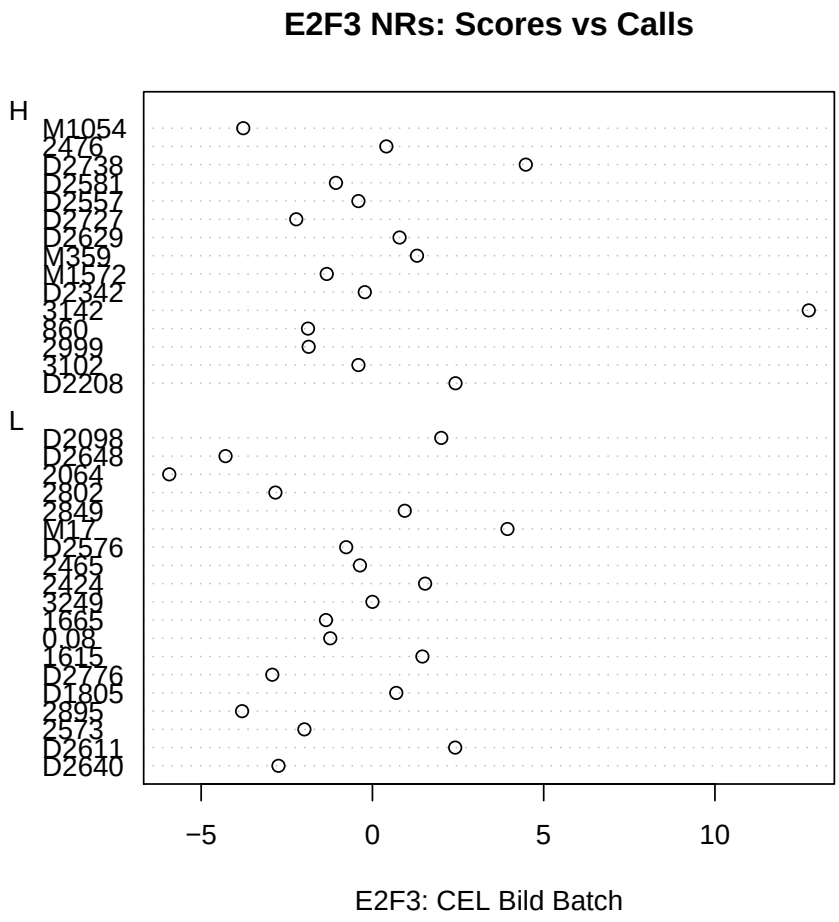
```
> temp <- pathwayScoresCELBild$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: CEL Bild")
```



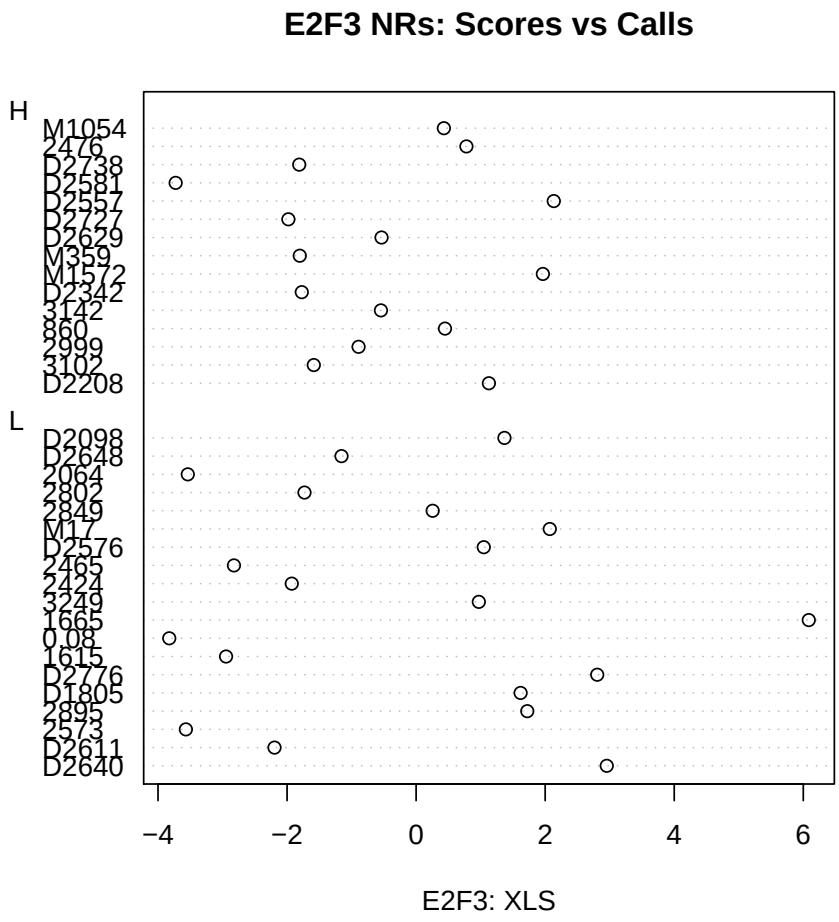

```
> temp <- pathwayScoresCELBatch$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: CEL Batch")
```



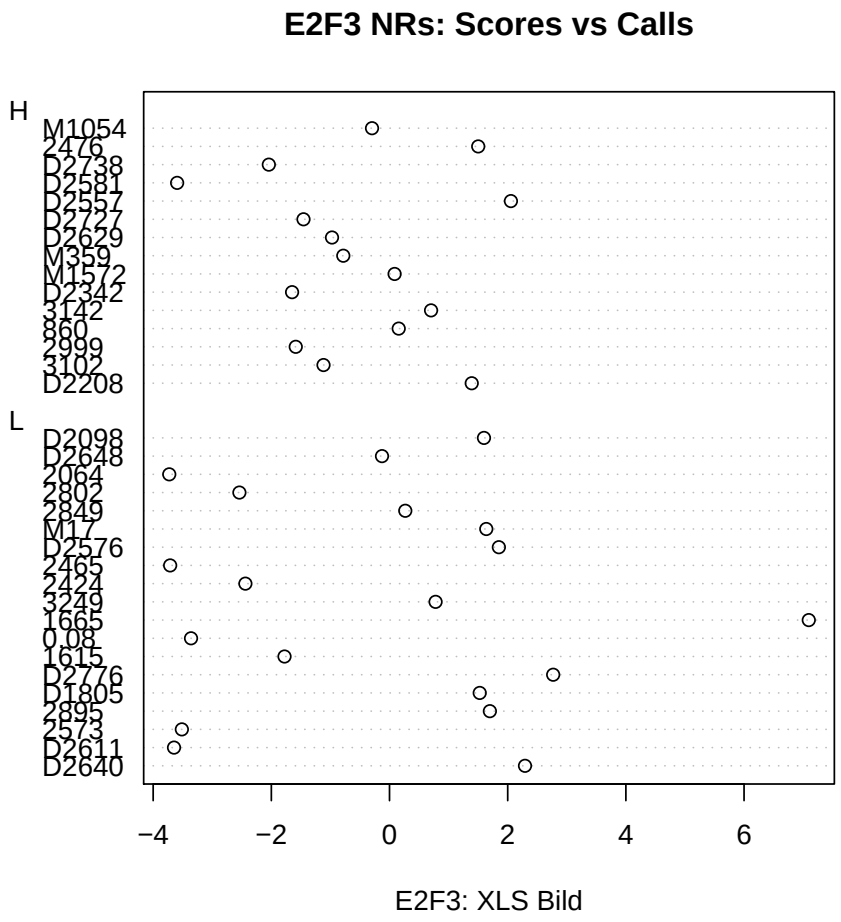

```
> temp <- pathwayScoresCELBildBatch$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: CEL Bild Batch")
```



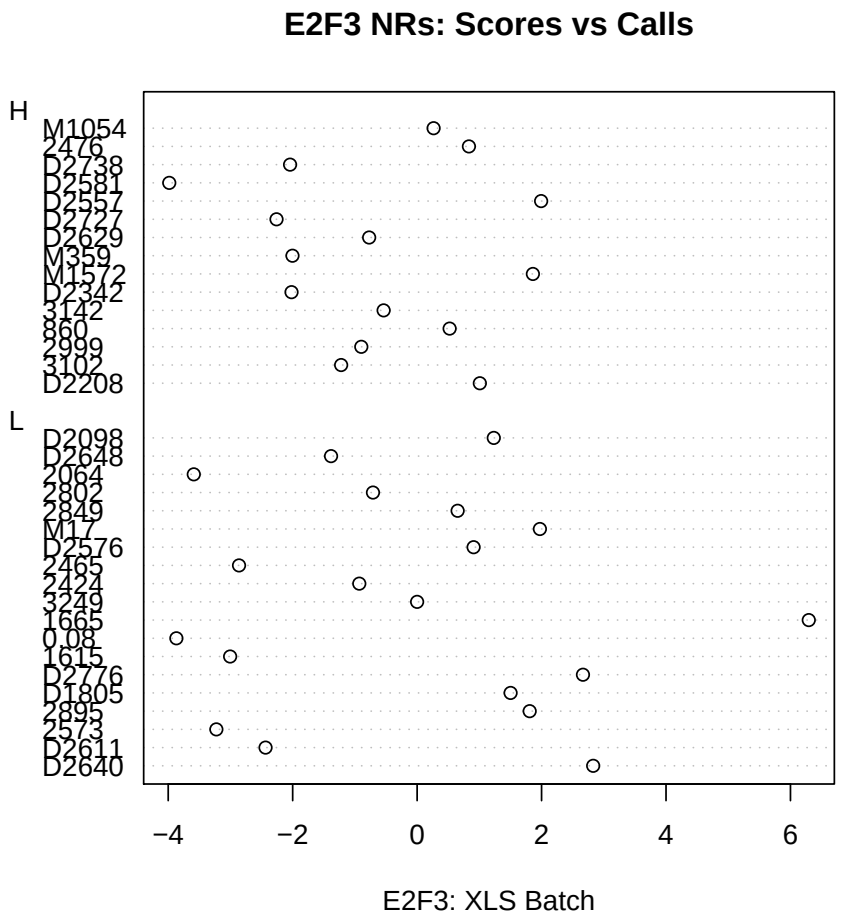

```
> temp <- pathwayScoresXLS$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: XLS")
```



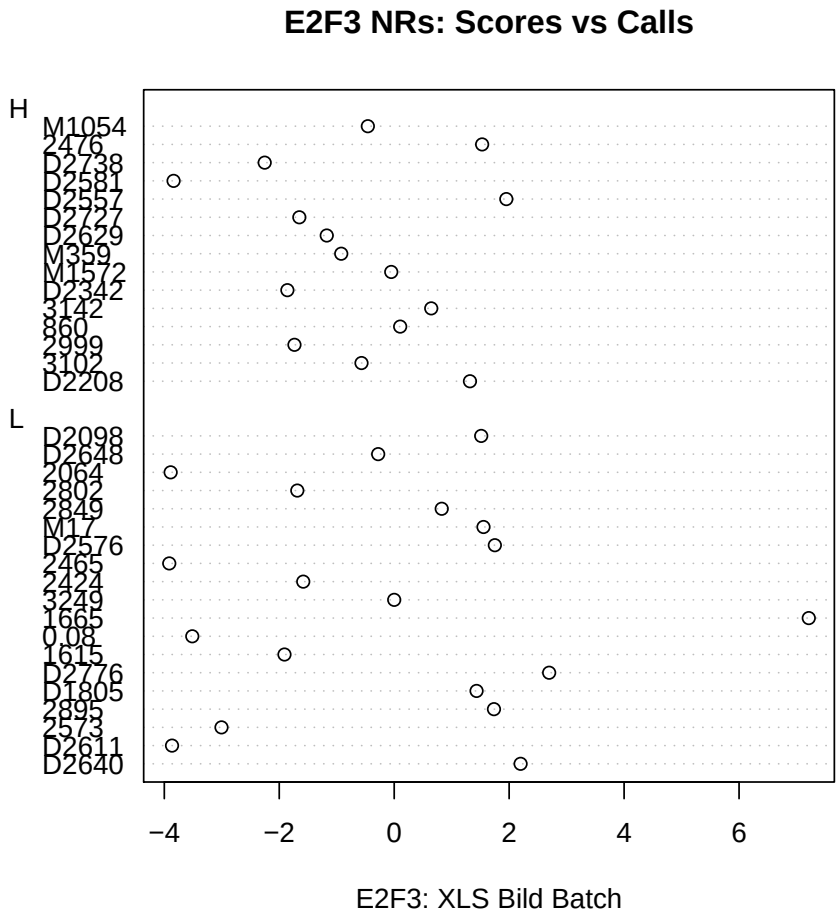

```
> temp <- pathwayScoresXLSBild$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: XLS Bild")
```




```
> temp <- pathwayScoresXLSBatch$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: XLS Batch")
```




```
> temp <- pathwayScoresXLSBildBatch$E2F3[names(e2f3HighLowNR)]
> dotchart(temp, groups = as.factor(e2f3HighLowNR), main = "E2F3 NRs: Scores vs Calls",
+     xlab = "E2F3: XLS Bild Batch")
```



In all of the above plots, there is substantial overlap between the scores assigned to the NR samples that Dressman et al designate as high and low. Their scores clearly do not match ours.

11 Discussion

Our assessment of the separation in survival due to differences in pathway scores is quite different than what Dressman et al report. We do find some separation for Src, E2F3, and Bcat, but not for Myc or Src. The differences that we find, however, are confounded with run batch, especially in the case of Src and Bcat. After correcting for batch, these pathways are not significantly involved. With E2F3, the differences that we find are driven by the CRs, not the NRs. Separation does get a bit better for E2F3 using the censoring from Bild et al, but this was not used in Dressman et al. There may be some separation to be exploited here, but we note that the p-value may not be that small in light of the number of different tests being performed.

Part of the issue is that the scores that we assign to the samples are different than the ones that they used. Our scores were based on defining a model for assigning scores using just the oncogenic pathway arrays, and then using that model to predict the status of the ovarian tumor samples. The scores computed in Bild

et al are based on a different approach, which proceeds as follows.

- (a) A matrix giving the expression values and train/test status for both the training and the testing data is supplied to a fitting routine.
- (b) The set of genes to explore is selected based on the training data alone, using two-sample t-tests and expression values on the log scale. The submatrix involving only these genes is extracted.
- (c) Rows of the selected submatrix are centered and scaled, and a singular value decomposition is applied to the result. Note that these steps are applied to both the training and the test data at the same time.
- (d) The weights associated with the first few singular vectors are then used with the training data to build a logistic regression model, returning a probability of the pathway being "high". This logistic regression model is then used to compute scores and weights for the test samples as well.

We chose to not emply the above approach, because we are uneasy with step (c), which allows values in the test data to affect the coefficients of the predictive model.

Using the scores that we compute, we do not see a large story here.

12 Appendix

12.1 Saves

```
> save(survDressman, file = paste("RDataObjects", "survDressman.Rda",
+     sep = .Platform$file.sep))
> save(survBild, file = paste("RDataObjects", "survBild.Rda", sep = .Platform$file.sep))
> save(pathwayScores, file = paste("RDataObjects", "pathwayScoreNames.Rda",
+     sep = .Platform$file.sep))
> save(patientSubgroups, file = paste("RDataObjects", "patientSubgroups.Rda",
+     sep = .Platform$file.sep))
> save(pathwaySurvTests, file = paste("RDataObjects", "pathwaySurvTests.Rda",
+     sep = .Platform$file.sep))
> save(pathwayKMFits, file = paste("RDataObjects", "pathwayKMFits.Rda",
+     sep = .Platform$file.sep))
> save(pathwaySurvTestsBild, file = paste("RDataObjects", "pathwaySurvTestsBild.Rda",
+     sep = .Platform$file.sep))
> save(pathwayKMFitsBild, file = paste("RDataObjects", "pathwayKMFitsBild.Rda",
+     sep = .Platform$file.sep))
```

12.2 SessionInfo

```
> sessionInfo()

R version 2.5.1 (2007-06-27)
i386-pc-mingw32

locale:
LC_COLLATE=English_United States.1252;LC_CTYPE=English_United States.1252;LC_MONETARY=English_United Sta
```

```
attached base packages:
[1] "splines"   "tools"     "stats"     "graphics"  "grDevices" "utils"
[7] "datasets"  "methods"   "base"

other attached packages:
       survival      colorspace  ClassDiscovery         cluster ClassComparison
         "2.32"          "0.95"         "2.5.0"        "1.11.7"         "2.5.0"
     PreProcess       oompaBase            affy          affyio         Biobase
        "2.5.0"         "2.5.0"        "1.14.2"         "1.4.1"        "1.14.1"
```