Computing MAS5 Present/Absent Calls

1 Introduction

This report describes the first step in the analysis of 75 experiments performed by Response Genetics for the Lung SPORE/DOD projects. All samples came from formalin-fixed paraffin-embedded (FFPE) samples of lung cancer.

In this particular report, we simply compute the MAS5 expression and present/absent calls.

2 MAS5 Computations

We use these libraries:

```
> library(simpleaffy)
> library(affy)
```

2.1 Data Storage and Memory Management

We start working in the following directory:

```
> getwd()
```

```
[1] "C:/Documents and Settings/GXIAO/Desktop/FFPE Fit Sweave 0315 2010"
```

All CEL files have been placed in the same working directory

```
> celpath <- file.path(".", "CELfile")
> fileNames <- list.celfiles(celpath)
```

We set aside as much RAM as possible for the R process.

```
> memory.limit(size = 4000)
```

```
NULL
```

Because the MAS5 computations require reading everything into RAM as an `AffyBatch` object, we need to work things out in batches. The following `kluge` works around this difficulty.

```
> kluge <- function(a, b) {
+     fnames <- file.path(celpath, fileNames[a:b])
+     data <- read.affybatch(fnames)
+     x.mas <- call.exprs(data, "mas5")
+     calls <- mas5calls(data)
```

```
+     write.exprs(x.mas, file = file.path(".", "Output", paste("mas5_",
+         a, "_", b, ".txt", sep = "")))
+     write.exprs(calls, file = file.path(".", "Output", paste("mas5_call_",
+         a, "_", b, ".txt", sep = "")))
+     qcs <- qc(data, x.mas)
+     qc.out <- data.frame(scale = qcs@scale.factors, percent.present = qcs@percent.present,
+         average.background = qcs@average.background, minimum.background = qcs@minimum.background,
+         maximum.background = qcs@maximum.background, qcs@spikes, qcs@qc.probes)
+     write.csv(qc.out, file = file.path(".", "Output", paste("qc_",
+         a, "_", b, ".csv", sep = "")), row.names = TRUE)
+ }
```

2.2 The Actual Computations

Now we manually run through computing the MAS5 calls twenty arrays at a time.

```
> kluge(1, 20)

Getting probe level data...
Computing p-values
Making P/M/A Calls

> gc()

         used (Mb) gc trigger  (Mb)  max used  (Mb)
Ncells 1081660 28.9    3962908 105.9   4953636 132.3
Vcells 1877674 14.4   81083053 618.7 115069044 878.0

> memory.size()

[1] 61.31004

> kluge(21, 40)

Getting probe level data...
Computing p-values
Making P/M/A Calls

> gc()

         used (Mb) gc trigger  (Mb)  max used  (Mb)
Ncells 1081683 28.9    3616491  96.6   4953636 132.3
Vcells 1877700 14.4   92750712 707.7 117337694 895.3

> memory.size()

[1] 173.5549

> kluge(41, 60)

Getting probe level data...
Computing p-values
Making P/M/A Calls
```

```
> gc()

          used (Mb) gc trigger  (Mb)  max used  (Mb)
Ncells 1081687 28.9    3654615  97.6   4953636 132.3
Vcells 1877717 14.4   92750727 707.7 117365293 895.5

> memory.size()

[1] 168.4987

> kluge(61, 75)

Getting probe level data...
Computing p-values
Making P/M/A Calls

> gc()

          used (Mb) gc trigger  (Mb)  max used  (Mb)
Ncells 1081691 28.9    3908271 104.4   4953636 132.3
Vcells 1877739 14.4   63986947 488.2 117365293 895.5

> memory.size()

[1] 61.31732
```

3 Appendix

This computation was performed in the following environment:

```
> sessionInfo()

R version 2.6.1 (2007-11-26)
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     datasets  methods
[9] base

other attached packages:
[1] hgu133plus2cdf_2.0.0 simpleaffy_2.14.05   gcrma_2.10.0         matchprobes_1.10.0
[5] genefilter_1.16.0    survival_2.34        affy_1.14.2          affyio_1.4.1
[9] Biobase_1.16.1

loaded via a namespace (and not attached):
[1] annotate_1.14.1     AnnotationDbi_1.0.6 DBI_0.2-4           RSQLite_0.7-1
```

Quality Control of the Response Genetics Data

1 Introduction

This report describes the second step in the analysis of 75 FFPE Affymetrix experiments performed by Response Genetics for the Lung SPORE/DOD projects.

In this particular report, we use quality control metrics to reduce the number of genes and arrays under consideration.

2 Reading the Quality Control Metrics

We use these libraries:

```
> library(simpleaffy)
> library(affy)
```

The quality control metrics were computed in the first report (`step01-MAS5`). Here we read them into R.

```
> qcs1 <- read.csv(file.path(".", "Output", "qc_1_20.csv"))
> qcs2 <- read.csv(file.path(".", "Output", "qc_21_40.csv"))
> qcs3 <- read.csv(file.path(".", "Output", "qc_41_60.csv"))
> qcs4 <- read.csv(file.path(".", "Output", "qc_61_75.csv"))
```

Now we combine the QC values into a single structure.

```
> qc.all <- rbind(qcs1, qcs2, qcs3, qcs4)
> rm(qcs1, qcs2, qcs3, qcs4)
> write.csv(qc.all, file = file.path(".", "Output", "qc_all.csv"), row.names = FALSE)
```

Next, we read the same data back in, using the first column to supply row names.

```
> qc.all <- read.csv(file.path(".", "Output", "qc_all.csv"), row.names = 1)
> qc.all[1:6, 1:3]
```

```
                          scale percent.present average.background
AGR08-562.CEL.present 9.142070       14.924554           64.24909
AGR08-563.CEL.present 7.711865        9.733882           87.86460
AGR08-564.CEL.present 4.034480       16.932785          103.96073
AGR08-565.CEL.present 7.191250        8.074989          121.73156
AGR08-566.CEL.present 9.117584        7.860997           97.36753
AGR08-567.CEL.present 6.276735       20.040238           70.77899
```

We can clearly simplify the row names.

```
> rownames(qc.all) <- substr(rownames(qc.all), 1, 9)
> qc.all[1:6, 1:4]
```

```
             scale percent.present average.background minimum.background
AGR08-562 9.142070       14.924554           64.24909           61.21925
AGR08-563 7.711865        9.733882           87.86460           83.11230
AGR08-564 4.034480       16.932785          103.96073           98.11364
AGR08-565 7.191250        8.074989          121.73156          115.24502
AGR08-566 9.117584        7.860997           97.36753           91.17045
AGR08-567 6.276735       20.040238           70.77899           67.44786
```

3 Some Clinical Annotations

We next read some clinical information about the samples.

```
> date <- read.csv("date.csv")
> dim(date)
```

```
[1] 120   5
```

```
> head(date)
```

```
  MDACC.Specimen.Number Date.of.Surgery      Now day360     year
1                  MDA1        8/5/2005 8/1/2008   1076 2.947945
2                 MDA10       3/18/2005 8/1/2008   1213 3.323288
3                MDA100       10/6/1999 8/1/2008   3175 8.698630
4                MDA101        9/3/1999 8/1/2008   3208 8.789041
5                MDA102      12/15/1999 8/1/2008   3106 8.509589
6                MDA103        8/5/1999 8/1/2008   3236 8.865753
```

```
> ID <- read.csv("ID.csv")
> dim(ID)
```

```
[1] 75  2
```

```
> head(ID)
```

```
  Specimen.Number   Chip.ID
1            MDA3 AGR08-562
2            MDA4 AGR08-563
3            MDA5 AGR08-564
4            MDA6 AGR08-565
5            MDA7 AGR08-566
6            MDA9 AGR08-567
```

Clearly, the first file maps the MDA specimen number to some survival information, while the second file maps the MDA specimen number to the CEL file. We can now merge these into a single structure.

```
> date <- merge(ID, date, by.x = "Specimen.Number", by.y = "MDACC.Specimen.Number")
> dim(date)
```

```
[1] 75  6
```

```
> head(date)
```

```
  Specimen.Number   Chip.ID Date.of.Surgery      Now day360     year
1           MDA10 AGR08-568       3/18/2005 8/1/2008   1213 3.323288
2          MDA109 AGR08-661       2/27/2001 8/1/2008   2674 7.326027
3           MDA11 AGR08-569       4/19/2005 8/1/2008   1182 3.238356
4          MDA111 AGR08-663        3/1/2001 8/1/2008   2670 7.315068
5          MDA114 AGR08-666       5/25/2001 8/1/2008   2586 7.084932
6          MDA117 AGR08-669       7/26/2001 8/1/2008   2525 6.917808
```

The next step is to merge the clinical annotations with the quality control measures on the CEL files.

```
> out <- merge(date, qc.all, by.x = "Chip.ID", by.y = "row.names")
> dim(out)
```

```
[1] 75 21
```

```
> head(out)
```

```
    Chip.ID Specimen.Number Date.of.Surgery      Now day360     year    scale
1 AGR08-562            MDA3       7/21/2005 8/1/2008   1090 2.986301 9.142070
2 AGR08-563            MDA4        7/1/2005 8/1/2008   1110 3.041096 7.711865
3 AGR08-564            MDA5       7/18/2005 8/1/2008   1093 2.994521 4.034480
4 AGR08-565            MDA6       5/20/2005 8/1/2008   1151 3.153425 7.191250
5 AGR08-566            MDA7       4/27/2005 8/1/2008   1174 3.216438 9.117584
6 AGR08-567            MDA9       4/28/2005 8/1/2008   1173 3.213699 6.276735
  percent.present average.background minimum.background maximum.background
1       14.924554           64.24909           61.21925           65.76459
2        9.733882           87.86460           83.11230           90.60596
3       16.932785          103.96073           98.11364          107.65860
4        8.074989          121.73156          115.24502          130.46763
5        7.860997           97.36753           91.17045          101.52322
6       20.040238           70.77899           67.44786           73.57750
  AFFX.r2.Ec.bioB.3_at AFFX.r2.Ec.bioC.3_at AFFX.r2.Ec.bioD.3_at AFFX.r2.P1.cre.3_at
1             11.94451             13.46356             15.44630            16.95217
2             10.81255             12.99849             15.02482            16.52847
3              9.97492             11.95150             13.77272            15.53087
4             10.87180             12.75666             14.89362            16.41770
5             10.98516             13.18725             15.26179            16.71391
6             11.09174             12.87277             14.75510            16.39212
  AFFX.HSAC07.X00351_3_at AFFX.HSAC07.X00351_5_at AFFX.HSAC07.X00351_M_at
1                11.47109                6.865734                5.208003
2                10.96383                7.186255                7.415933
3                11.48040                6.101688                4.982652
4                10.52984                7.246127                5.986817
5                10.98318                7.605745                7.718577
6                11.80284                6.529268                3.897452
  AFFX.HUMGAPDH.M33197_3_at AFFX.HUMGAPDH.M33197_5_at AFFX.HUMGAPDH.M33197_M_at
```

```
1                 9.327008                 6.043686                 5.933223
2                10.568375                 7.355697                 5.417545
3                11.587130                 6.766997                 6.909541
4                10.812368                 7.659383                 5.812756
5                10.869056                 6.381063                 5.873258
6                 9.024217                 6.724048                 5.616518
```

Now we save this combined annotation information into a file.

```
> rm(date, ID)
> write.csv(out, file.path(".", "Output", "qc.csv"), row.names = FALSE)
```

In order to get the row names re-attached, we use the previous trick of reading the same data back in.

```
> qc <- read.csv(file.path(".", "Output", "qc.csv"), row.names = 1)
> head(qc)
```

```
          Specimen.Number Date.of.Surgery      Now day360     year    scale
AGR08-562            MDA3       7/21/2005 8/1/2008   1090 2.986301 9.142070
AGR08-563            MDA4        7/1/2005 8/1/2008   1110 3.041096 7.711865
AGR08-564            MDA5       7/18/2005 8/1/2008   1093 2.994521 4.034480
AGR08-565            MDA6       5/20/2005 8/1/2008   1151 3.153425 7.191250
AGR08-566            MDA7       4/27/2005 8/1/2008   1174 3.216438 9.117584
AGR08-567            MDA9       4/28/2005 8/1/2008   1173 3.213699 6.276735
          percent.present average.background minimum.background maximum.background
AGR08-562       14.924554           64.24909           61.21925           65.76459
AGR08-563        9.733882           87.86460           83.11230           90.60596
AGR08-564       16.932785          103.96073           98.11364          107.65860
AGR08-565        8.074989          121.73156          115.24502          130.46763
AGR08-566        7.860997           97.36753           91.17045          101.52322
AGR08-567       20.040238           70.77899           67.44786           73.57750
          AFFX.r2.Ec.bioB.3_at AFFX.r2.Ec.bioC.3_at AFFX.r2.Ec.bioD.3_at
AGR08-562             11.94451             13.46356             15.44630
AGR08-563             10.81255             12.99849             15.02482
AGR08-564              9.97492             11.95150             13.77272
AGR08-565             10.87180             12.75666             14.89362
AGR08-566             10.98516             13.18725             15.26179
AGR08-567             11.09174             12.87277             14.75510
          AFFX.r2.P1.cre.3_at AFFX.HSAC07.X00351_3_at AFFX.HSAC07.X00351_5_at
AGR08-562            16.95217                11.47109                6.865734
AGR08-563            16.52847                10.96383                7.186255
AGR08-564            15.53087                11.48040                6.101688
AGR08-565            16.41770                10.52984                7.246127
AGR08-566            16.71391                10.98318                7.605745
AGR08-567            16.39212                11.80284                6.529268
          AFFX.HSAC07.X00351_M_at AFFX.HUMGAPDH.M33197_3_at AFFX.HUMGAPDH.M33197_5_at
AGR08-562                5.208003                 9.327008                 6.043686
AGR08-563                7.415933                10.568375                 7.355697
AGR08-564                4.982652                11.587130                 6.766997
AGR08-565                5.986817                10.812368                 7.659383
```

```
AGR08-566                7.718577                 10.869056                  6.381063
AGR08-567                3.897452                  9.024217                  6.724048
          AFFX.HUMGAPDH.M33197_M_at
AGR08-562                  5.933223
AGR08-563                  5.417545
AGR08-564                  6.909541
AGR08-565                  5.812756
AGR08-566                  5.873258
AGR08-567                  5.616518
```

4 Gene Information

We read the gene information from a file. This file maps Affymetrix probe set IDs to GenBank accession numbers.

```
> gene.info <- read.csv("gene_info.csv")
> dim(gene.info)

[1] 54675     3

> head(gene.info)

          Affy.ID Accession Symbol
1  AFFX-BioB-5_at    J04423
2  AFFX-BioB-M_at    U00096
3  AFFX-BioB-3_at    U00096
4  AFFX-BioC-5_at    U00096
5  AFFX-BioC-3_at    J04423
6 AFFX-BioDn-5_at    U00096
```

5 Affymetrix Present/Absent Calls

In the previous report, we computed the Affymetrix present/absent calls for each probe set in each sample. Here we read that information back into R.

```
> c1 <- read.table(file.path(".", "Output", "mas5_call_1_20.txt"), sep = "\t",
+     row.names = 1, head = TRUE)
> c2 <- read.table(file.path(".", "Output", "mas5_call_21_40.txt"), sep = "\t",
+     row.names = 1, head = TRUE)
> c3 <- read.table(file.path(".", "Output", "mas5_call_41_60.txt"), sep = "\t",
+     row.names = 1, head = TRUE)
> c4 <- read.table(file.path(".", "Output", "mas5_call_61_75.txt"), sep = "\t",
+     row.names = 1, head = TRUE)
```

As before, we combine this information into a single structure.

```
> call <- data.frame(c1, c2, c3, c4)
> head(call, 2)
```

```
          AGR08.562.CEL AGR08.563.CEL AGR08.564.CEL AGR08.565.CEL AGR08.566.CEL
1007_s_at             P             A             P             A             A
1053_at               A             A             A             A             A
          AGR08.567.CEL AGR08.568.CEL AGR08.569.CEL AGR08.570.CEL AGR08.571.CEL
1007_s_at             P             P             A             P             P
1053_at               A             A             A             A             A
          AGR08.572.CEL AGR08.573.CEL AGR08.574.CEL AGR08.575.CEL AGR08.576.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.577.CEL AGR08.579.CEL AGR08.580.CEL AGR08.581.CEL AGR08.582.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.583.CEL AGR08.584.CEL AGR08.586.CEL AGR08.588.CEL AGR08.589.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.591.CEL AGR08.593.CEL AGR08.595.CEL AGR08.596.CEL AGR08.597.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.598.CEL AGR08.599.CEL AGR08.600.CEL AGR08.601.CEL AGR08.602.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.603.CEL AGR08.604.CEL AGR08.605.CEL AGR08.606.CEL AGR08.607.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.608.CEL AGR08.609.CEL AGR08.610.CEL AGR08.612.CEL AGR08.613.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.614.CEL AGR08.615.CEL AGR08.616.CEL AGR08.617.CEL AGR08.618.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.620.CEL AGR08.621.CEL AGR08.622.CEL AGR08.623.CEL AGR08.625.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.626.CEL AGR08.627.CEL AGR08.629.CEL AGR08.630.CEL AGR08.631.CEL
1007_s_at             M             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.633.CEL AGR08.634.CEL AGR08.635.CEL AGR08.637.CEL AGR08.638.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.640.CEL AGR08.641.CEL AGR08.643.CEL AGR08.644.CEL AGR08.648.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
          AGR08.661.CEL AGR08.663.CEL AGR08.666.CEL AGR08.669.CEL AGR08.672.CEL
1007_s_at             P             P             P             P             P
1053_at               A             A             A             A             A
```

For various purposes, it will be easier to recode the calls numerically.

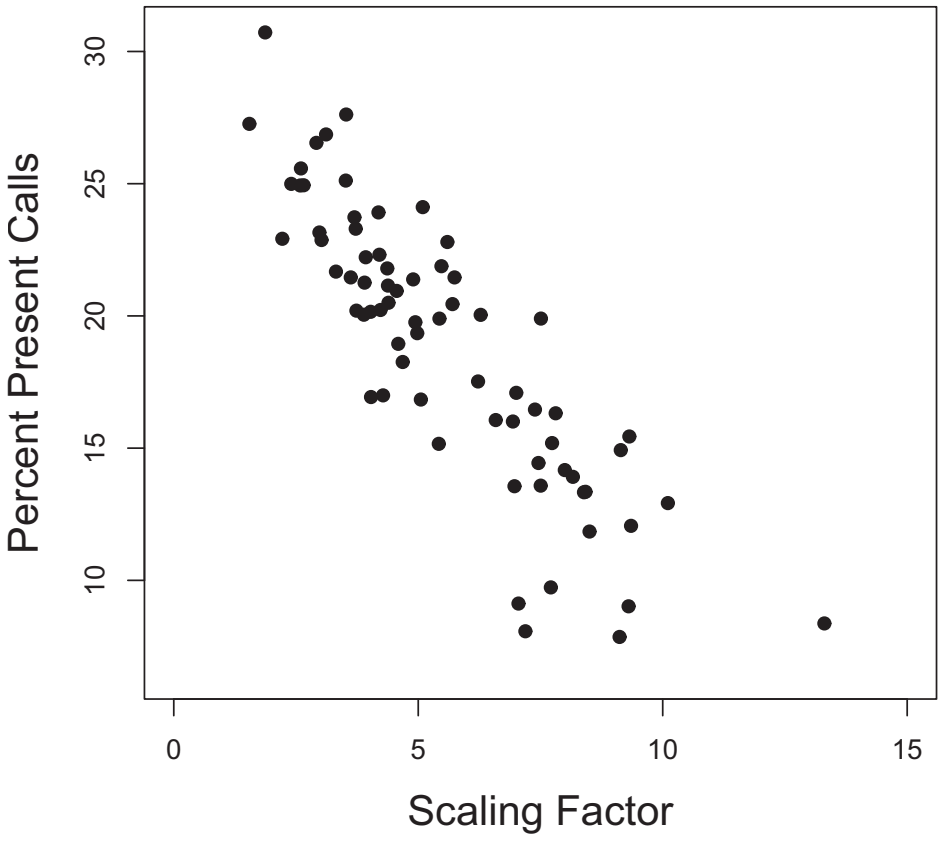


Figure 1: Scatter plot of the MAS5 scaling factors and percent present calls on the CEL files.

```
> to.numeric <- function(x) {
+     ifelse(x == "A", 0, 1)
+ }
> call <- to.numeric(call)
> id <- data.frame(qc[, c(1, 7)], colnames(call))
```

Now we start exploring the QC data. Figure 1 shows that the percentage of present calls on an array is negatively correlated with its scaling factor. Arrays with larger scaling factors (because the overall brightness is low) have fewer genes that are called present. We also note that the percentage of present calls is well below the usual range of 30% to 60% that we typically see when using fresh frozen samples; the difference is almost certainly explained by RNA degradation of the FFPE samples.

We decide (more or less arbitrarily) to ignore arrays on which at most 15% of the genes are called present.

```
> ind.array <- which(id$percent.present > 15)
```

That will leave only 55 CEL files for use in our further analyses.

Now we look at the present/absent calls for individual probe sets. The next piece of code computes, for each gene, the percentage of times it is called present among the QC-selected arrays. The results are summarized in Figure 2.

```
> call.selected <- call[, ind.array]
> rate <- apply(call.selected, 1, mean)
> length(which(rate == 1))

[1] 1804

> sum(rate > 0.5)

[1] 10778

> sum(rate > 0.05)

[1] 29108
```

We again arbitrarily decide that we will only consider probe sets that are called present in all of the QC-selected samples.

```
> dat.selected <- call[rate == 1, ind.array]
> dim(dat.selected)

[1] 1804   55

> dat.selected <- merge(gene.info, dat.selected, by.x = "Affy.ID", by.y = "row.names")
> nblank <- length(which(dat.selected$Symbol == ""))
> nsel <- nrow(dat.selected) - nblank
> nblank

[1] 404
```

Data for the 1400 selected genes are now saved in anticipation of the next step in the analysis.

```
> write.csv(dat.selected[dat.selected$Symbol != "", ], file.path(".",
+     "Output", "dat.selected.csv"), row.names = FALSE)
```

6 Appendix

This computation was performed in the following environment:

```
> sessionInfo()

R version 2.6.1 (2007-11-26)
i386-pc-mingw32

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

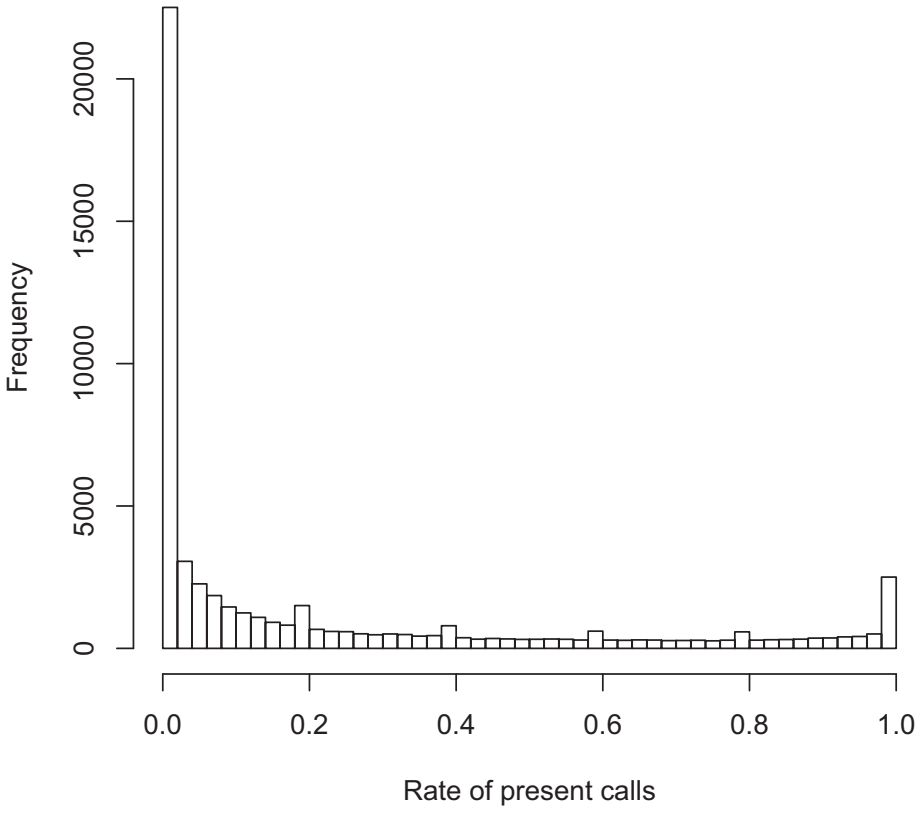


Figure 2: Histogram of the fraction of time a probe set was called present among the QC-selected arrays.

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

other attached packages:
[1] hgu133plus2cdf_2.0.0 simpleaffy_2.14.05   gcrma_2.10.0         matchprobes_1.10.0
[5] genefilter_1.16.0    survival_2.34        affy_1.14.2          affyio_1.4.1
[9] Biobase_1.16.1

loaded via a namespace (and not attached):
[1] annotate_1.14.1     AnnotationDbi_1.0.6 DBI_0.2-4           RSQLite_0.7-1
```

Extracting Probe-Level Response Genetics Data

1 Introduction

This report describes the third step in the analysis of 75 FFPE Affymetrix experiments performed by Response Genetics for the Lung SPORE/DOD projects.

In this particular report, we combine probe locations with probe intensities for the genes and arrays that passed our QC filtering in Step 2.

2 Getting Ready

We use these libraries:

```
> library(simpleaffy)
> library(affy)
> library(MASS)
```

We increase the (RAM) memory limit.

```
> memory.limit(4000)
```

```
NULL
```

We load the (combined) QC data from step 2.

```
> qcs <- read.csv(file.path(".", "Output", "qc.csv"), row.names = 1)
> head(qcs)
```

```
          Specimen.Number Date.of.Surgery      Now day360     year    scale
AGR08-562            MDA3       7/21/2005 8/1/2008   1090 2.986301 9.142070
AGR08-563            MDA4        7/1/2005 8/1/2008   1110 3.041096 7.711865
AGR08-564            MDA5       7/18/2005 8/1/2008   1093 2.994521 4.034480
AGR08-565            MDA6       5/20/2005 8/1/2008   1151 3.153425 7.191250
AGR08-566            MDA7       4/27/2005 8/1/2008   1174 3.216438 9.117584
AGR08-567            MDA9       4/28/2005 8/1/2008   1173 3.213699 6.276735
          percent.present average.background minimum.background maximum.background
AGR08-562       14.924554           64.24909           61.21925           65.76459
AGR08-563        9.733882           87.86460           83.11230           90.60596
AGR08-564       16.932785          103.96073           98.11364          107.65860
AGR08-565        8.074989          121.73156          115.24502          130.46763
AGR08-566        7.860997           97.36753           91.17045          101.52322
AGR08-567       20.040238           70.77899           67.44786           73.57750
```

```
          AFFX.r2.Ec.bioB.3_at AFFX.r2.Ec.bioC.3_at AFFX.r2.Ec.bioD.3_at
AGR08-562             11.94451             13.46356             15.44630
AGR08-563             10.81255             12.99849             15.02482
AGR08-564              9.97492             11.95150             13.77272
AGR08-565             10.87180             12.75666             14.89362
AGR08-566             10.98516             13.18725             15.26179
AGR08-567             11.09174             12.87277             14.75510
          AFFX.r2.P1.cre.3_at AFFX.HSAC07.X00351_3_at AFFX.HSAC07.X00351_5_at
AGR08-562            16.95217                11.47109                6.865734
AGR08-563            16.52847                10.96383                7.186255
AGR08-564            15.53087                11.48040                6.101688
AGR08-565            16.41770                10.52984                7.246127
AGR08-566            16.71391                10.98318                7.605745
AGR08-567            16.39212                11.80284                6.529268
          AFFX.HSAC07.X00351_M_at AFFX.HUMGAPDH.M33197_3_at AFFX.HUMGAPDH.M33197_5_at
AGR08-562                5.208003                 9.327008                 6.043686
AGR08-563                7.415933                10.568375                 7.355697
AGR08-564                4.982652                11.587130                 6.766997
AGR08-565                5.986817                10.812368                 7.659383
AGR08-566                7.718577                10.869056                 6.381063
AGR08-567                3.897452                 9.024217                 6.724048
          AFFX.HUMGAPDH.M33197_M_at
AGR08-562                  5.933223
AGR08-563                  5.417545
AGR08-564                  6.909541
AGR08-565                  5.812756
AGR08-566                  5.873258
AGR08-567                  5.616518
```

Use the QC rownames and percent present calls to construct a list of the names of files having more than 15% present calls.

```
> celpath <- file.path(".", "CELfile")
> fileNames <- paste(rownames(qcs), "CEL", sep = ".")[qcs$percent.present >
+     15]
```

Also get a list of the probe set IDs that were selected in step 2 because they passed the QC filter (i.e., they were called present on all arrays that had at least 15% present calls).

```
> selected.id <- read.csv(file.path(".", "Output", "dat.selected.csv"),
+     row.names = 1)
> head(selected.id, 1)
```

```
          Accession Symbol AGR08.564.CEL AGR08.567.CEL AGR08.568.CEL AGR08.571.CEL
1007_s_at NM_013994   DDR1             1             1             1             1
          AGR08.574.CEL AGR08.576.CEL AGR08.577.CEL AGR08.579.CEL AGR08.580.CEL
1007_s_at             1             1             1             1             1
          AGR08.581.CEL AGR08.582.CEL AGR08.583.CEL AGR08.584.CEL AGR08.586.CEL
1007_s_at             1             1             1             1             1
          AGR08.588.CEL AGR08.589.CEL AGR08.591.CEL AGR08.593.CEL AGR08.595.CEL
```

```
1007_s_at             1             1             1             1             1
          AGR08.597.CEL AGR08.598.CEL AGR08.600.CEL AGR08.601.CEL AGR08.602.CEL
1007_s_at             1             1             1             1             1
          AGR08.603.CEL AGR08.604.CEL AGR08.605.CEL AGR08.606.CEL AGR08.607.CEL
1007_s_at             1             1             1             1             1
          AGR08.608.CEL AGR08.609.CEL AGR08.613.CEL AGR08.614.CEL AGR08.615.CEL
1007_s_at             1             1             1             1             1
          AGR08.616.CEL AGR08.617.CEL AGR08.618.CEL AGR08.621.CEL AGR08.622.CEL
1007_s_at             1             1             1             1             1
          AGR08.625.CEL AGR08.627.CEL AGR08.629.CEL AGR08.630.CEL AGR08.631.CEL
1007_s_at             1             1             1             1             1
          AGR08.633.CEL AGR08.634.CEL AGR08.635.CEL AGR08.637.CEL AGR08.638.CEL
1007_s_at             1             1             1             1             1
          AGR08.640.CEL AGR08.643.CEL AGR08.644.CEL AGR08.666.CEL AGR08.669.CEL
1007_s_at             1             1             1             1             1
          AGR08.672.CEL
1007_s_at             1
```

3 Probe Level Data

For each array, we first use RMA to background correct the probe level data. We then extract the probes for the selected probe sets and save them in a separate file.

```
> for (i in 1:length(fileNames)) {
+     Data <- read.affybatch(file.path(celpath, fileNames[i]))
+     bgc <- bg.correct(Data, method = "rma")
+     probe.selected <- pm(bgc, rownames(selected.id))
+     write.csv(probe.selected, file.path(".", "Output", paste("selected_probe",
+         i, "csv", sep = ".")))
+ }
```

Now we merge these data back into a single structure.

```
> selected.dat <- read.csv(file.path(".", "Output", paste("selected_probe",
+     1, "csv", sep = ".")), row.names = 1)
> for (i in 2:length(fileNames)) {
+     tmp <- read.csv(file.path(".", "Output", paste("selected_probe",
+         i, "csv", sep = ".")), row.names = 1)
+     print(dim(tmp))
+     if (any(rownames(tmp) != rownames(selected.dat))) {
+         stop(paste("Probe file", i, "is missing some probes"))
+     }
+     selected.dat <- data.frame(selected.dat, tmp)
+ }
```

```
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
```

```
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
[1] 15775     1
```

```
[1] 15775     1
[1] 15775     1
```

```
> dim(selected.dat)
```

```
[1] 15775    55
```

4 Relative probe locations

Read a file containing the probe locations.

```
> loc <- read.csv("Probe location.csv")
> head(loc)
```

```
      Probe Probe.locations    X  X.1  X.2  X.3  X.4  X.5  X.6  X.7  X.8  X.9 X.10 X.11
1 1007_s_at            3330 3443 3512 3563 3570 3576 3583 3589 3615 3713 3786 3793 3799
2   1053_at            1090 1102 1108 1126 1180 1186 1210 1216 1228 1288 1300 1312 1318
3    117_at            1463 1631 1643 1649 1655 1691 1697 1703 1727 1733 1739 1751 1811
4    121_at            2170 2218 2308 2374 2380 2392 2398 2464 2500 2572 2584 2590 2602
5 1255_g_at            1225 1236 1417 1436 1441 1446 1477 1487 1494 1518 1547 1559 1588
6   1294_at            2688 2694 2736 2814 2820 2922 2952 3018 3030 3090 3102 3114 3198
  X.12 X.13 X.14
1 3807 3871 3878
2 1360 1372 1396
3 1871 1877 1883
4 2608 2650 2668
5 1593 1599 1609
6 3204 3210 3264
```

Pick out the subset of these probe locations that matter for the selected probe sets.

```
> locf <- loc[loc$Probe %in% rownames(selected.id), ]
> dim(locf)
```

```
[1] 1400   17
```

Get the largest location within each probe set.

```
> locf.max <- apply(locf[, -1], 1, max, na.rm = TRUE)
```

Recompute the probe locations in terms of the position relative to the largest location in each set.

```
> probe.loc <- unlist(data.frame(t(locf[, -1] - locf.max)), use.names = FALSE)
> head(probe.loc, 50)
```

```
 [1] -548 -435 -366 -315 -308 -302 -295 -289 -263 -165  -92  -85  -79  -71   -7    0
[17] -285 -219 -156 -153 -129  -72  -69  -60  -48  -45  -21  -15  -12   -9   -3    0
[33] -471 -451 -340  -17  -13  -10   -8   -7   -4   -1    0   NA   NA   NA   NA   NA
[49] -387 -346
```

Put the new probe location into a data frame.

```
> loc.dat <- data.frame(set = rep(locf$Probe, each = 16), probe = paste(rep(locf$Probe,
+     each = 16), rep(1:16, times = length(locf$Probe)), sep = ""), loc = probe.loc)
> dim(loc.dat)
```

```
[1] 22400     3
```

```
> head(loc.dat)
```

```
        set      probe  loc
1 1007_s_at 1007_s_at1 -548
2 1007_s_at 1007_s_at2 -435
3 1007_s_at 1007_s_at3 -366
4 1007_s_at 1007_s_at4 -315
5 1007_s_at 1007_s_at5 -308
6 1007_s_at 1007_s_at6 -302
```

Merge the relative probe locations with the probe intensity data.

```
> loc.dat <- merge(loc.dat, selected.dat, by.x = "probe", by.y = "row.names")
> dim(loc.dat)
```

```
[1] 15710    58
```

Finally, save the results.

```
> write.csv(loc.dat, file.path(".", "Output", "probe_selected_dat.csv"),
+     row.names = FALSE)
```

5 Appendix

This computation was performed in the following environment:

```
> sessionInfo()
```

```
R version 2.6.1 (2007-11-26)
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     datasets  methods
[9] base

other attached packages:
 [1] hgu133plus2cdf_2.0.0 MASS_7.2-38          simpleaffy_2.14.05   gcrma_2.10.0
 [5] matchprobes_1.10.0   genefilter_1.16.0    survival_2.34        affy_1.14.2
 [9] affyio_1.4.1         Biobase_1.16.1

loaded via a namespace (and not attached):
[1] annotate_1.14.1     AnnotationDbi_1.0.6 DBI_0.2-4           RSQLite_0.7-1
```

Fitting intensity models to the Response Genetics Data

1 Introduction

This report describes the fourth step in the analysis of 75 FFPE Affymetrix experiments performed by Response Genetics for the Lung SPORE/DOD projects.

In this particular report, we fit a probe-level model to the (RMA) background-corrected intensities. This model explicitly incorporates the relative probe position (measured in bases from the most extreme probe in the probe set) that were recorded in Step 3. A robust regression model is used reduce the effect of FFPE sample degradation and minimize the effect of extreme values.

2 Getting Ready

We use these libraries:

```
> library(affy)
> library(MASS)
```

We increase the (RAM) memory limit.

```
> memory.limit(4000)
```

```
NULL
```

We read the probe level data (and locations) that was collected in Step 3.

```
> selected.dat <- read.csv(file.path(".", "Output", "probe_selected_dat.csv"))
> selected.dat[1:50, 1:4]
```

```
           probe          set  loc AGR08.564.CEL
1      1007_s_at1    1007_s_at -548     23.295146
2     1007_s_at10    1007_s_at -165    655.862394
3     1007_s_at11    1007_s_at  -92   3338.862394
4     1007_s_at12    1007_s_at  -85   2129.862394
5     1007_s_at13    1007_s_at  -79   1638.862394
6     1007_s_at14    1007_s_at  -71    831.862394
7     1007_s_at15    1007_s_at   -7   6144.862394
8     1007_s_at16    1007_s_at    0  12122.862394
9      1007_s_at2    1007_s_at -435     87.902734
10     1007_s_at3    1007_s_at -366    265.862394
11     1007_s_at4    1007_s_at -315    218.862394
12     1007_s_at5    1007_s_at -308    472.862394
```

```
13     1007_s_at6    1007_s_at -302   1234.862394
14     1007_s_at7    1007_s_at -295     70.175582
15     1007_s_at8    1007_s_at -289    174.862394
16     1007_s_at9    1007_s_at -263    242.862394
17       1316_at1      1316_at -285     66.326560
18      1316_at10      1316_at  -45   3255.862394
19      1316_at11      1316_at  -21      7.520074
20      1316_at12      1316_at  -15     21.333594
21      1316_at13      1316_at  -12     14.493832
22      1316_at14      1316_at   -9     21.708411
23      1316_at15      1316_at   -3     11.611369
24      1316_at16      1316_at    0     11.316169
25       1316_at2      1316_at -219     26.433032
26       1316_at3      1316_at -156     49.105807
27       1316_at4      1316_at -153     14.710180
28       1316_at5      1316_at -129    136.862408
29       1316_at6      1316_at  -72    414.862394
30       1316_at7      1316_at  -69     77.995711
31       1316_at8      1316_at  -60     13.481984
32       1316_at9      1316_at  -48   1850.862394
33    1552302_at1   1552302_at -471     15.389390
34   1552302_at10   1552302_at   -1     77.995711
35   1552302_at11   1552302_at    0     91.886419
36    1552302_at2   1552302_at -451     21.708411
37    1552302_at3   1552302_at -340    635.862394
38    1552302_at4   1552302_at  -17     13.481984
39    1552302_at5   1552302_at  -13     19.916866
40    1552302_at6   1552302_at  -10     22.483959
41    1552302_at7   1552302_at   -8     55.188464
42    1552302_at8   1552302_at   -7     48.268878
43    1552302_at9   1552302_at   -4     54.297324
44  1552426_a_at1 1552426_a_at -387    214.862394
45 1552426_a_at10 1552426_a_at  -18    651.862394
46 1552426_a_at11 1552426_a_at    0   2121.862394
47  1552426_a_at2 1552426_a_at -346     16.895508
48  1552426_a_at3 1552426_a_at -277     15.868637
49  1552426_a_at4 1552426_a_at -161     46.621529
50  1552426_a_at5 1552426_a_at -154     17.167305
```

We reorder these data by location within probe set.

```
> selected.dat <- selected.dat[order(selected.dat$set, selected.dat$loc),
+     ]
> dim(selected.dat)

[1] 15710    58
```

Computations will be performed for each probe set.

```
> set.name <- unique(selected.dat$set)
> NG <- length(set.name)
> NS <- dim(selected.dat)[2] - 3
```

We set aside room to hold the results.

```
> b0 <- b1 <- data.frame(matrix(0, NG, NS))
> rownames(b0) <- rownames(b1) <- set.name
> colnames(b0) <- colnames(b1) <- colnames(selected.dat)[-(1:3)]
```

Now we go ahead and fit the models. This step may take several hours!

```
> for (i in 1:NG) {
+     ind <- which(selected.dat$set == set.name[i])
+     for (j in 1:NS) {
+         fit <- rlm(log2(selected.dat[ind, j + 3]) ~ selected.dat$loc[ind])
+         b0[i, j] <- coef(fit)[1]
+         b1[i, j] <- coef(fit)[2]
+     }
+ }
```

And we perform quantile normalization.

```
> b0.n <- b0
> b0.n[] <- normalize.quantiles(as.matrix(b0))
```

Finally, we save the results.

```
> write.csv(b0.n, file.path(".", "Output", "selected_data_norm_b0.csv"))
```

3 Appendix

This computation was performed in the following environment:

```
> sessionInfo()
```

```
R version 2.6.1 (2007-11-26)
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     datasets  methods
[9] base

other attached packages:
 [1] hgu133plus2cdf_2.0.0 MASS_7.2-38          simpleaffy_2.14.05   gcrma_2.10.0
 [5] matchprobes_1.10.0   genefilter_1.16.0    survival_2.34        affy_1.14.2
 [9] affyio_1.4.1         Biobase_1.16.1

loaded via a namespace (and not attached):
[1] annotate_1.14.1     AnnotationDbi_1.0.6 DBI_0.2-4           RSQLite_0.7-1
```

Predict Prognosis Using Robust Gene Signature

1 Introduction

This report describes the sixth step in the analysis of 75 FFPE Affymetrix experiments performed by Response Genetics for the Lung SPORE/DOD projects.

In this particular report, we apply the fitted model to the Consortium data set of 442 patients reported in their Nature Medicine article (Shedden et al., 2008).

2 Getting Ready

We use these libraries:

```
> library(survival)
> library(superpc)
> library(affy)
> library(preprocessCore)
> pv.expr <- function(x, digits = 1) {
+     if (!x)
+         return(0)
+     exponent <- floor(log10(x))
+     base <- round(x/10^exponent, digits)
+     ifelse(x > 1e-06, paste("p = ", base * (10^exponent), sep = ""),
+         paste("p = ", base, "E", exponent, sep = ""))
+ }
> par(mar = c(4, 4, 3, 1), mfrow = c(1, 1))
```

3 FFPE data clinical variable only

In this section, we explore the association between clinical variables (histology and stage) and the survival or progression in FFPE data.

```
> clin <- read.csv("FFPE.clin.csv", row.names = 1)
> head(clin)
```

	Specimen.Number	SPOR.N	Histology	Final.Pat.Stage	Time_to_Progression
AGR08.564.CEL	MDA5	1724	Squamous	IIA	2.3846680
AGR08.567.CEL	MDA9	1663	Adenocarcinoma	IB	2.2642026
AGR08.568.CEL	MDA10	1623	Squamous	IIIB	0.3778234
AGR08.571.CEL	MDA14	1576	Adenocarcinoma	IIB	2.8199863

```
AGR08.574.CEL           MDA18   1547 Adenocarcinoma             IIB           2.8966461
AGR08.576.CEL           MDA20   1537 Adenocarcinoma              IB           2.1300479
             Progression Death_Time Death_Event stage
AGR08.564.CEL           0   2.384668           0     2
AGR08.567.CEL           0   2.264203           0     1
AGR08.568.CEL           1   1.404517           1     3
AGR08.571.CEL           0   2.819986           0     2
AGR08.574.CEL           1   2.896646           0     2
AGR08.576.CEL           0   2.130048           0     1
```

```
> dim(clin)
```

```
[1] 55  9
```

```
> ffpe.fitted <- read.csv(".\\Output\\selected_data_norm_b0.csv", row.names = 1)
> dim(ffpe.fitted)
```

```
[1] 1400   55
```

```
> expr <- ffpe.fitted[, rownames(clin)]
> dim(expr)
```

```
[1] 1400   55
```

```
> all(rownames(clin) == colnames(expr))
```

```
[1] TRUE
```

In the survival analysis, we show that the Histology is not significantly assoicated with survival and progression. While the stage is associated with survival but not progression.

```
> fit <- survfit(Surv(Death_Time, Death_Event) ~ Histology, data = clin)
> logrank <- survdiff(Surv(Death_Time, Death_Event) ~ Histology, data = clin)
> logrank
```

```
Call:
survdiff(formula = Surv(Death_Time, Death_Event) ~ Histology,
    data = clin)

                         N Observed Expected (O-E)^2/E (O-E)^2/V
Histology=Adenocarcinoma 30        8    10.04     0.413     0.936
Histology=Squamous       25       10     7.96     0.520     0.936

 Chisq= 0.9  on 1 degrees of freedom, p= 0.333
```

Histology is not significantly associated with survival (Figure **??**).

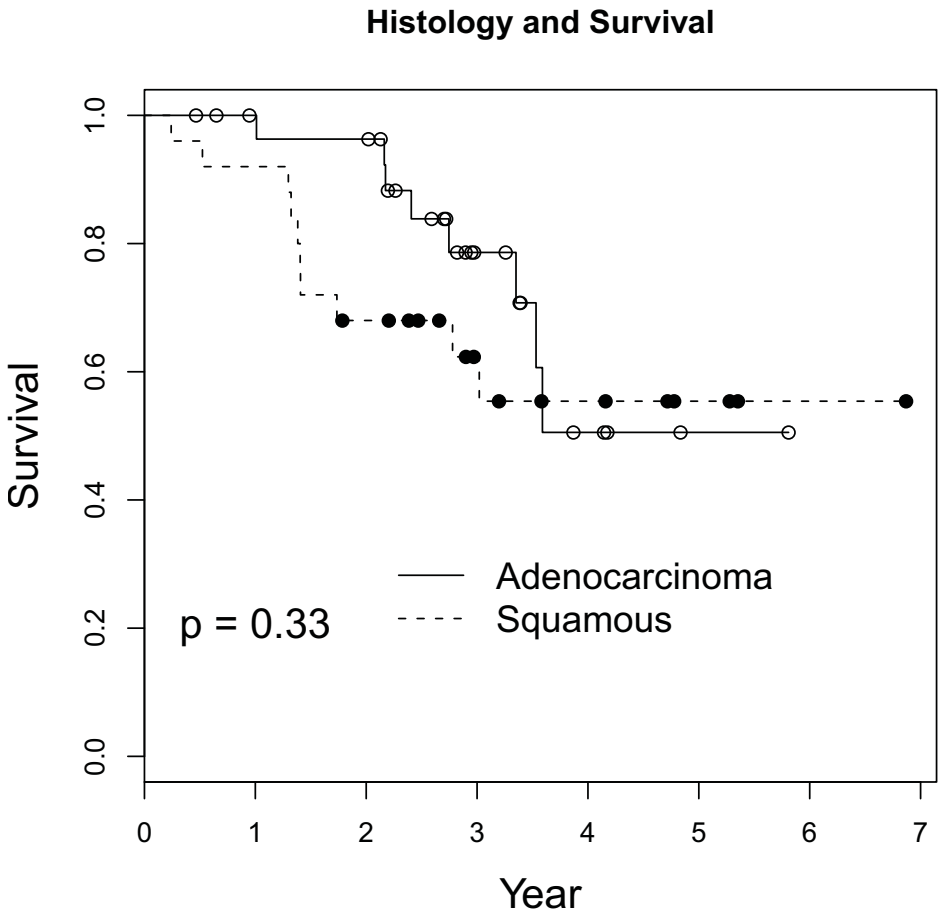


Figure 1: Kaplan Meier plots of survival for different histology

4 FFPE data hierarchical clustering

In this section, we used unsupervised classification to separate patients into 2 groups.

```
> hc <- hclust(dist(t(expr)))
> group <- cutree(hc, 2)
> table(clin$Histology, group)

               group
                 1  2
  Adenocarcinoma  5 25
  Squamous       23  2

> clin$RGS <- group
> sdt <- apply(expr, 1, sd)
> hist(sdt)
> rgb.palette <- colorRampPalette(c("green", "black", "red"), space = "rgb")
```

Heatmap (Figure **??**) only shows the genes with standard deviation greater than 1 to be clear, but note, the cluster dendrogram is generated using all genes.

```
> group1 <- c("High Risk", "Low Risk")[group]
> fit <- survfit(Surv(Death_Time, Death_Event) ~ group1, data = clin)
> fit

Call: survfit(formula = Surv(Death_Time, Death_Event) ~ group1, data = clin)

                 records n.max n.start events median 0.95LCL 0.95UCL
group1=High Risk      28    28      28     14   3.02    2.17      NA
group1=Low Risk       27    27      27      4     NA    3.53      NA

> logrank <- survdiff(Surv(Death_Time, Death_Event) ~ group1, data = clin)
> logrank

Call:
survdiff(formula = Surv(Death_Time, Death_Event) ~ group1, data = clin)

                  N Observed Expected (O-E)^2/E (O-E)^2/V
group1=High Risk 28       14     8.94      2.86      5.72
group1=Low Risk  27        4     9.06      2.82      5.72

 Chisq= 5.7  on 1 degrees of freedom, p= 0.0168
```

5 Predict Survival: FFPE data training to testing

In this section, we will use the FFPE training data to prediction the survival of the FFPE testing data. We will address the variability in the partition of training and testing data by permutation.

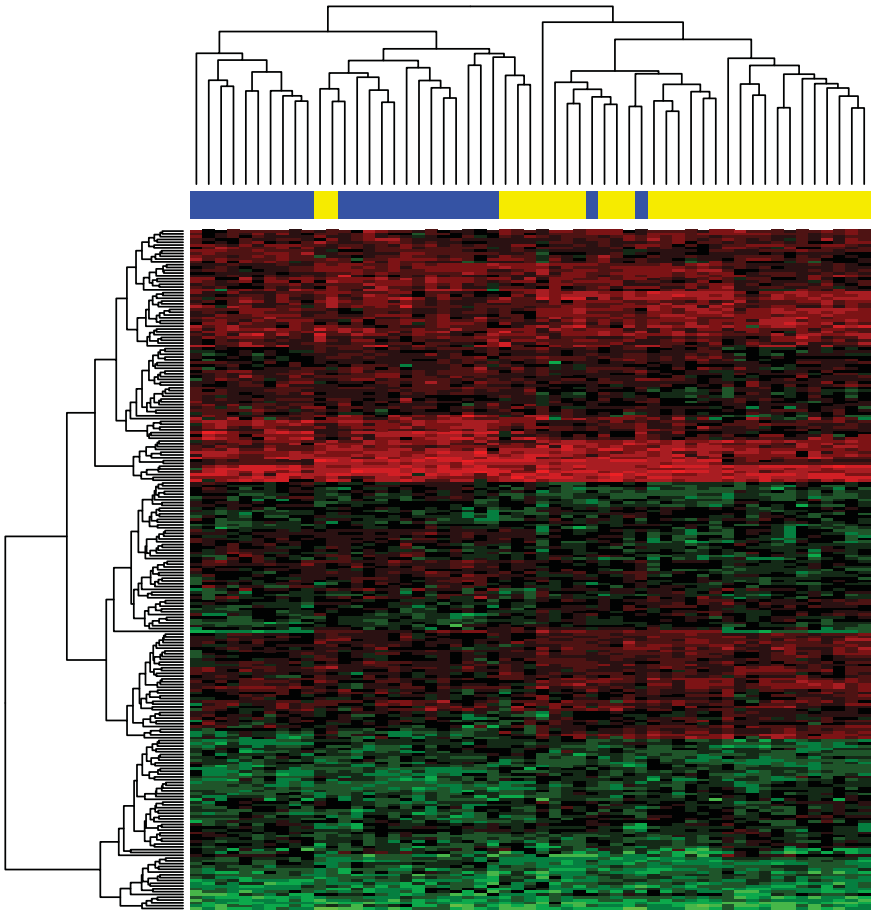


Figure 2: Heatmap of robust gene signature

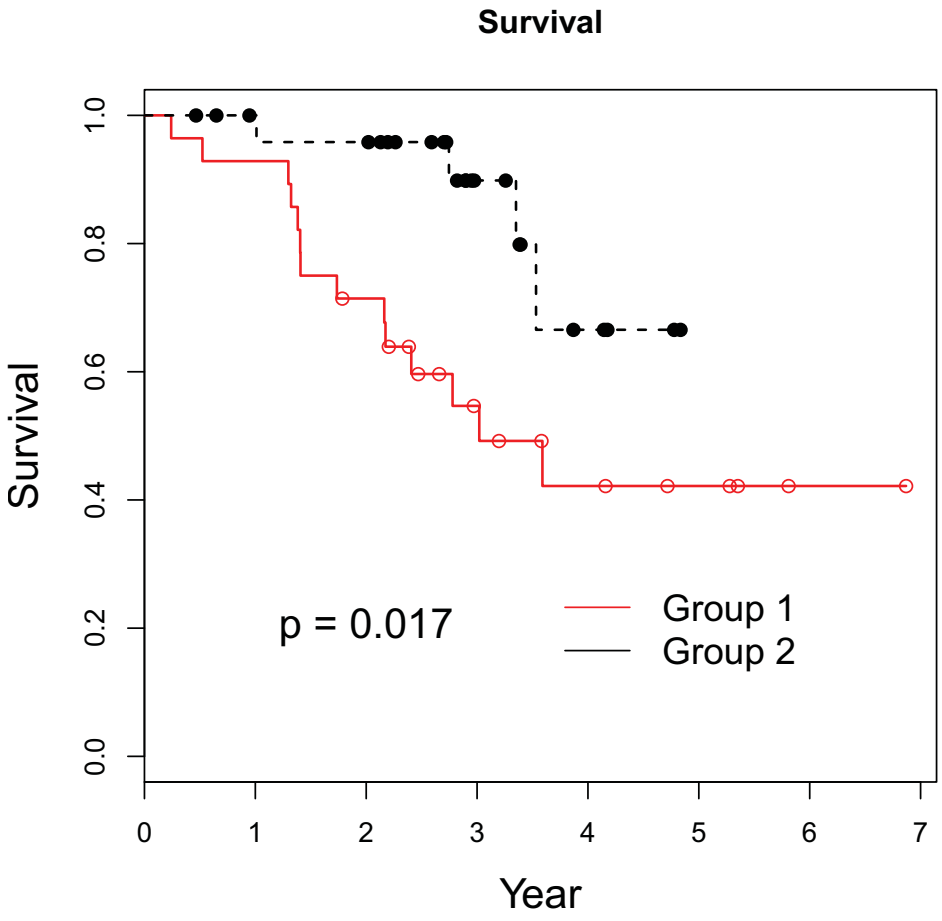
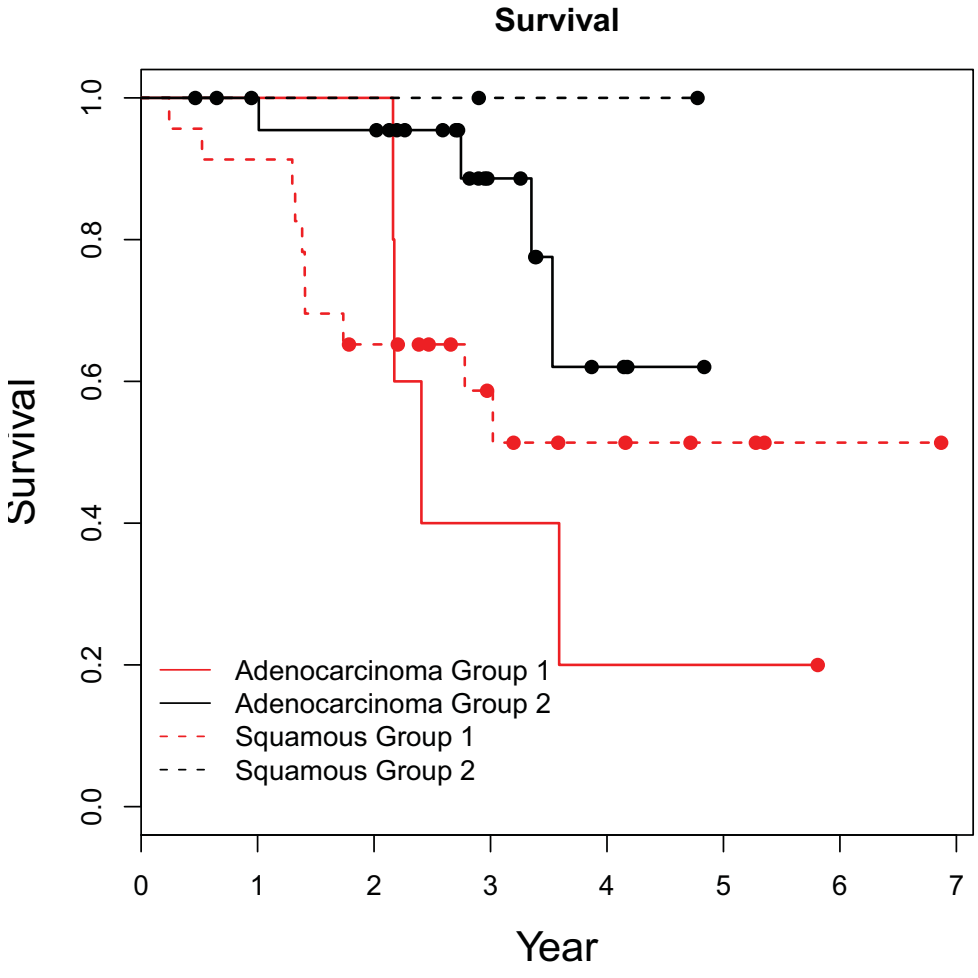


Figure 3: Kaplan Meier plots of survival for different clusters

```
Call: survfit(formula = Surv(Death_Time, Death_Event) ~ group1 + Histology,
    data = clin)

                                          records n.max n.start events median 0.95LCL
group1=High Risk, Histology=Adenocarcinoma       5     5       5      4   2.41    2.17
group1=High Risk, Histology=Squamous            23    23      23     10     NA    1.74
group1=Low Risk, Histology=Adenocarcinoma       25    25      25      4     NA    3.53
group1=Low Risk, Histology=Squamous              2     2       2      0     NA      NA
                                          0.95UCL
group1=High Risk, Histology=Adenocarcinoma      NA
group1=High Risk, Histology=Squamous            NA
group1=Low Risk, Histology=Adenocarcinoma       NA
group1=Low Risk, Histology=Squamous             NA

Call:
survdiff(formula = Surv(Death_Time, Death_Event) ~ group1 + Histology,
    data = clin)

                                           N Observed Expected (O-E)^2/E (O-E)^2/V
group1=High Risk, Histology=Adenocarcinoma  5        4    1.920     2.254     2.536
group1=High Risk, Histology=Squamous       23       10    7.023     1.262     2.076
group1=Low Risk, Histology=Adenocarcinoma  25        4    8.115     2.087     3.828
group1=Low Risk, Histology=Squamous         2        0    0.942     0.942     0.998

 Chisq= 6.6  on 3 degrees of freedom, p= 0.0863
```



```
> ind1 <- 1:25
> ind2 <- 26:55
> data.train <- list(x = expr[, ind1], y = clin$Death_Time[ind1], censoring.status = clin$Death_Event[in
+     featurenames = rownames(expr))
> data.test <- list(x = expr[, ind2], y = clin$Death_Time[ind2], censoring.status = clin$Death_Event[ind
+     featurenames = rownames(expr))
> train.obj <- superpc.train(data.train, type = "survival")
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> fit$v.pred.1df

AGR08.604.CEL AGR08.605.CEL AGR08.606.CEL AGR08.607.CEL AGR08.608.CEL AGR08.609.CEL
            2             2             2             1             2             1
AGR08.613.CEL AGR08.614.CEL AGR08.615.CEL AGR08.616.CEL AGR08.617.CEL AGR08.618.CEL
            2             1             2             1             1             1
AGR08.621.CEL AGR08.622.CEL AGR08.625.CEL AGR08.627.CEL AGR08.629.CEL AGR08.630.CEL
            1             1             1             1             2             2
AGR08.631.CEL AGR08.633.CEL AGR08.634.CEL AGR08.635.CEL AGR08.637.CEL AGR08.638.CEL
            2             1             2             1             2             2
AGR08.640.CEL AGR08.643.CEL AGR08.644.CEL AGR08.666.CEL AGR08.669.CEL AGR08.672.CEL
            2             2             2             2             2             2

> surv.fit <- survfit(Surv(Death_Time, Death_Event) ~ fit$v.pred.1df,
+     data = clin[ind2, ])
> surv.fit

Call: survfit(formula = Surv(Death_Time, Death_Event) ~ fit$v.pred.1df,
    data = clin[ind2, ])

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      12    12      12      0     NA      NA      NA
fit$v.pred.1df=2      18    18      18     10   2.78    2.16      NA

> logrank <- survdiff(Surv(Death_Time, Death_Event) ~ fit$v.pred.1df,
+     data = clin[ind2, ])
> logrank

Call:
survdiff(formula = Surv(Death_Time, Death_Event) ~ fit$v.pred.1df,
    data = clin[ind2, ])

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 12        0     3.83      3.83      6.21
fit$v.pred.1df=2 18       10     6.17      2.37      6.21

 Chisq= 6.2  on 1 degrees of freedom, p= 0.0127
```

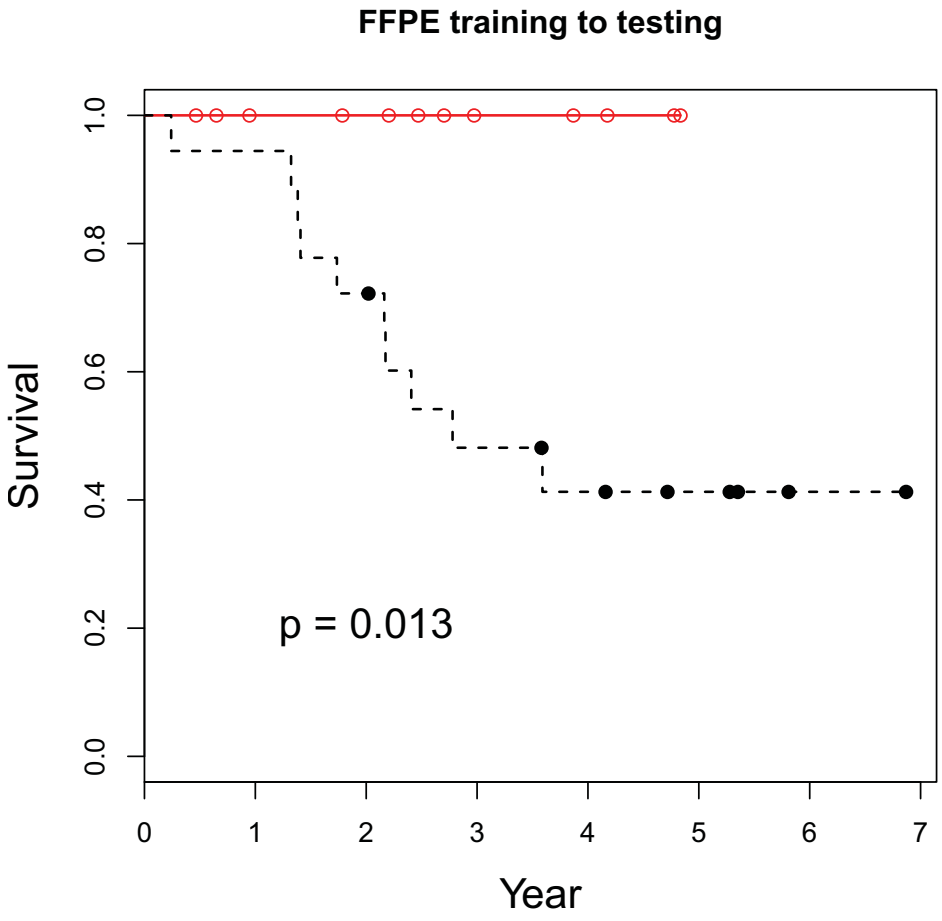


Figure 5: Kaplan Meier plots of survival, with groups predicted from FFPE training to testing

6 Predict Survival: FFPE data training to testing, bootstrap for robustness

```
> BT <- 500
> pv.logrank <- pv.coxph <- rep(NA, BT)
> set.seed(1234)
> for (t in 1:BT) {
+     ind1 <- sample(1:55, 25)
+     ind2 <- setdiff(1:55, ind1)
+     data.train <- list(x = expr[, ind1], y = clin$Death_Time[ind1],
+         censoring.status = clin$Death_Event[ind1], featurenames = NULL)
+     data.test <- list(x = expr[, ind2], y = clin$Death_Time[ind2],
+         censoring.status = clin$Death_Event[ind2], featurenames = NULL)
+     train.obj <- superpc.train(data.train, type = "survival")
+     fail <- try(fit <- superpc.predict(train.obj, data.train, data.test,
+         threshold = 1))
+     pv.coxph[t] <- summary(coxph(Surv(Death_Time, Death_Event) ~ fit$v.pred.1df,
+         data = clin[ind2, ]))$coef[5]
+     fail <- try(fit <- superpc.predict(train.obj, data.train, data.test,
+         threshold = 1, prediction.type = "discrete"))
+     logrank <- survdiff(Surv(Death_Time, Death_Event) ~ fit$v.pred.1df,
+         data = clin[ind2, ])
+     pv.logrank[t] <- pchisq(logrank$chisq, 1, lower.tail = F)
+ }
> hist(pv.logrank, nclass = 30)
> ks.test(pv.logrank, runif(500, 0, 1))

        Two-sample Kolmogorov-Smirnov test

data:  pv.logrank and runif(500, 0, 1)
D = 0.098, p-value = 0.01643
alternative hypothesis: two-sided

> summary(pv.logrank)

    Min.  1st Qu.   Median     Mean  3rd Qu.     Max.
0.001418 0.195900 0.406400 0.441500 0.674700 0.998200
```

7 Consortium Training to Consortium testing and FFPE

In this section, we using consortium training data to prediction the consortium testing data and the FFPE data. The consortium training and tesing data are define in the Nature Medicine paper.

We first read the consortium data, align the consortium and FFPE data and do normalization.

```
> beer.clin.dat <- read.csv("Consortium.clin.csv", row.names = 1)
> beer.expr <- read.csv("Consortium.expr.csv", row.names = 1)
> head(beer.clin.dat)
```

```
                    MICROARRAY dat.SITE death month      year GENDER AGE_AT_DIAGNOSIS
CL2004110909AA   CL2004110909AA     DFCI     0   110 9.1666667 Female              55
CL2004111002AA   CL2004111002AA     DFCI     0    98 8.1666667 Female              41
CL2004111003AA   CL2004111003AA     DFCI     0   110 9.1666667   Male              47
CL20041110100AA CL20041110100AA     DFCI     0    66 5.5000000   Male              73
CL20041110102AA CL20041110102AA     DFCI     1    29 2.4166667 Female              63
CL20041110103AA CL20041110103AA     DFCI     1     7 0.5833333   Male              72
                smoking stage adj_chemo_YN
CL2004110909AA        1     1            0
CL2004111002AA        1     1            0
CL2004111003AA        1     1            1
CL20041110100AA       0     1           NA
CL20041110102AA       1     2           NA
CL20041110103AA       0     1           NA
```

```
> beer.clin <- beer.clin.dat[colnames(beer.expr), ]
> dim(beer.clin)

[1] 442  10

> dim(beer.expr)

[1] 1012  442

> which(rownames(beer.clin) != colnames(beer.expr))

integer(0)

> ffpe.expr <- expr[rownames(beer.expr), ]
> dim(ffpe.expr)

[1] 1012   55

> cb.expr <- data.frame(ffpe.expr, beer.expr)
> site <- c(rep("FFPE", dim(ffpe.expr)[2]), beer.clin$dat.SITE)
> cb.expr[] <- normalize.quantiles(as.matrix(cb.expr))
> ffpe.expr <- cb.expr[, site == "FFPE"]
> beer.expr <- cb.expr[, site != "FFPE"]
```

Using consortium training (HLM and MI) data to build model, first to predict the consortium testing (DFCI and MSKCC) data, and then predict the FFPE data using the same model.

```
> ind1 <- which(!is.na(beer.clin$year) & beer.clin$dat.SITE %in% c("HLM",
+     "MI"))
> length(ind1)

[1] 254

> data.train <- list(x = beer.expr[, ind1], y = beer.clin$year[ind1],
+     censoring.status = beer.clin$death[ind1], featurenames = NULL)
> train.obj <- superpc.train(data.train, type = "survival")
> ind2 <- which(beer.clin$year < 7 & beer.clin$dat.SITE %in% c("DFCI",
+     "MSKCC"))
> length(ind2)
```

```
[1] 157

> data.test <- list(x = beer.expr[, ind2], y = beer.clin$year[ind2],
+     censoring.status = beer.clin$death[ind2], featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> fit$v.pred.1df

     NCI_U133A_1L      NCI_U133A_2L      NCI_U133A_3L      NCI_U133A_4L
                2                 1                 1                 1
     NCI_U133A_5L      NCI_U133A_6L      NCI_U133A_7L      NCI_U133A_8L
                1                 1                 1                 1
     NCI_U133A_9L     NCI_U133A_11L     NCI_U133A_12L     NCI_U133A_13L
                1                 2                 1                 2
    NCI_U133A_14L     NCI_U133A_15L     NCI_U133A_16L     NCI_U133A_17L
                2                 2                 2                 1
    NCI_U133A_18L     NCI_U133A_19L     NCI_U133A_20L     NCI_U133A_21L
                2                 2                 1                 1
    NCI_U133A_22L     NCI_U133A_24L     NCI_U133A_25L     NCI_U133A_27L
                1                 1                 1                 1
    NCI_U133A_28L     NCI_U133A_29L     NCI_U133A_30L     NCI_U133A_32L
                1                 1                 1                 1
    NCI_U133A_33L     NCI_U133A_34L NCI_U133A_37L_Rep     NCI_U133A_39L
                2                 2                 2                 1
    NCI_U133A_40L     NCI_U133A_41L     NCI_U133A_42L     NCI_U133A_43L
                1                 1                 1                 2
    NCI_U133A_44L     NCI_U133A_45L     NCI_U133A_46L     NCI_U133A_47L
                1                 1                 1                 2
    NCI_U133A_49L     NCI_U133A_50L     NCI_U133A_51L     NCI_U133A_54L
                1                 1                 1                 1
    NCI_U133A_55L     NCI_U133A_56L     NCI_U133A_57L     NCI_U133A_58L
                1                 1                 2                 2
    NCI_U133A_59L     NCI_U133A_60L     NCI_U133A_61L     NCI_U133A_62L
                2                 1                 1                 1
    NCI_U133A_63L     NCI_U133A_64L     NCI_U133A_65L     NCI_U133A_66L
                1                 1                 1                 2
    NCI_U133A_67L     NCI_U133A_68L     NCI_U133A_69L     NCI_U133A_70L
                2                 1                 1                 2
    NCI_U133A_71L     NCI_U133A_72L     NCI_U133A_73L     NCI_U133A_74L
                1                 2                 2                 2
    NCI_U133A_75L     NCI_U133A_76L     NCI_U133A_77L     NCI_U133A_78L
                1                 1                 1                 1
    NCI_U133A_79L     NCI_U133A_80L     NCI_U133A_81L     NCI_U133A_82L
                1                 2                 1                 2
    NCI_U133A_83L     NCI_U133A_85L     NCI_U133A_86L     NCI_U133A_87L
                1                 1                 1                 1
    NCI_U133A_88L     NCI_U133A_89L     NCI_U133A_90L     NCI_U133A_91L
                1                 1                 2                 1
    NCI_U133A_92L     NCI_U133A_93L NCI_U133A_94L_Rep     NCI_U133A_95L
```

```
              2               1               1               2
    NCI_U133A_96L    NCI_U133A_99L   NCI_U133A_100L   NCI_U133A_101L
              1               2               1               1
   NCI_U133A_102L   NCI_U133A_103L   NCI_U133A_106L   NCI_U133A_107L
              1               1               1               2
    NCI_U133A_97L   CL2004111022AA   CL2004111043AA   CL2005060345AA
              1               1               1               1
   CL2004111046AA   CL20041116136AA   CL2004111017AA   CL2004111040AA
              1               1               2               1
   CL2005060338AA   CL2004111048AA   CL2005060327AA   CL20041119167AA
              2               1               2               1
   CL2005060344AA   CL2005060346AA   CL2004111041AA   CL20041116134AA
              2               2               1               2
   CL2005060341AA   CL2004111010AA   CL2005060332AA   CL2005060334AA
              2               1               2               1
   CL2005060343AA   CL20041119175AA   CL2004111045AA   CL2005060333AA
              1               1               1               1
   CL2005060337AA   CL2005060336AA   CL2005060340AA   CL20041116156AA
              1               2               1               1
   CL2005060342AA   CL2004111038AA   CL2005060349AA   CL2005060350AA
              1               1               1               1
   CL2005060351AA   CL2005060352AA   CL2005060353AA   CL2005060354AA
              1               2               1               1
   CL2005060355AA   CL2005060356AA   CL2005060357AA   CL2005060358AA
              1               1               1               1
   CL2005060360AA   CL2004111097AA   CL2004111098AA   CL20041110100AA
              1               1               1               1
  CL20041110102AA   CL20041110103AA   CL20041110104AA   CL20041110107AA
              2               2               1               1
  CL20041110108AA   CL20041110109AA   CL20041110110AA   CL20041110111AA
              1               2               1               1
  CL20041116112AA   CL20041116113AA   CL20041116114AA   CL20041119180AA
              1               1               1               1
  CL20041119181AA   CL20041119182AA   CL20041119183AA   CL20041119184AA
              1               1               2               1
  CL20041119186AA   CL20041119187AA   CL20041119191AA   CL20041119192AA
              2               1               1               1
  CL20041119194AA
              2
```

```
> surv.fit <- survfit(Surv(beer.clin$year[ind2], beer.clin$death[ind2]) ~
+     fit$v.pred.1df)
> logrank <- survdiff(Surv(beer.clin$year[ind2], beer.clin$death[ind2]) ~
+     fit$v.pred.1df)
> logrank

Call:
survdiff(formula = Surv(beer.clin$year[ind2], beer.clin$death[ind2]) ~
    fit$v.pred.1df)
```

```
                   N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 112       42     55.2      3.17      14.5
fit$v.pred.1df=2  45       29     15.8     11.11      14.5

 Chisq= 14.5  on 1 degrees of freedom, p= 0.00014
```

```
> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
```

```
> ind3 <- which(beer.clin$year < 7 & beer.clin$dat.SITE %in% c("DFCI"))
> length(ind3)
```

```
[1] 64
```

```
> data.test <- list(x = beer.expr[, ind3], y = beer.clin$year[ind3],
+     censoring.status = beer.clin$death[ind3], featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> fit$v.pred.1df
```

```
 CL2004111022AA  CL2004111043AA  CL2005060345AA  CL2004111046AA CL20041116136AA
              1               1               1               1               1
 CL2004111017AA  CL2004111040AA  CL2005060338AA  CL2004111048AA  CL2005060327AA
              2               1               2               1               2
CL20041119167AA  CL2005060344AA  CL2005060346AA  CL2004111041AA CL20041116134AA
              1               2               2               1               2
 CL2005060341AA  CL2004111010AA  CL2005060332AA  CL2005060334AA  CL2005060343AA
              2               1               2               1               1
CL20041119175AA  CL2004111045AA  CL2005060333AA  CL2005060337AA  CL2005060336AA
              1               1               1               1               2
 CL2005060340AA CL20041116156AA  CL2005060342AA  CL2004111038AA  CL2005060349AA
              1               1               1               1               1
 CL2005060350AA  CL2005060351AA  CL2005060352AA  CL2005060353AA  CL2005060354AA
              1               1               2               1               1
 CL2005060355AA  CL2005060356AA  CL2005060357AA  CL2005060358AA  CL2005060360AA
              1               1               1               1               1
 CL2004111097AA  CL2004111098AA CL20041110100AA CL20041110102AA CL20041110103AA
              1               1               1               2               2
CL20041110104AA CL20041110107AA CL20041110108AA CL20041110109AA CL20041110110AA
              1               1               1               2               1
CL20041110111AA CL20041116112AA CL20041116113AA CL20041116114AA CL20041119180AA
              1               1               1               1               1
CL20041119181AA CL20041119182AA CL20041119183AA CL20041119184AA CL20041119186AA
              1               1               2               1               2
CL20041119187AA CL20041119191AA CL20041119192AA CL20041119194AA
              1               1               1               2
```

```
> surv.fit <- survfit(Surv(beer.clin$year[ind3], beer.clin$death[ind3]) ~
+     fit$v.pred.1df)
```

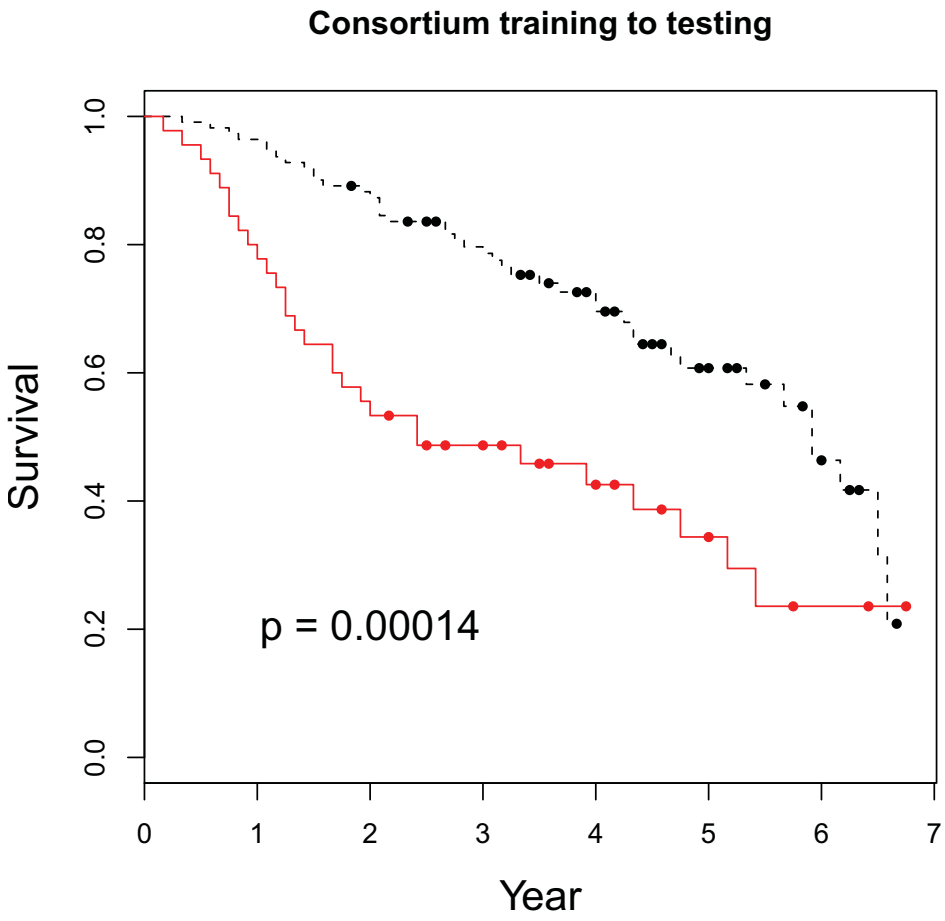


Figure 6: Kaplan Meier plots of survival, with groups predicted from consortium training to consortium testing

```
> logrank <- survdiff(Surv(beer.clin$year[ind3], beer.clin$death[ind3]) ~
+     fit$v.pred.1df)
> logrank

Call:
survdiff(formula = Surv(beer.clin$year[ind3], beer.clin$death[ind3]) ~
    fit$v.pred.1df)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 48       23    28.24     0.973       5.9
fit$v.pred.1df=2 16       11     5.76     4.769       5.9

 Chisq= 5.9  on 1 degrees of freedom, p= 0.0151

> pv <- pchisq(logrank$chisq, 1, lower.tail = F)


> ind4 <- which(beer.clin$year < 7 & beer.clin$dat.SITE %in% c("MSKCC"))
> length(ind4)

[1] 93

> data.test <- list(x = beer.expr[, ind4], y = beer.clin$year[ind4],
+     censoring.status = beer.clin$death[ind4], featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> fit$v.pred.1df

     NCI_U133A_1L      NCI_U133A_2L      NCI_U133A_3L      NCI_U133A_4L
                2                 1                 1                 1
     NCI_U133A_5L      NCI_U133A_6L      NCI_U133A_7L      NCI_U133A_8L
                1                 1                 1                 1
     NCI_U133A_9L     NCI_U133A_11L     NCI_U133A_12L     NCI_U133A_13L
                1                 2                 1                 2
    NCI_U133A_14L     NCI_U133A_15L     NCI_U133A_16L     NCI_U133A_17L
                2                 2                 2                 1
    NCI_U133A_18L     NCI_U133A_19L     NCI_U133A_20L     NCI_U133A_21L
                2                 2                 1                 1
    NCI_U133A_22L     NCI_U133A_24L     NCI_U133A_25L     NCI_U133A_27L
                1                 1                 1                 1
    NCI_U133A_28L     NCI_U133A_29L     NCI_U133A_30L     NCI_U133A_32L
                1                 1                 1                 1
    NCI_U133A_33L     NCI_U133A_34L NCI_U133A_37L_Rep     NCI_U133A_39L
                2                 2                 2                 1
    NCI_U133A_40L     NCI_U133A_41L     NCI_U133A_42L     NCI_U133A_43L
                1                 1                 1                 2
    NCI_U133A_44L     NCI_U133A_45L     NCI_U133A_46L     NCI_U133A_47L
                1                 1                 1                 2
    NCI_U133A_49L     NCI_U133A_50L     NCI_U133A_51L     NCI_U133A_54L
                1                 1                 1                 1
```

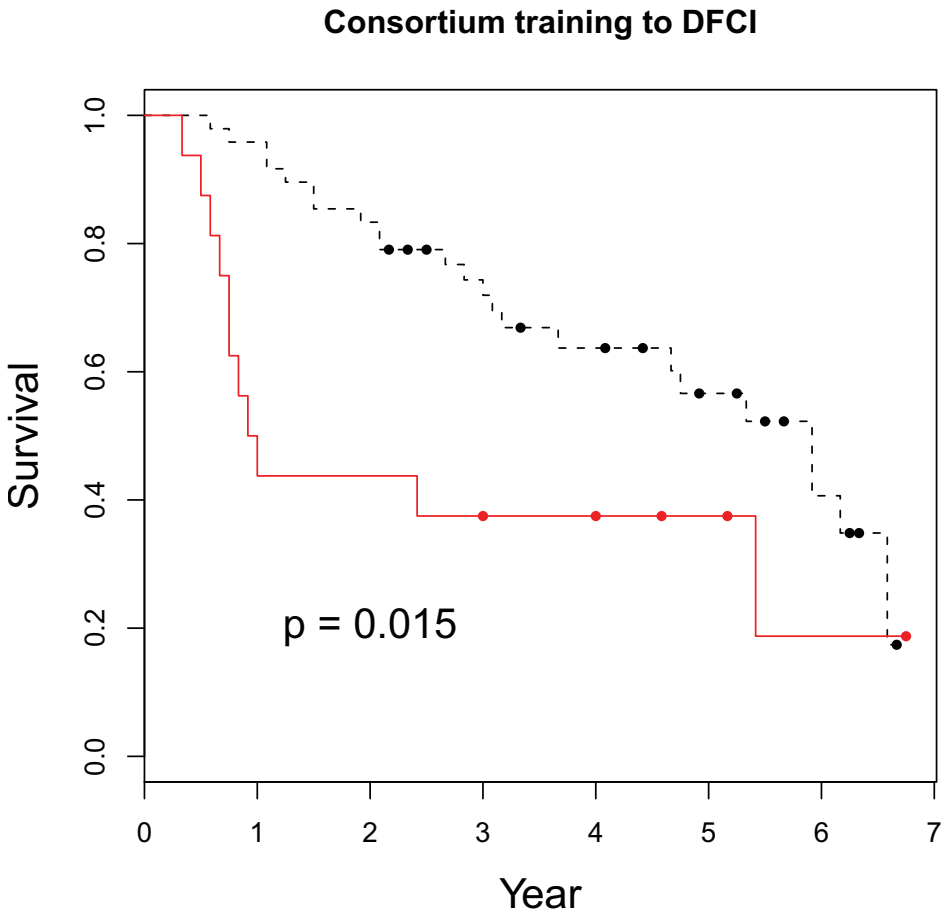


Figure 7: Kaplan Meier plots of survival, with groups predicted from consortium training to DFCI site

```
    NCI_U133A_55L     NCI_U133A_56L     NCI_U133A_57L     NCI_U133A_58L
                1                 1                 2                 2
    NCI_U133A_59L     NCI_U133A_60L     NCI_U133A_61L     NCI_U133A_62L
                2                 1                 1                 1
    NCI_U133A_63L     NCI_U133A_64L     NCI_U133A_65L     NCI_U133A_66L
                1                 1                 1                 2
    NCI_U133A_67L     NCI_U133A_68L     NCI_U133A_69L     NCI_U133A_70L
                2                 1                 1                 2
    NCI_U133A_71L     NCI_U133A_72L     NCI_U133A_73L     NCI_U133A_74L
                1                 2                 2                 2
    NCI_U133A_75L     NCI_U133A_76L     NCI_U133A_77L     NCI_U133A_78L
                1                 1                 1                 1
    NCI_U133A_79L     NCI_U133A_80L     NCI_U133A_81L     NCI_U133A_82L
                1                 2                 1                 2
    NCI_U133A_83L     NCI_U133A_85L     NCI_U133A_86L     NCI_U133A_87L
                1                 1                 1                 1
    NCI_U133A_88L     NCI_U133A_89L     NCI_U133A_90L     NCI_U133A_91L
                1                 1                 2                 1
    NCI_U133A_92L     NCI_U133A_93L NCI_U133A_94L_Rep     NCI_U133A_95L
                2                 1                 1                 2
    NCI_U133A_96L     NCI_U133A_99L    NCI_U133A_100L    NCI_U133A_101L
                1                 2                 1                 1
   NCI_U133A_102L    NCI_U133A_103L    NCI_U133A_106L    NCI_U133A_107L
                1                 1                 1                 2
    NCI_U133A_97L
                1
```

```
> surv.fit <- survfit(Surv(beer.clin$year[ind4], beer.clin$death[ind4]) ~
+     fit$v.pred.1df)
> logrank <- survdiff(Surv(beer.clin$year[ind4], beer.clin$death[ind4]) ~
+     fit$v.pred.1df)
> logrank

Call:
survdiff(formula = Surv(beer.clin$year[ind4], beer.clin$death[ind4]) ~
    fit$v.pred.1df)

                  N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 64       19    27.66      2.71      11.0
fit$v.pred.1df=2 29       18     9.34      8.03      11.0

 Chisq= 11  on 1 degrees of freedom, p= 0.000899

> pv <- pchisq(logrank$chisq, 1, lower.tail = F)


> ind1 <- which(!is.na(beer.clin$year))
> length(ind1)

[1] 440
```

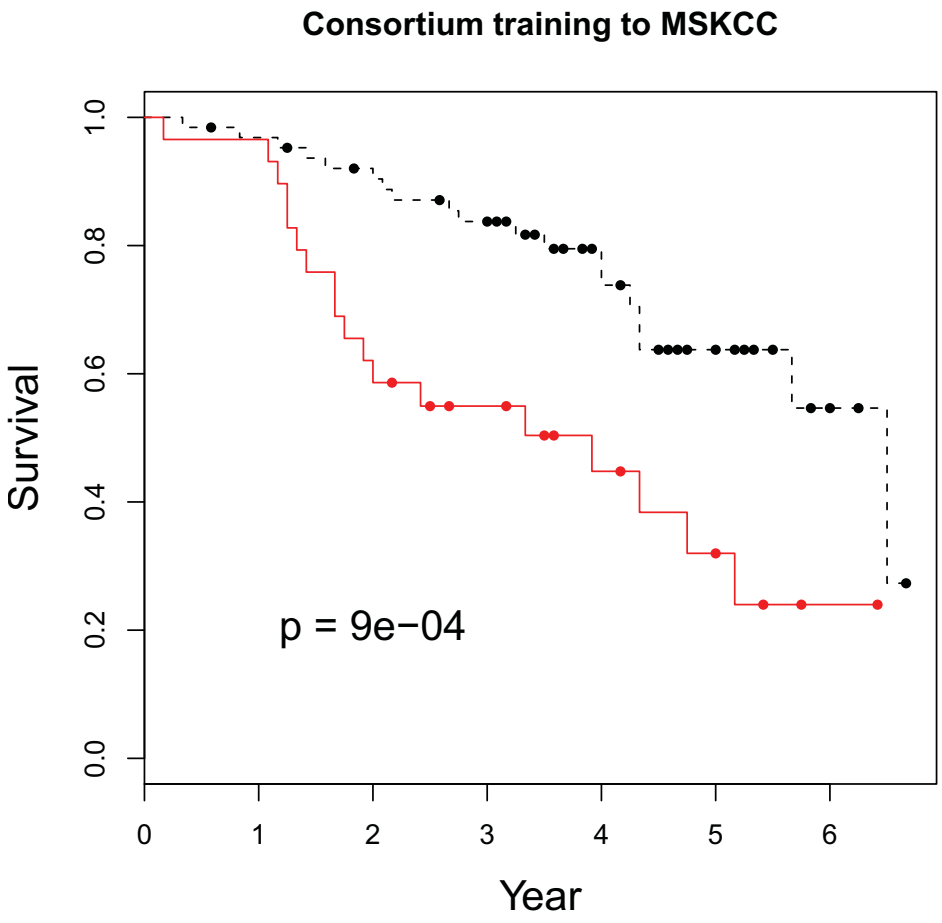


Figure 8: Kaplan Meier plots of survival, with groups predicted from consortium training to MSKCC site

```
> data.train <- list(x = beer.expr[, ind1], y = beer.clin$year[ind1],
+     censoring.status = beer.clin$death[ind1], featurenames = NULL)
> train.obj <- superpc.train(data.train, type = "survival")
> data.test <- list(x = ffpe.expr, y = clin$Death_Time, censoring.status = clin$Death_Event,
+     featurenames = rownames(ffpe.expr))
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> fit$v.pred.1df
```

```
AGR08.564.CEL AGR08.567.CEL AGR08.568.CEL AGR08.571.CEL AGR08.574.CEL AGR08.576.CEL
            2             1             2             2             1             1
AGR08.577.CEL AGR08.579.CEL AGR08.580.CEL AGR08.581.CEL AGR08.582.CEL AGR08.583.CEL
            1             2             2             2             2             1
AGR08.584.CEL AGR08.586.CEL AGR08.588.CEL AGR08.589.CEL AGR08.591.CEL AGR08.593.CEL
            1             2             1             2             1             2
AGR08.595.CEL AGR08.597.CEL AGR08.598.CEL AGR08.600.CEL AGR08.601.CEL AGR08.602.CEL
            2             2             1             2             1             2
AGR08.603.CEL AGR08.604.CEL AGR08.605.CEL AGR08.606.CEL AGR08.607.CEL AGR08.608.CEL
            2             2             2             2             2             2
AGR08.609.CEL AGR08.613.CEL AGR08.614.CEL AGR08.615.CEL AGR08.616.CEL AGR08.617.CEL
            2             2             1             2             1             2
AGR08.618.CEL AGR08.621.CEL AGR08.622.CEL AGR08.625.CEL AGR08.627.CEL AGR08.629.CEL
            1             2             2             1             1             2
AGR08.630.CEL AGR08.631.CEL AGR08.633.CEL AGR08.634.CEL AGR08.635.CEL AGR08.637.CEL
            2             2             1             2             2             2
AGR08.638.CEL AGR08.640.CEL AGR08.643.CEL AGR08.644.CEL AGR08.666.CEL AGR08.669.CEL
            2             2             2             2             2             2
AGR08.672.CEL
            2
```

```
> table(fit$v.pred.1df)
```

```
 1  2
16 39
```

```
> surv.fit <- survfit(Surv(clin$Death_Time, clin$Death_Event) ~ fit$v.pred.1df)
> logrank <- survdiff(Surv(clin$Death_Time, clin$Death_Event) ~ fit$v.pred.1df)
> logrank
```

```
Call:
survdiff(formula = Surv(clin$Death_Time, clin$Death_Event) ~
    fit$v.pred.1df)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 16        2     5.57      2.29      3.32
fit$v.pred.1df=2 39       16    12.43      1.02      3.32

 Chisq= 3.3  on 1 degrees of freedom, p= 0.0684
```

```
> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
```

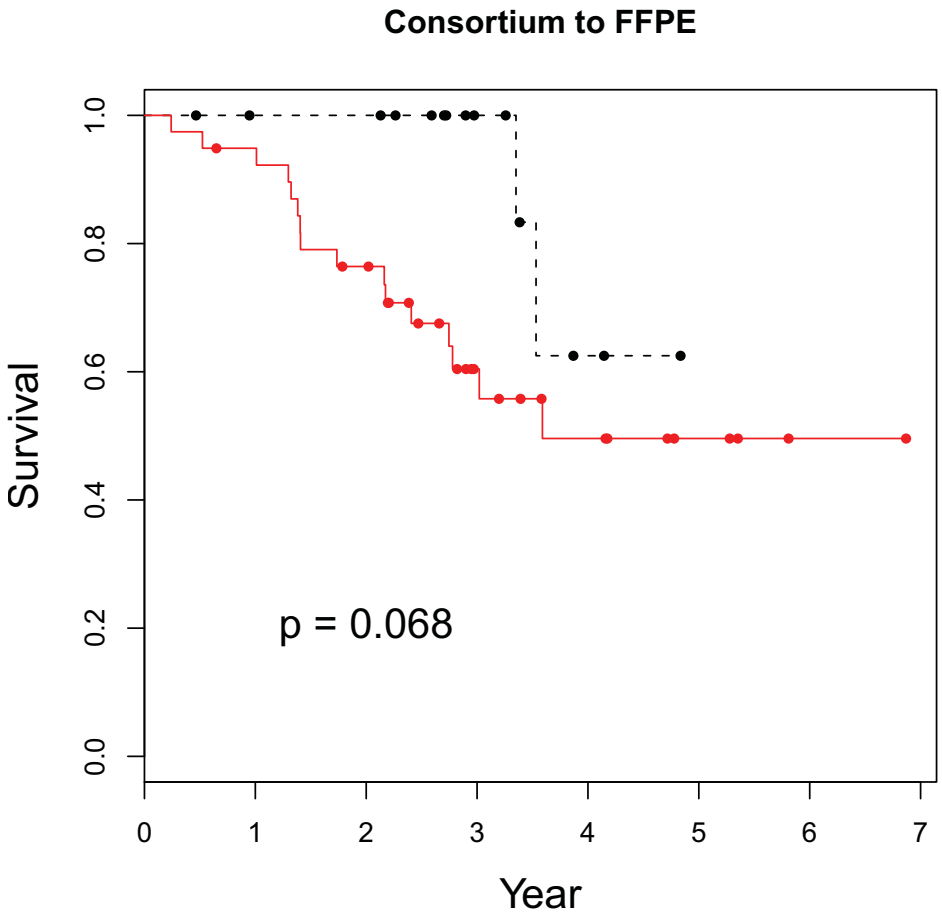


Figure 9: Kaplan Meier plots of survival, with groups predicted from consortium training to ffpe

8 FFPE predict Consortium

In this section, we built prediction model from FFPE data, and used the model to predict the Consortium dataset. Then we test the prediction performance for the whole consortium data, for each individual site, for different stages and with/without Chemo. We define a function to simply the code.

```
> data.train <- list(x = ffpe.expr, y = clin$Death_Time, censoring.status = clin$Death_Event,
+     featurenames = rownames(ffpe.expr))
> train.obj <- superpc.train(data.train, type = "survival")
> data.test <- list(x = beer.expr, y = beer.clin$year, censoring.status = beer.clin$death,
+     featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> fit$v.pred.1df
```

NCI_U133A_1L	NCI_U133A_2L	NCI_U133A_3L	NCI_U133A_4L
2	1	1	1
NCI_U133A_5L	NCI_U133A_6L	NCI_U133A_7L	NCI_U133A_8L
1	1	1	1
NCI_U133A_9L	NCI_U133A_10L	NCI_U133A_11L	NCI_U133A_12L
1	1	2	2
NCI_U133A_13L	NCI_U133A_14L	NCI_U133A_15L	NCI_U133A_16L
2	2	2	2
NCI_U133A_17L	NCI_U133A_18L	NCI_U133A_19L	NCI_U133A_20L
1	2	2	1
NCI_U133A_21L	NCI_U133A_22L	NCI_U133A_23L	NCI_U133A_24L
1	1	2	1
NCI_U133A_25L	NCI_U133A_26L	NCI_U133A_27L	NCI_U133A_28L
1	2	1	2
NCI_U133A_29L	NCI_U133A_30L	NCI_U133A_31L	NCI_U133A_32L
2	1	2	1
NCI_U133A_33L	NCI_U133A_34L	NCI_U133A_35L	NCI_U133A_37L_Rep
2	2	1	2
NCI_U133A_38L	NCI_U133A_39L	NCI_U133A_40L	NCI_U133A_41L
1	1	1	1
NCI_U133A_42L	NCI_U133A_43L	NCI_U133A_44L	NCI_U133A_45L
1	2	1	1
NCI_U133A_46L	NCI_U133A_47L	NCI_U133A_48L	NCI_U133A_49L
1	2	1	1
NCI_U133A_50L	NCI_U133A_51L	NCI_U133A_52L	NCI_U133A_53L
1	1	1	2
NCI_U133A_54L	NCI_U133A_55L	NCI_U133A_56L	NCI_U133A_57L
1	2	1	2
NCI_U133A_58L	NCI_U133A_59L	NCI_U133A_60L	NCI_U133A_61L
1	2	1	1
NCI_U133A_62L	NCI_U133A_63L	NCI_U133A_64L	NCI_U133A_65L
1	1	1	1
NCI_U133A_66L	NCI_U133A_67L	NCI_U133A_68L	NCI_U133A_69L
2	2	1	1

NCI_U133A_70L	NCI_U133A_71L	NCI_U133A_72L	NCI_U133A_73L
1	1	2	2
NCI_U133A_74L	NCI_U133A_75L	NCI_U133A_76L	NCI_U133A_77L
2	1	1	1
NCI_U133A_78L	NCI_U133A_79L	NCI_U133A_80L	NCI_U133A_81L
1	1	2	1
NCI_U133A_82L	NCI_U133A_83L	NCI_U133A_84L	NCI_U133A_85L
2	1	1	1
NCI_U133A_86L	NCI_U133A_87L	NCI_U133A_88L	NCI_U133A_89L
1	1	1	1
NCI_U133A_90L	NCI_U133A_91L	NCI_U133A_92L	NCI_U133A_93L
2	1	2	1
NCI_U133A_94L_Rep	NCI_U133A_95L	NCI_U133A_96L	NCI_U133A_98L
1	1	1	1
NCI_U133A_99L	NCI_U133A_100L	NCI_U133A_101L	NCI_U133A_102L
1	1	1	1
NCI_U133A_103L	NCI_U133A_106L	NCI_U133A_107L	NCI_U133A_97L
1	1	1	1
CL2004111013AA	CL2004111022AA	CL2004111042AA	CL2004111002AA
1	1	1	1
CL2004111043AA	CL2005060326AA	CL2005060345AA	CL2004111046AA
1	1	1	1
CL20041116136AA	CL2004111003AA	CL2004110909AA	CL2004111017AA
1	1	1	2
CL2004111033AA	CL2004111040AA	CL2005060338AA	CL2004111044AA
1	1	2	2
CL20041116127AA	CL2004111048AA	CL2005060327AA	CL20041119167AA
1	2	2	1
CL2005060344AA	CL2005060346AA	CL2004111041AA	CL20041116126AA
2	2	1	2
CL20041116134AA	CL2005060341AA	CL2004111010AA	CL2004111018AA
2	2	1	1
CL2005060339AA	CL2005060332AA	CL2005060334AA	CL2005060343AA
2	2	1	1
CL20041119175AA	CL2004111045AA	CL2005060328AA	CL2005060329AA
1	1	1	1
CL2005060333AA	CL2005060337AA	CL2005060336AA	CL2005060340AA
2	1	2	1
CL20041116156AA	CL2005060342AA	CL2004111038AA	CL2005060348AA
1	1	1	2
CL2005060349AA	CL2005060350AA	CL2005060351AA	CL2005060352AA
1	1	1	2
CL2005060353AA	CL2005060354AA	CL2005060355AA	CL2005060356AA
1	1	1	1
CL2005060357AA	CL2005060358AA	CL2005060360AA	CL2004111097AA
1	1	1	1
CL2004111098AA	CL20041110100AA	CL20041110102AA	CL20041110103AA
1	1	1	2

CL20041110104AA	CL20041110107AA	CL20041110108AA	CL20041110109AA
1	1	1	1
CL20041110110AA	CL20041110111AA	CL20041116112AA	CL20041116113AA
1	1	1	1
CL20041116114AA	CL20041119179AA	CL20041119180AA	CL20041119181AA
1	1	1	1
CL20041119182AA	CL20041119183AA	CL20041119184AA	CL20041119185AA
1	2	1	1
CL20041119186AA	CL20041119187AA	CL20041119190AA	CL20041119191AA
1	1	1	1
CL20041119192AA	CL20041119194AA	CL2004113027AA	NCI_Lung_1_U133A
1	2	2	1
NCI_Lung_3_U133A	NCI_Lung_4_U133A	NCI_Lung_5_U133A	NCI_Lung_6_U133A
1	1	1	1
NCI_Lung_7_U133A	NCI_Lung_8_U133A	NCI_Lung_10_U133A	NCI_Lung_11_U133A
1	1	1	1
NCI_Lung_12_U133A	NCI_Lung_13_U133A	NCI_Lung_14_U133A	NCI_Lung_15_U133A
1	1	1	1
NCI_Lung_16_U133A	NCI_Lung_17_U133A	NCI_Lung_18_U133A	NCI_Lung_19_U133A
2	2	2	2
NCI_Lung_20_U133A	NCI_Lung_21_U133A	NCI_Lung_22_U133A	NCI_Lung_23_U133A
1	2	1	1
NCI_Lung_25_U133A	NCI_Lung_26_U133A	NCI_Lung_27_U133A	NCI_Lung_28_U133A
2	2	2	2
NCI_Lung_29_U133A	NCI_Lung_30_U133A	NCI_Lung31_U133A	NCI_Lung32_U133A
1	2	1	2
NCI_Lung33_U133A	NCI_Lung34_U133A	NCI_Lung35_U133A	NCI_Lung36_U133A
2	2	2	2
NCI_Lung37_U133A	NCI_Lung38_U133A	NCI_Lung39_U133A	NCI_Lung40_U133A
2	2	1	2
NCI_Lung41_U133A	NCI_Lung42_U133A	NCI_Lung43_U133A	NCI_Lung44_U133A
2	1	1	1
NCI_Lung45_U133A	NCI_Lung46_U133A	NCI_Lung47_U133A	NCI_Lung48_U133A
1	1	2	1
NCI_Lung49_U133A	NCI_Lung50_U133A	NCI_Lung51_U133A	NCI_Lung52_U133A
1	2	2	1
NCI_Lung53_U133A	NCI_Lung54_U133A	NCI_Lung55_U133A	NCI_Lung56_U133A
1	2	2	1
NCI_Lung57_U133A	NCI_Lung58_U133A	NCI_Lung_59_U133A	NCI_Lung_60_U133A
1	2	2	1
NCI_Lung_61_U133A	NCI_Lung_62_U133A	NCI_Lung_63_U133A	NCI_Lung_64_U133A
2	2	1	2
NCI_Lung_65_U133A	NCI_Lung_66_U133A	NCI_Lung_67_U133A	NCI_Lung_68_U133A
2	1	1	1
NCI_Lung_69_U133A	NCI_Lung_70_U133A	NCI_Lung_71_U133A	NCI_Lung_72_U133A
2	2	1	2
NCI_Lung_73_U133A	NCI_Lung_74_U133A	NCI_Lung_75_U133A	NCI_Lung_76_U133A
2	1	1	2

```
 NCI_Lung_77_U133A  NCI_Lung_78_U133A  NCI_Lung_79_U133A  NCI_Lung_80_U133A
                 1                  2                  2                  1
 NCI_Lung_81_U133A  NCI_Lung_82_U133A  NCI_Lung_83_U133A  NCI_Lung_84_U133A
                 1                  1                  1                  1
 NCI_Lung_85_U133A  NCI_Lung_86_U133A  NCI_Lung_89_U133A  NCI_Lung_90_U133A
                 2                  1                  1                  1
 NCI_Lung_91_U133A  NCI_Lung_92_U133A  NCI_Lung_93_U133A  NCI_Lung_95_U133A
                 1                  1                  2                  1
 NCI_Lung_96_U133A  NCI_Lung_97_U133A  NCI_Lung_98_U133A  NCI_Lung_99_U133A
                 1                  1                  1                  1
NCI_Lung_100_U133A  NCI_lung201_U133A  NCI_lung202_U133A  NCI_lung203_U133A
                 2                  1                  2                  2
 NCI_lung204_U133A  NCI_lung205_U133A  NCI_lung206_U133A  NCI_lung207_U133A
                 1                  1                  2                  1
 NCI_lung209_U133A  NCI_lung210_U133A  NCI_lung212_U133A  NCI_lung213_U133A
                 2                  1                  2                  1
 NCI_lung214_U133A  NCI_lung215_U133A  NCI_lung216_U133A  NCI_lung218_U133A
                 1                  2                  2                  1
 NCI_lung219_U133A  NCI_lung220_U133A  NCI_lung221_U133A  NCI_lung222_U133A
                 1                  1                  2                  1
 NCI_lung224_U133A  NCI_lung227_U133A  NCI_lung228_U133A  NCI_lung229_U133A
                 2                  1                  1                  1
 NCI_lung230_U133A  NCI_lung231_U133A  NCI_lung234_U133A  NCI_lung235_U133A
                 2                  1                  2                  2
 NCI_lung236_U133A  NCI_lung237_U133A  NCI_lung238_U133A  NCI_lung239_U133A
                 1                  2                  1                  1
 NCI_lung240_U133A  NCI_lung246_U133A  NCI_lung247_U133A  NCI_lung249_U133A
                 2                  2                  1                  2
 NCI_lung258_U133A  NCI_lung259_U133A  NCI_lung263_U133A  NCI_lung267_U133A
                 2                  1                  1                  2
 NCI_lung268_U133A  NCI_lung269_U133A  NCI_lung270_U133A  NCI_lung271_U133A
                 1                  2                  2                  2
 NCI_lung272_U133A  NCI_lung273_U133A  NCI_lung274_U133A         Moff_0130D
                 2                  2                  2                  2
        Moff_0184A         Moff_0404A         Moff_0480I         Moff_0516E
                 1                  2                  1                  1
        Moff_0604H         Moff_0683H         Moff_0686G         Moff_0711B
                 2                  2                  1                  2
        Moff_0770K         Moff_0928E         Moff_0936G         Moff_1221F
                 2                  1                  1                  1
        Moff_1276H         Moff_1300F         Moff_1343G         Moff_1400H
                 2                  1                  2                  1
        Moff_1473F         Moff_1477G         Moff_1487F         Moff_1514K
                 2                  1                  1                  1
        Moff_1576F         Moff_1734M         Moff_1835I         Moff_1864H
                 1                  2                  1                  2
        Moff_1883G         Moff_1888E         Moff_1903E         Moff_2072G
                 2                  2                  2                  1
```

```
        Moff_2186B         Moff_2199G         Moff_2219F         Moff_2296D
                 1                  2                  1                  2
        Moff_2310K         Moff_2325E         Moff_2333J         Moff_2352G
                 1                  1                  1                  1
        Moff_2362A         Moff_2366C         Moff_2373H         Moff_2401B
                 1                  2                  1                  2
        Moff_2463A         Moff_2470G         Moff_2472B         Moff_2497D
                 2                  1                  1                  1
       Moff_2517G1         Moff_2550F         Moff_2595A         Moff_2603I
                 1                  2                  1                  1
        Moff_2629F         Moff_2634G         Moff_2658C         Moff_2663F
                 1                  1                  1                  2
        Moff_2690H         Moff_2886F         Moff_2955B         Moff_2958D
                 1                  1                  1                  2
        Moff_2996G         Moff_3003G         Moff_3009D         Moff_3053B
                 1                  2                  1                  1
        Moff_3191I         Moff_3274K         Moff_3298E         Moff_3401F
                 2                  2                  2                  2
        Moff_3730B         Moff_4094F         Moff_4112G         Moff_4172B
                 1                  1                  2                  2
        Moff_4217B         Moff_3510G         Moff_0142C         Moff_0291A
                 1                  2                  2                  1
        Moff_0298B         Moff_0542F        Moff_1817G1         Moff_1940G
                 1                  2                  2                  1
        Moff_0170D         Moff_0978B  NCI_lung275_U133A  NCI_lung276_U133A
                 2                  1                  1                  2
 NCI_lung277_U133A  NCI_lung278_U133A  NCI_lung280_U133A   NCI_lung281_U133
                 2                  2                  1                  2
  NCI_lung282_U133   NCI_lung283_U133   NCI_lung285_U133   NCI_lung286_U133
                 2                  2                  1                  1
  NCI_lung289_U133   NCI_lung290_U133   NCI_lung291_U133   NCI_lung292_U133
                 1                  2                  1                  2
  NCI_lung293_U133   NCI_lung294_U133   NCI_lung295_U133   NCI_Lung296_U133
                 2                  2                  1                  2
  NCI_Lung297_U133   NCI_Lung298_U133   NCI_Lung299_U133   NCI_Lung300_U133
                 2                  2                  1                  2
  NCI_Lung304_U133   NCI_Lung305_U133   NCI_Lung306_U133   NCI_Lung307_U133
                 1                  1                  2                  2
  NCI_Lung309_U133   NCI_Lung310_U133   NCI_Lung311_U133   NCI_Lung312_U133
                 2                  2                  2                  2
  NCI_Lung313_U133  NCI_Lung314_U133A  NCI_Lung318_U133A  NCI_Lung319_U133A
                 2                  1                  2                  1
 NCI_Lung320_U133A  NCI_Lung321_U133A
                 1                  2
```

```
> beer.clin$predicted <- fit$v.pred.1df
> write.csv(beer.clin, "beer.clin.pred.csv")
> display.surv <- function(subset, main, tx = 2) {
```

```
+     surv.fit <- survfit(Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
+         subset = subset)
+     logrank <- survdiff(Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
+         subset = subset)
+     pv <- pchisq(logrank$chisq, 1, lower.tail = F)
+     plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+         mark = 20, cex.lab = 1.5, main = main)
+     text(tx, 0.1, pv.expr(pv), cex = 1.5)
+     print(surv.fit)
+ }
```

Kaplan-Meier plots showing the predictive power of the robust gene signature developed from training set (55 FFPE tumor samples) and independently tested from different sets (To avoid the extrapolation of the prediction model, the comparison of survival time between predicted groups are truncated at 7 years):
Figure 8. FFPE to consortium over all.
Figure 9. FFPE to consortium MSKCC.
Figure 10. FFPE to consortium DFCI.
Figure 11. FFPE to consortium MI.
Figure 12. FFPE to consortium HLM.
Figure 13. FFPE to consortium stage I.
Figure 14. FFPE to consortium stage II.
Figure 15. FFPE to consortium stage III.
Figure 16. FFPE to consortium with chemotherapy.
Figure 17. FFPE to consortium without chemotherapy.

```
> display.surv(subset = which(beer.clin$year < 7), main = "FFPE to Consortium")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1    223   223     223    113   5.33    4.33    5.92
fit$v.pred.1df=2    139   139     139    105   2.31    1.74    3.33
```

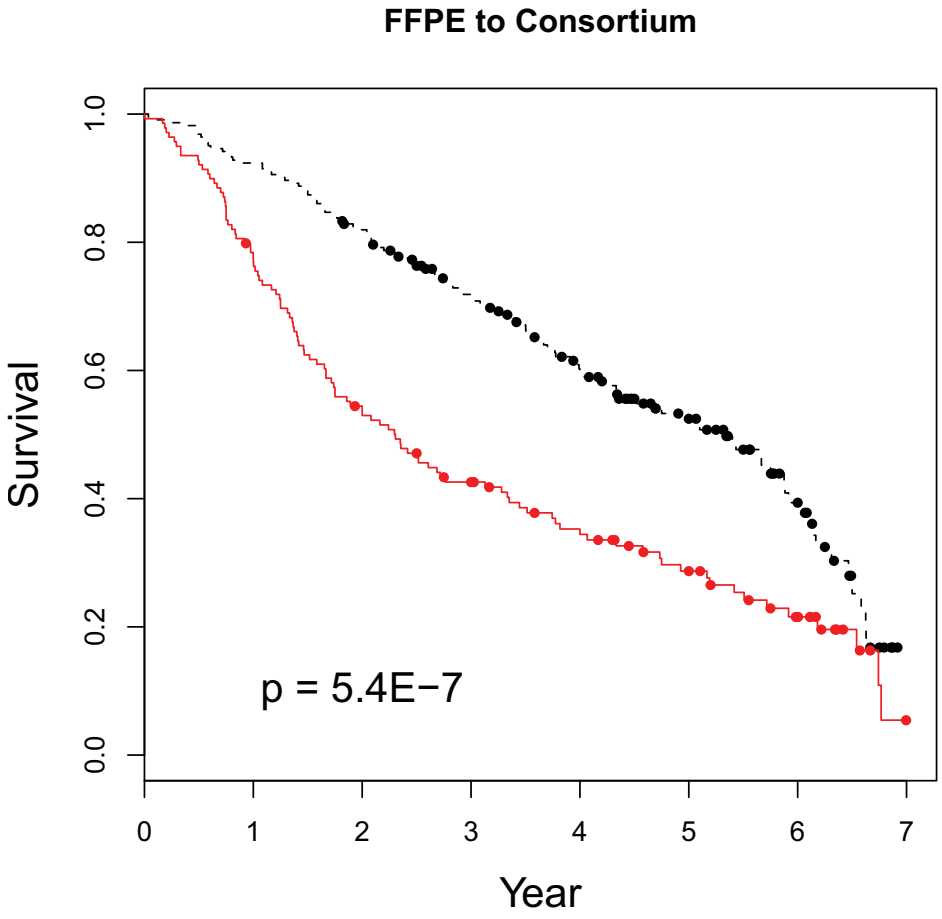


Figure 10: Kaplan Meier plots of survival, with groups predicted from FFPE to Consortium

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$dat.SITE ==
+     "MSKCC"), main = "FFPE to MSKCC")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      65    65      65     20   6.50    4.33      NA
fit$v.pred.1df=2      28    28      28     17   3.33    1.75      NA
```

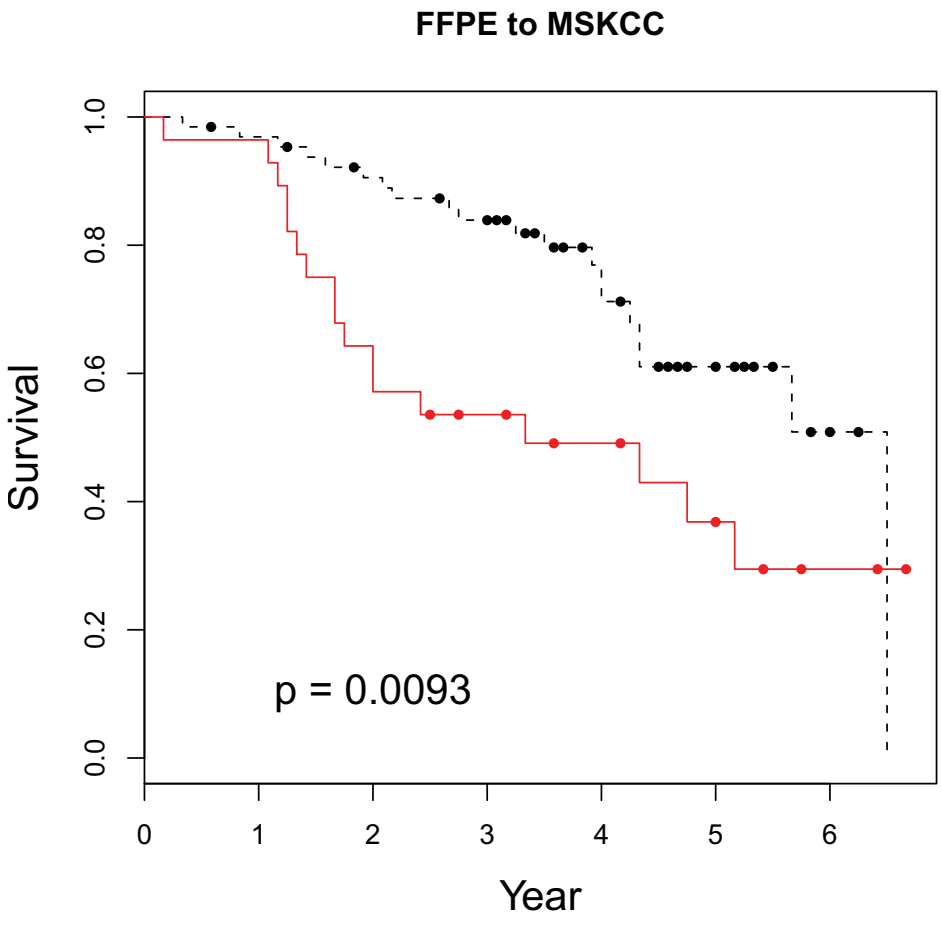


Figure 11: Kaplan Meier plots of survival, with groups predicted from FFPE to MSKCC

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$dat.SITE ==
+     "DFCI"), main = "FFPE to DFCI")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      49    49      49     23  5.917    3.67      NA
fit$v.pred.1df=2      15    15      15     11  0.917    0.75      NA
```

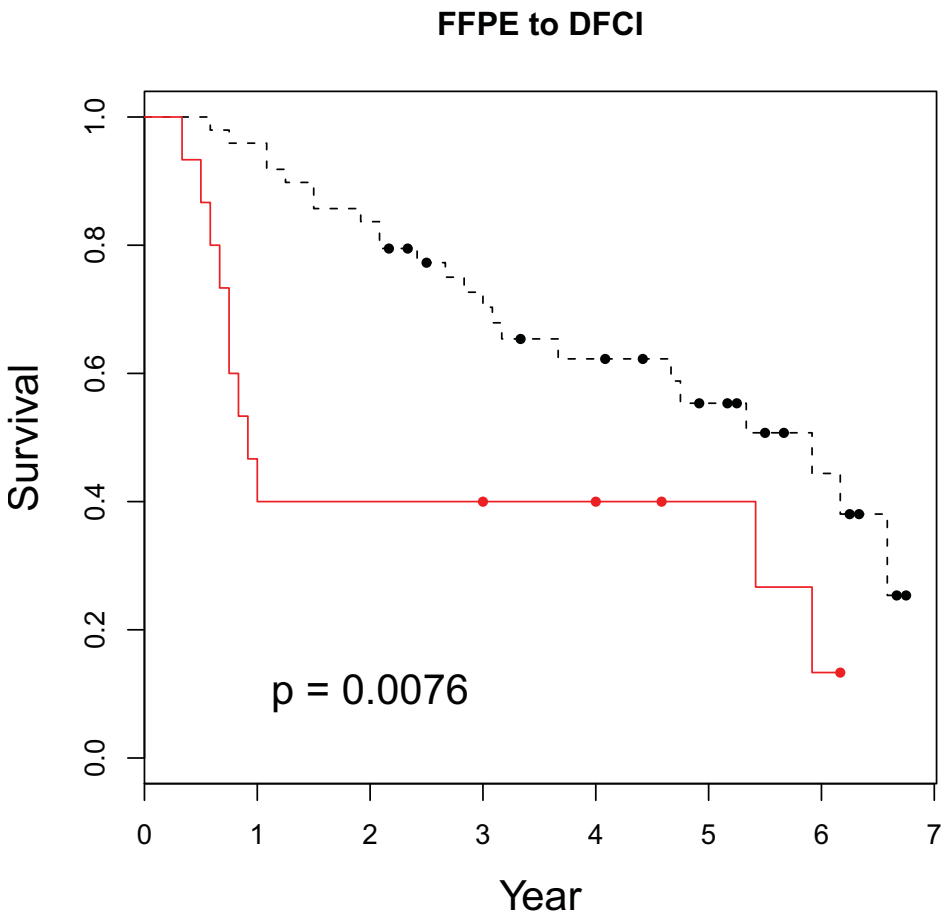


Figure 12: Kaplan Meier plots of survival, with groups predicted from FFPE to DFCI

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$dat.SITE ==
+     "MI"), main = "FFPE to MI")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      70    70      70     38   5.40    3.98    6.14
fit$v.pred.1df=2      66    66      66     52   2.24    1.58    3.82
```

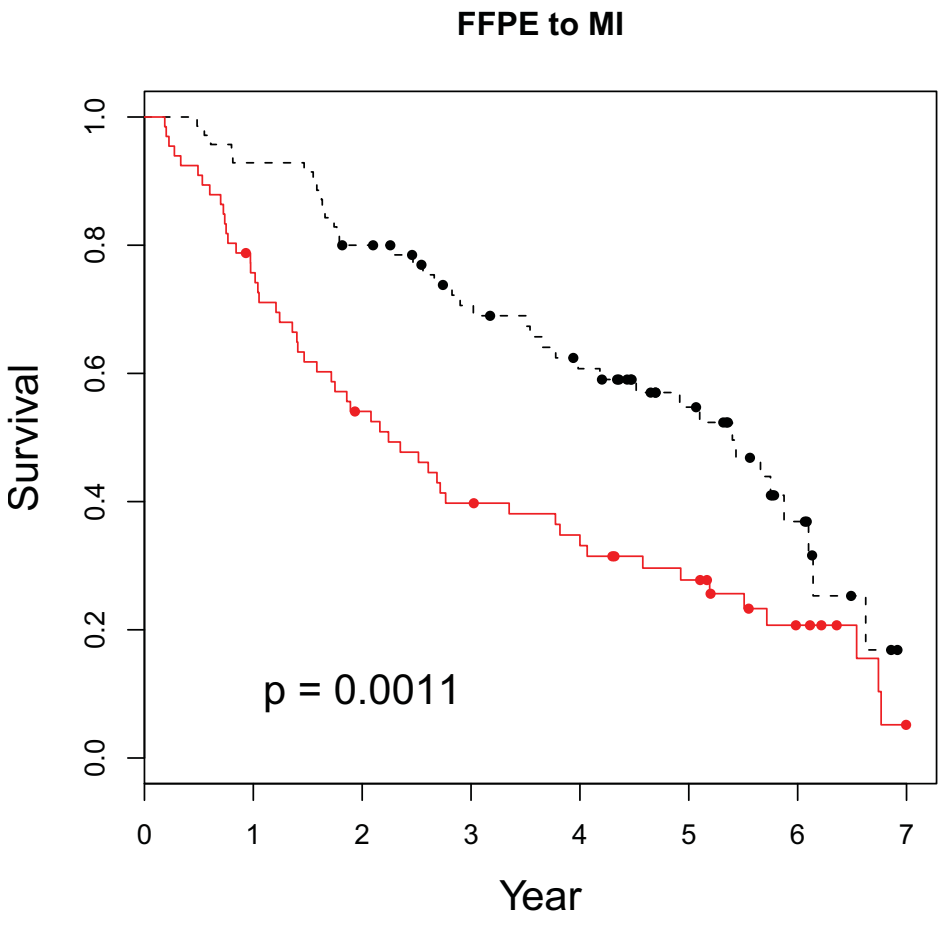


Figure 13: Kaplan Meier plots of survival, with groups predicted from FFPE to MI

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$dat.SITE ==
+     "HLM"), main = "FFPE to HLM")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      39    39      39     32   3.35    2.20    5.10
fit$v.pred.1df=2      30    30      30     25   2.30    1.46    3.52
```

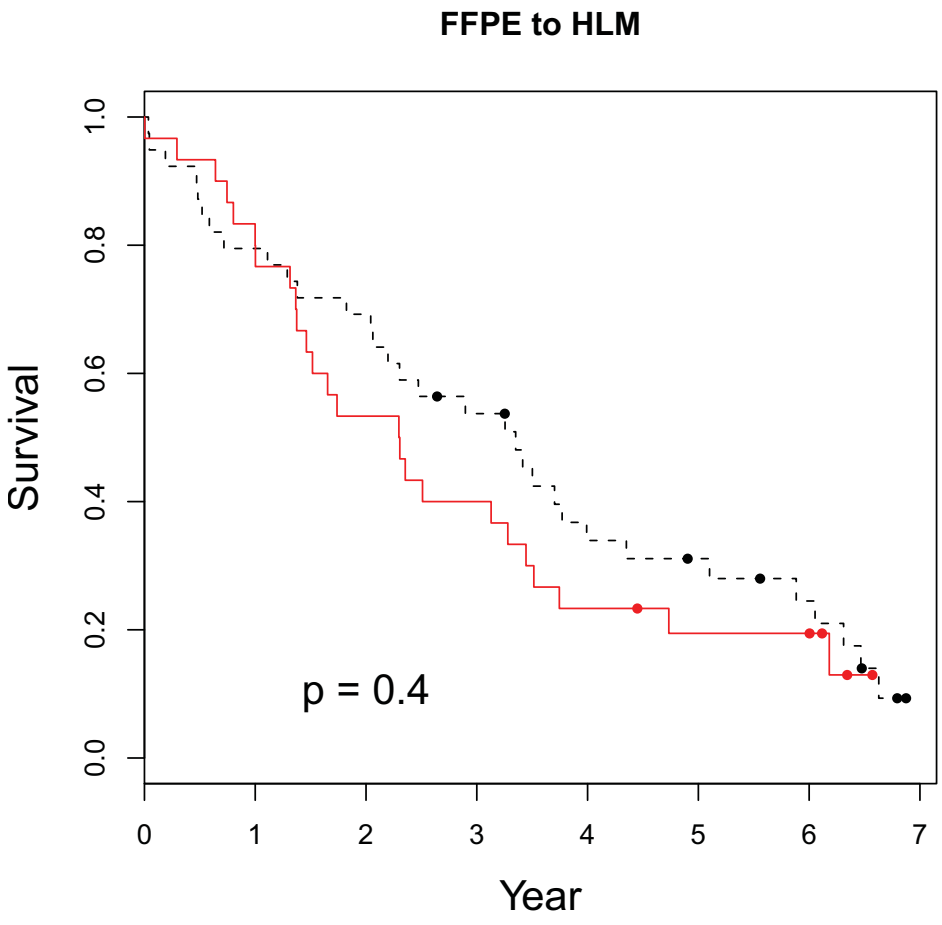


Figure 14: Kaplan Meier plots of survival, with groups predicted from FFPE to HLM

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$stage ==
+     1), main = "FFPE to Consortium, stage 1")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1     150   150     150     61   5.92    5.40    6.58
fit$v.pred.1df=2      65    65      65     38   4.75    3.33      NA
```

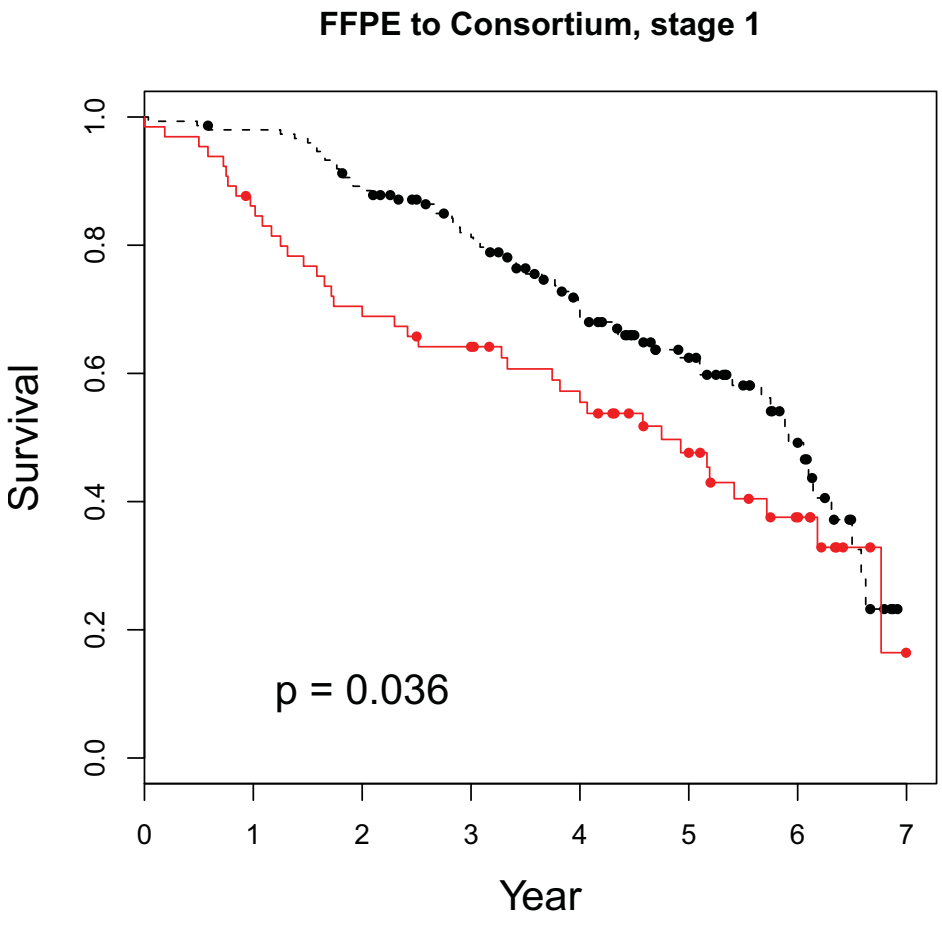


Figure 15: Kaplan Meier plots of survival, with groups predicted from FFPE to Consortium, stage 1

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$stage ==
+     2), main = "FFPE to Consortium, stage 2")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      44    44      44     27   3.67    2.47    6.17
fit$v.pred.1df=2      38    38      38     33   1.75    1.04    3.44
```

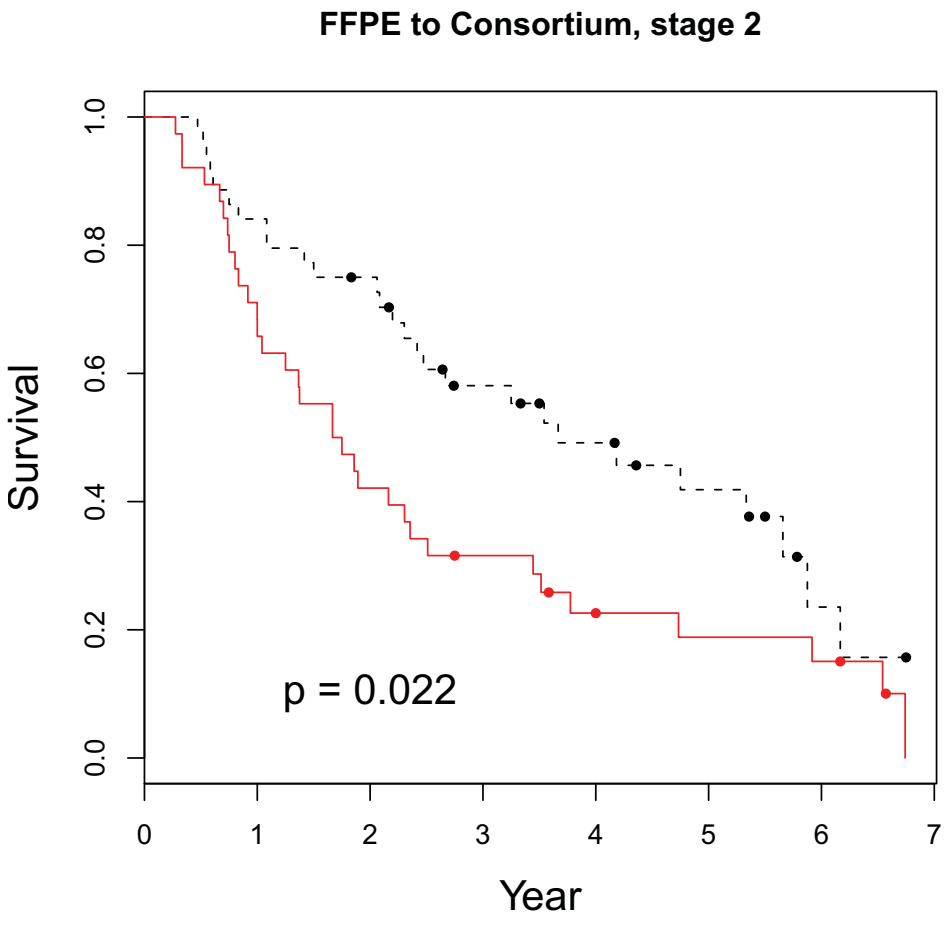


Figure 16: Kaplan Meier plots of survival, with groups predicted from FFPE to Consortium, stage 2

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$stage ==
+     3), main = "FFPE to Consortium, stage 3")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      28    28      28     24   2.17    1.47    4.25
fit$v.pred.1df=2      36    36      36     34   1.41    1.21    2.24
```

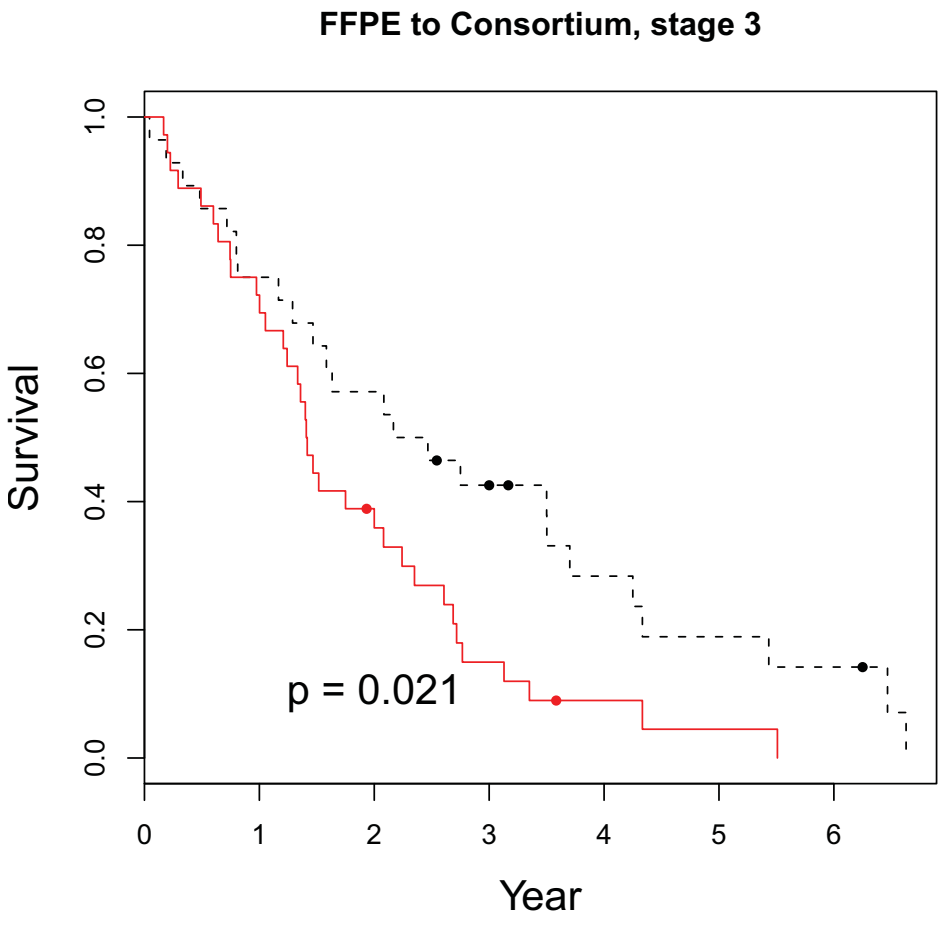


Figure 17: Kaplan Meier plots of survival, with groups predicted from FFPE to Consortium, stage 3

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$adj_chemo_YN ==
+     1), main = "FFPE to Consortium, with Chemo")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      45    45      45     35   3.77    2.47    5.33
fit$v.pred.1df=2      26    26      26     23   1.69    1.37    4.75
```

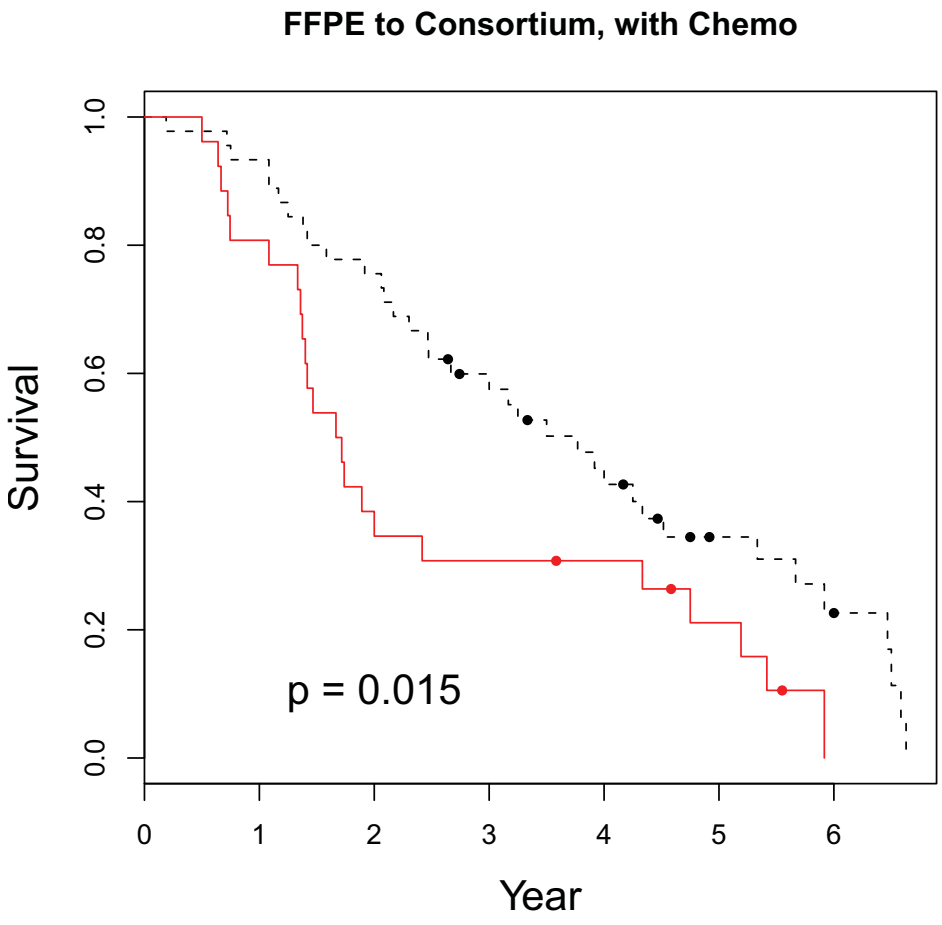


Figure 18: Kaplan Meier plots of survival, with groups predicted from FFPE to Consortium, with Chemo

```
> display.surv(subset = which(beer.clin$year < 7 & beer.clin$adj_chemo_YN ==
+     0), main = "FFPE to Consortium, without Chemo")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1     120   120     120     49   5.88    5.10    6.62
fit$v.pred.1df=2      70    70      70     47   2.51    1.75    4.07
```

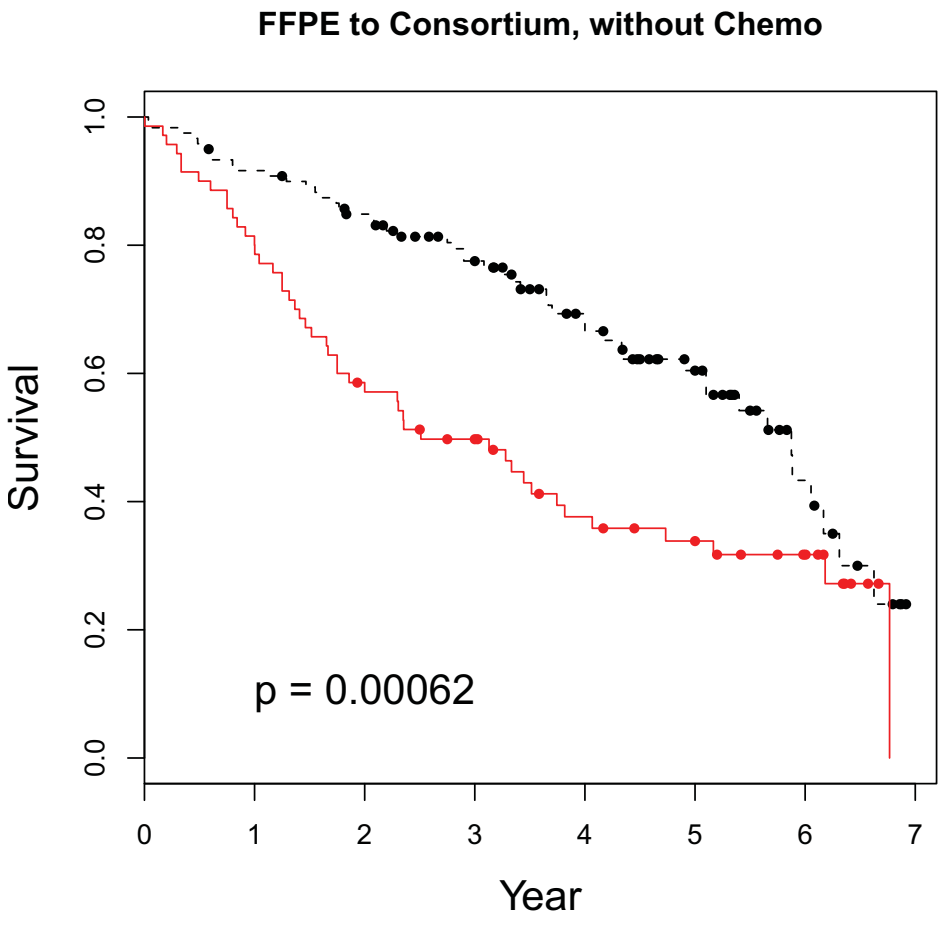


Figure 19: Kaplan Meier plots of survival, with groups predicted from FFPE to Consortium, without Chemo

9 FFPE data predict Consortium without truncation

Kaplan-Meier plots showing the predictive power of the robust gene signature developed from training set (55 FFPE tumor samples) and independently tested from different sets:

```
> display.surv(subset = which((!is.na(beer.clin$year))), main = "FFPE to Consortium w/o truncation",
+     tx = 4)

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1     269   269     269    120   6.47    5.88    8.01
fit$v.pred.1df=2     171   171     171    116   3.52    2.51    5.19
```

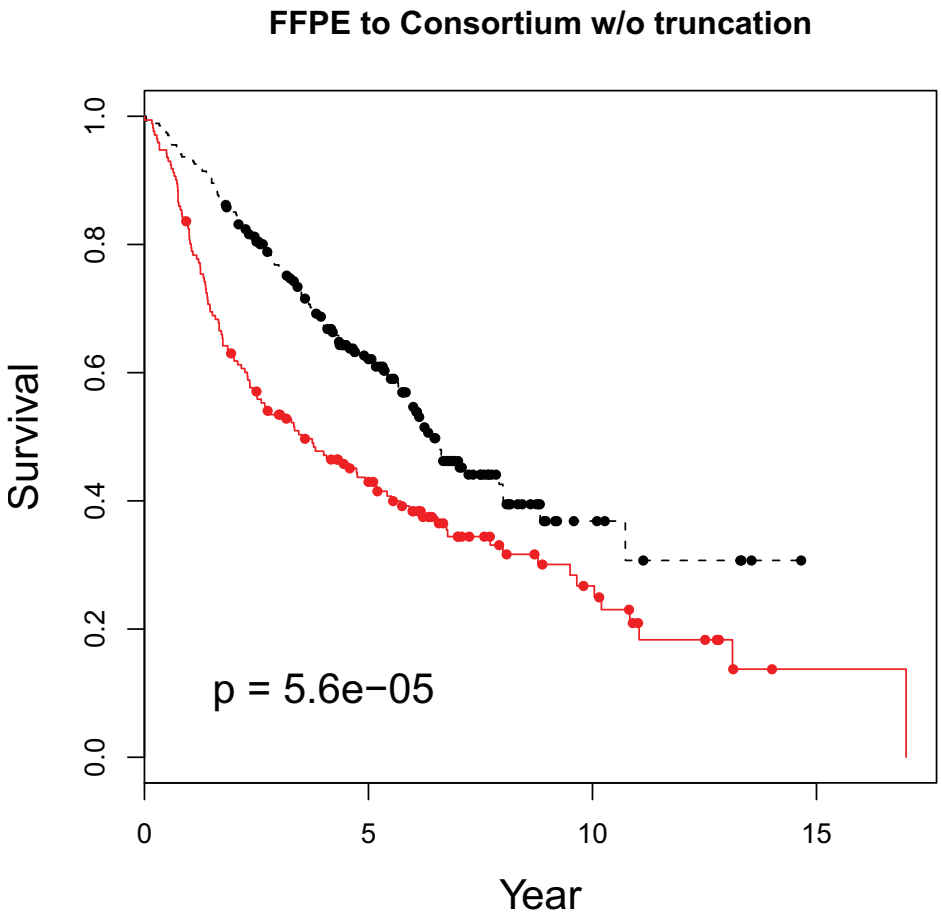


Figure 20: Kaplan Meier plots of survival, with groups predicted from FFPE to Consortium w/o truncation

```
> display.surv(subset = which((!is.na(beer.clin$year)) & beer.clin$dat.SITE ==
+     "MSKCC"), main = "FFPE to MSKCC  w/o truncation", tx = 3)

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      72    72      72     20     NA    5.67      NA
fit$v.pred.1df=2      32    32      32     19   4.75    2.00      NA
```

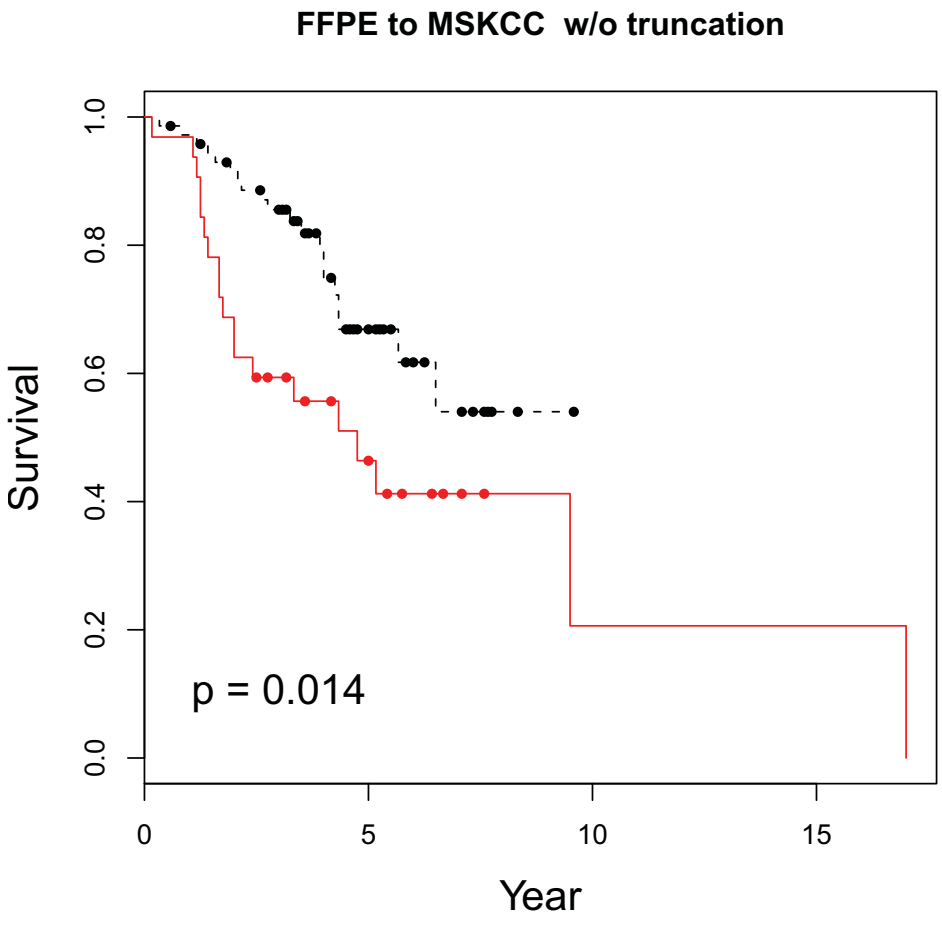


Figure 21: Kaplan Meier plots of survival, with groups predicted from FFPE to MSKCC

```
> display.surv(subset = which((!is.na(beer.clin$year)) & beer.clin$dat.SITE ==
+     "DFCI"), main = "FFPE to DFCI  w/o truncation")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      63    63      63     24   7.92   5.917      NA
fit$v.pred.1df=2      19    19      19     11   5.42   0.833      NA
```

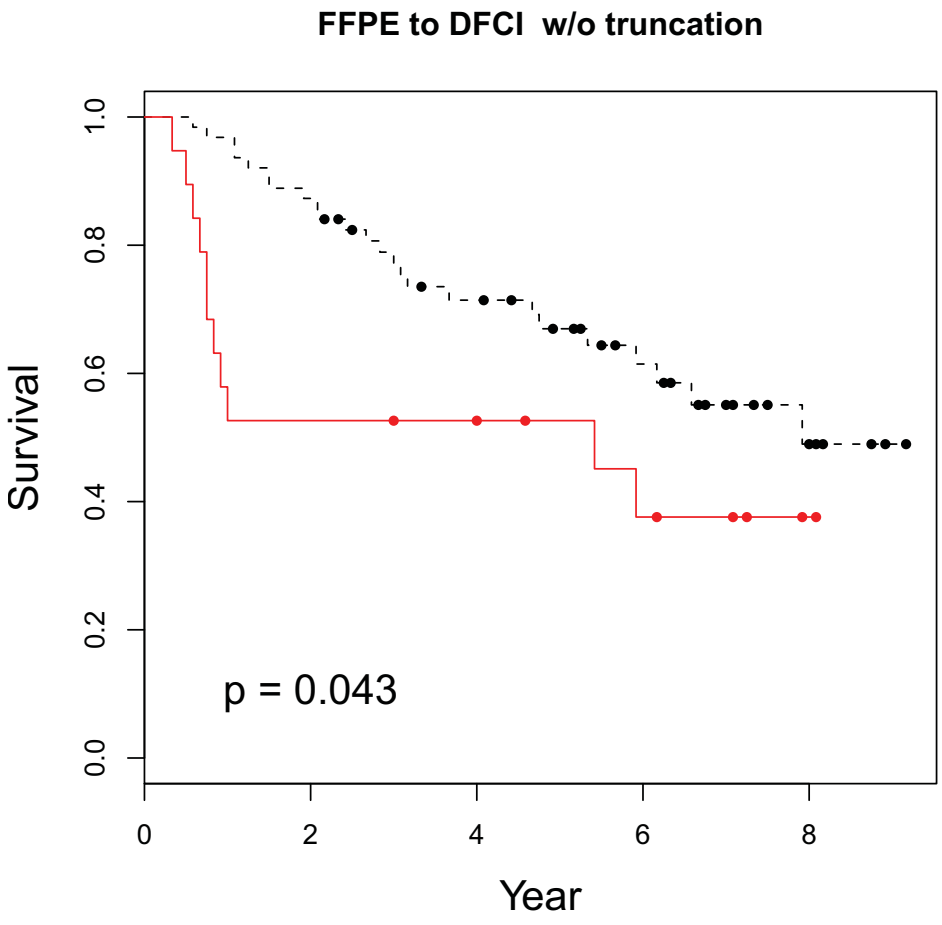


Figure 22: Kaplan Meier plots of survival, with groups predicted from FFPE to DFCI

```
> display.surv(subset = which((!is.na(beer.clin$year)) & beer.clin$dat.SITE ==
+     "MI"), main = "FFPE to MI  w/o truncation", tx = 3)

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      89    89      89     41   6.62    5.66      NA
fit$v.pred.1df=2      86    86      86     61   4.00    2.52    6.77
```

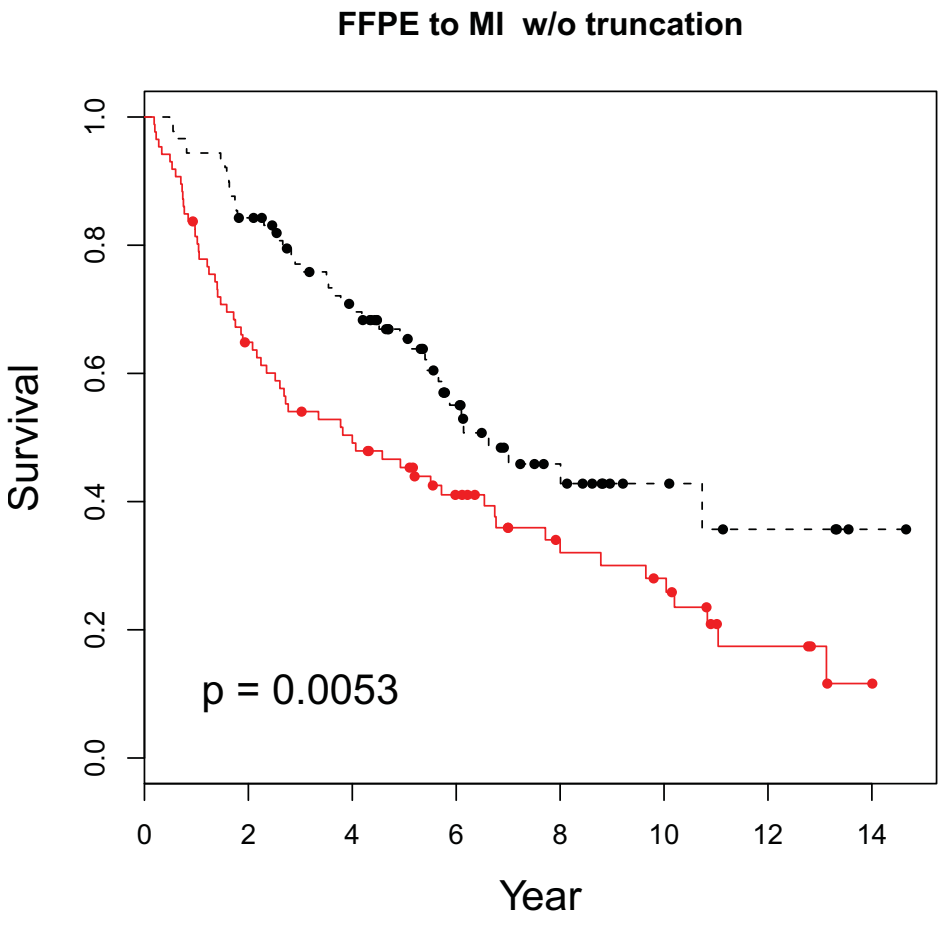


Figure 23: Kaplan Meier plots of survival, with groups predicted from FFPE to MI

```
> display.surv(subset = which((!is.na(beer.clin$year)) & beer.clin$dat.SITE ==
+     "HLM"), main = "FFPE to HLM  w/o truncation")

Call: survfit(formula = Surv(beer.clin$year, beer.clin$death) ~ fit$v.pred.1df,
    subset = subset)

                 records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      45    45      45     35   3.70    2.47    6.31
fit$v.pred.1df=2      34    34      34     25   2.43    1.65    4.73
```

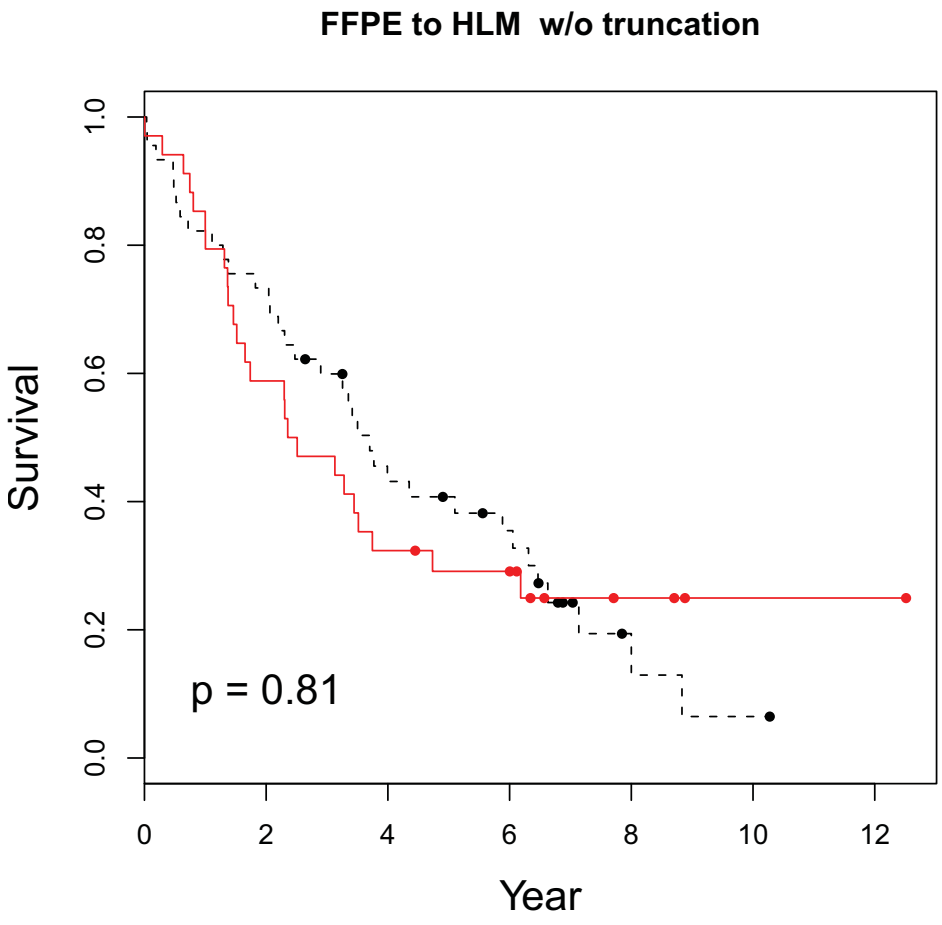


Figure 24: Kaplan Meier plots of survival, with groups predicted from FFPE to HLM

10 Survival analysis for individual genes

In this section, we did the survival analysis for individual genes, and compared the HR between FFPE and Consortium data, and each individual site of the Consortium.

```
> coxph.beer <- function(x, subset) {
+     summary(coxph(Surv(beer.clin$year, beer.clin$death) ~ unlist(x),
+         subset = subset))$coef
+ }
> coxph.ffpe <- function(x) {
+     summary(coxph(Surv(Death_Time, Death_Event) ~ unlist(x), data = clin))$coef
+ }
> beer.coef <- t(apply(beer.expr, 1, coxph.beer, subset = which(beer.clin$year <
+     7)))
> hist(beer.coef[, 5], nclass = 50)
> ffpe.coef <- t(apply(ffpe.expr, 1, coxph.ffpe))
> hist(ffpe.coef[, 5], nclass = 50)
> length(intersect(which(beer.coef[, 5] < 0.05), which(ffpe.coef[, 5] <
+     0.05)))

[1] 59

> length(setdiff(which(beer.coef[, 5] < 0.05), which(ffpe.coef[, 5] <
+     0.05)))

[1] 307

> length(setdiff(which(ffpe.coef[, 5] < 0.05), which(beer.coef[, 5] <
+     0.05)))

[1] 62


> gene.info <- read.csv("gene_info.csv")
> head(gene.info)

          Affy.ID Accession Symbol
1  AFFX-BioB-5_at    J04423
2  AFFX-BioB-M_at    U00096
3  AFFX-BioB-3_at    U00096
4  AFFX-BioC-5_at    U00096
5  AFFX-BioC-3_at    J04423
6 AFFX-BioDn-5_at    U00096

> out <- data.frame(ffpe.coef[, c(1, 2, 5)], beer.coef[, c(1, 2, 5)])
> colnames(out) <- c("ffpe.coef", "ffpe.HR", "ffpe.pv", "beer.coef",
+     "beer.HR", "beer.pv")
> head(out)
```

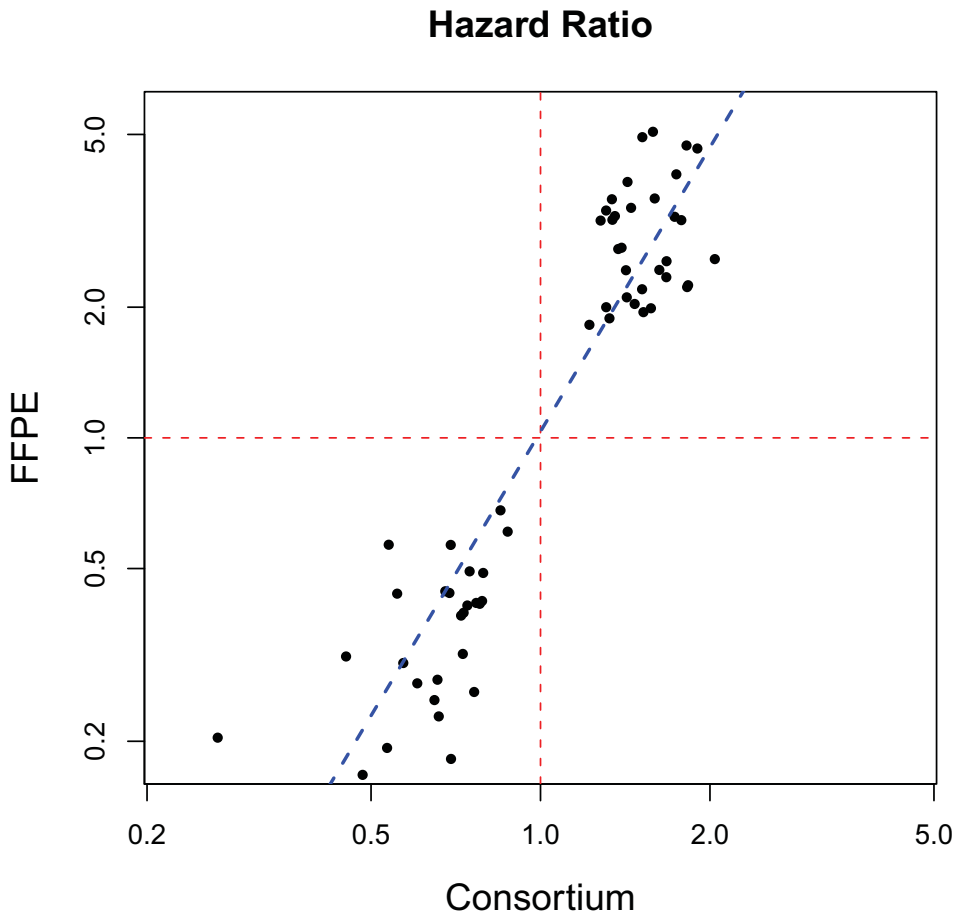


Figure 25: HR comparison between FFPE and Consortium

```
              ffpe.coef   ffpe.HR   ffpe.pv    beer.coef   beer.HR    beer.pv
1007_s_at   -0.43073807 0.6500291 0.2303817 -0.112608926 0.8935000 0.38405896
1316_at     -0.70127004 0.4959550 0.1029630 -0.031518456 0.9689731 0.78846129
200004_at   -0.04396638 0.9569861 0.8758638 -0.498065271 0.6077053 0.03048689
200007_at    0.32310218 1.3814065 0.5378768 -0.154367380 0.8569571 0.41758612
200012_x_at  0.27403627 1.3152625 0.5534911  0.255823805 1.2915251 0.19895556
200015_s_at -0.81764306 0.4414710 0.1854671 -0.003755492 0.9962516 0.98580058
```

```
> out <- merge(gene.info, out, by.x = "Affy.ID", by.y = "row.names")
> write.csv(out[out$ffpe.pv < 0.05 & out$beer.pv < 0.05, ], "59 genes.csv",
+     row.names = F)
```

11 Conclusions

12 Appendix

This computation was performed in the following environment:

```
> sessionInfo()

R version 2.10.0 (2009-10-26)
i386-pc-mingw32

locale:
[1] LC_COLLATE=English_United States.1252  LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
[5] LC_TIME=English_United States.1252

attached base packages:
[1] splines   stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] preprocessCore_1.8.0 affy_1.24.2          Biobase_2.6.1        superpc_1.06
[5] survival_2.35-8

loaded via a namespace (and not attached):
[1] affyio_1.14.0 tools_2.10.0
```

Predict Prognosis Using Refined Gene Signature

1 Introduction

In this particular report, we apply the 59-gene signature developed from FFPE and Consortium datasets to prediction the lung cancer prognosis in Bild et al (Nature 2006) and Bhattacharjee et al (PNAS 2001) datasets.

The Bild et al (Nature 2006) dataset was download from GEO website.

Both gene expression and clinical information of Bhattacharjee et al (PNAS 2001) dataset was download from the supplimentary matrials of the Bhattacharjee et al (2001) from PNAS website.

2 Getting Ready

We use these libraries:

```
> library(survival)
> library(superpc)
> library(affy)
> library(preprocessCore)
> pv.expr <- function(x, digits = 1) {
+     if (!x)
+         return(0)
+     exponent <- floor(log10(x))
+     base <- round(x/10^exponent, digits)
+     ifelse(x > 1e-06, paste("p = ", base * (10^exponent), sep = ""),
+         paste("p = ", base, "E", exponent, sep = ""))
+ }
> par(mar = c(4, 4, 3, 1), mfrow = c(1, 1))
```

3 Predict lung cancer patients' survival in Bhattacharjee et al dataset

In this section, we use the 59-gene signature developed from FFPE and Consortium datasets to prediction the lung cancer prognosis in Bhattacharjee et al (PNAS 2001) dataset.

Read FFPE data.

```
> ffpe.clin <- read.csv("FFPE.clin.csv", row.names = 1)
> head(ffpe.clin)
```

```
             Specimen.Number SPOR.N      Histology Final.Pat.Stage Time_to_Progression
AGR08.564.CEL            MDA5   1724       Squamous             IIA           2.3846680
AGR08.567.CEL            MDA9   1663 Adenocarcinoma              IB           2.2642026
AGR08.568.CEL           MDA10   1623       Squamous            IIIB           0.3778234
AGR08.571.CEL           MDA14   1576 Adenocarcinoma             IIB           2.8199863
AGR08.574.CEL           MDA18   1547 Adenocarcinoma             IIB           2.8966461
AGR08.576.CEL           MDA20   1537 Adenocarcinoma              IB           2.1300479
             Progression Death_Time Death_Event stage
AGR08.564.CEL           0   2.384668           0     2
AGR08.567.CEL           0   2.264203           0     1
AGR08.568.CEL           1   1.404517           1     3
AGR08.571.CEL           0   2.819986           0     2
AGR08.574.CEL           1   2.896646           0     2
AGR08.576.CEL           0   2.130048           0     1
```

```
> dim(ffpe.clin)
```

```
[1] 55  9
```

```
> clin <- ffpe.clin
> ffpe.fitted <- read.csv("selected_data_norm_b0.csv", row.names = 1)
> dim(ffpe.fitted)
```

```
[1] 1400   55
```

```
> expr <- ffpe.fitted[, rownames(clin)]
> dim(expr)
```

```
[1] 1400   55
```

```
> all(rownames(clin) == colnames(expr))
```

```
[1] TRUE
```

Read Bhattacharjee data. The expression data has been reduced to only the 59 probes that are corresponding the 59-gene signature.

```
> pnas.clin.dat <- read.csv("clin.Bhattacharjee.csv", row.names = 1)
> pnas.expr <- read.csv("expr.Bhattacharjee.csv", row.names = 1)
> head(pnas.clin.dat)
```

```
     Survival Sex Smoking Summary.Stage     Censor stage dead
A7       14.1   M     2.5          <NA> Uncensored    NA    1
A8       72.4   F    40.0            IA   Censored     1    0
A9       66.2   M    40.0            IA   Censored     1    0
A12      21.9   F    75.0           IIB Uncensored     2    1
A13      49.6   M    25.0            IA Uncensored     1    1
A195     76.6   f    10.0            IB   Censored     1    0
```

```
> dim(pnas.clin.dat)
```

```
[1] 203   7
```

```
> dim(pnas.expr)
```

```
[1]  45 203
```

```
> pnas.clin <- pnas.clin.dat[colnames(pnas.expr), ]
> pnas.clin$year <- pnas.clin$Survival/12
> which(rownames(pnas.clin) != colnames(pnas.expr))
```

```
integer(0)
```

Align and normalize FFPE and Bhattacharjee expression datasets.

```
> g59 <- read.csv("59 genes.csv", row.names = 1)
> head(g59, 2)
```

```
            Accession Symbol ffpe.coef   ffpe.HR    ffpe.pv  beer.coef   beer.HR
200080_s_at NM_002107  H3F3A -1.194011 0.3030034 0.03408396 -0.5612282 0.5705079
200775_s_at NM_031263 HNRNPK  1.172992 3.2316484 0.01769802  0.5485713 1.7307785
               beer.pv
200080_s_at 0.01900127
200775_s_at 0.01928427
```

```
> ffpe.expr <- expr[rownames(g59), ]
> dim(ffpe.expr)
```

```
[1] 59 55
```

```
> rownames(ffpe.expr) <- g59$Symbol
> cb.expr <- data.frame(ffpe.expr[rownames(pnas.expr), ], pnas.expr)
> dim(cb.expr)
```

```
[1]  45 258
```

```
> site <- c(rep("FFPE", dim(ffpe.expr)[2]), rep("pnas", dim(pnas.expr)[2]))
> cb.expr[] <- normalize.quantiles(as.matrix(cb.expr))
> ffpe.expr <- cb.expr[, site == "FFPE"]
> pnas.expr <- cb.expr[, site != "FFPE"]
```

Build prodiction model from FFPE data and predict the lung cancer patients' survival in Bhattacharjee et al dataset.

```
> data.train <- list(x = ffpe.expr, y = clin$Death_Time, censoring.status = clin$Death_Event,
+     featurenames = rownames(ffpe.expr))
> train.obj <- superpc.train(data.train, type = "survival")
> head(pnas.clin)
```

```
     Survival Sex Smoking Summary.Stage     Censor stage dead     year
A7       14.1   M     2.5          <NA> Uncensored    NA    1 1.175000
A8       72.4   F    40.0            IA   Censored     1    0 6.033333
A9       66.2   M    40.0            IA   Censored     1    0 5.516667
A12      21.9   F    75.0           IIB Uncensored     2    1 1.825000
A13      49.6   M    25.0            IA Uncensored     1    1 4.133333
A195     76.6   f    10.0            IB   Censored     1    0 6.383333
```

```
> data.test <- list(x = pnas.expr, y = pnas.clin$year, censoring.status = pnas.clin$dead,
+     featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> table(fit$v.pred.1df)

  1   2
138  65

> risk <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "continuous")$v.pred.1df
> subset <- which(pnas.clin$year < 7)
> summary(coxph(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset))

Call:
coxph(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

  n= 117

                 coef exp(coef) se(coef)     z Pr(>|z|)
fit$v.pred.1df 0.5932    1.8097   0.2499 2.373   0.0176 *
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

               exp(coef) exp(-coef) lower .95 upper .95
fit$v.pred.1df     1.810     0.5526     1.109     2.954

Rsquare= 0.044   (max possible= 0.993 )
Likelihood ratio test= 5.31  on 1 df,   p=0.02122
Wald test            = 5.63  on 1 df,   p=0.01763
Score (logrank) test = 5.8  on 1 df,   p=0.01607

> surv.fit <- survfit(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)

Call: survfit(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      82    82      82     43   4.07    2.93      NA
fit$v.pred.1df=2      35    35      35     26   2.02    1.27    3.93

> logrank <- survdiff(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> logrank
```

```
Call:
survdiff(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 82       43     51.6      1.44      5.78
fit$v.pred.1df=2 35       26     17.4      4.28      5.78

 Chisq= 5.8  on 1 degrees of freedom, p= 0.0162
```

```
> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
> pnas.clin$gender <- NA
> pnas.clin$gender[pnas.clin$Sex == "M"] <- "M"
> pnas.clin$gender[pnas.clin$Sex == "m"] <- "M"
> pnas.clin$gender[pnas.clin$Sex == "F"] <- "F"
> pnas.clin$gender[pnas.clin$Sex == "f"] <- "F"
> table(pnas.clin$gender)
```

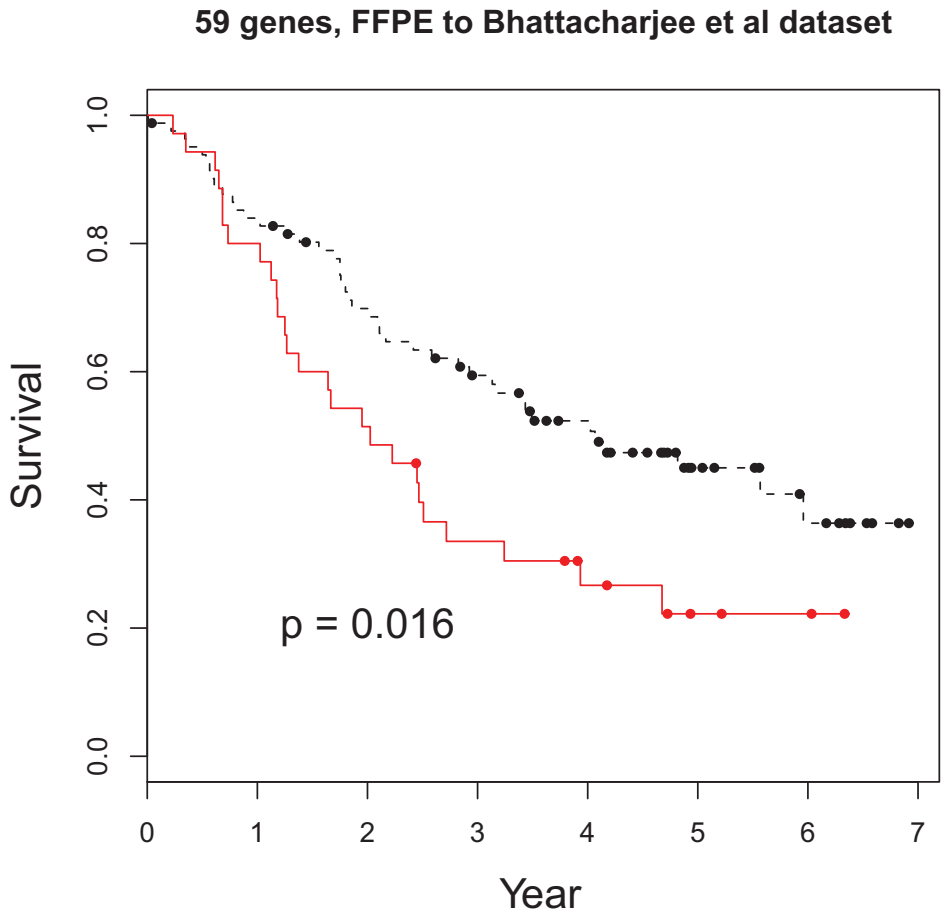
```
 F  M
72 53
```

```
> summary(coxph(Surv(pnas.clin$year, pnas.clin$dead) ~ risk + gender +
+     Smoking, data = pnas.clin, subset = subset))
```

```
Call:
coxph(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ risk +
    gender + Smoking, data = pnas.clin, subset = subset)

  n=115 (2 observations deleted due to missingness)

              coef  exp(coef)   se(coef)      z Pr(>|z|)
risk     0.3294649  1.3902240  0.1289459  2.555   0.0106 *
genderM  0.0795014  1.0827471  0.2494269  0.319   0.7499
Smoking -0.0001487  0.9998513  0.0037220 -0.040   0.9681
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

        exp(coef) exp(-coef) lower .95 upper .95
risk       1.3902     0.7193    1.0798     1.790
genderM    1.0827     0.9236    0.6641     1.765
Smoking    0.9999     1.0001    0.9926     1.007

Rsquare= 0.056   (max possible= 0.993 )
Likelihood ratio test= 6.62  on 3 df,   p=0.08487
Wald test            = 6.55  on 3 df,   p=0.08764
Score (logrank) test = 6.62  on 3 df,   p=0.0852
```

```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bhattacharjee et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```

```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bhattacharjee et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```



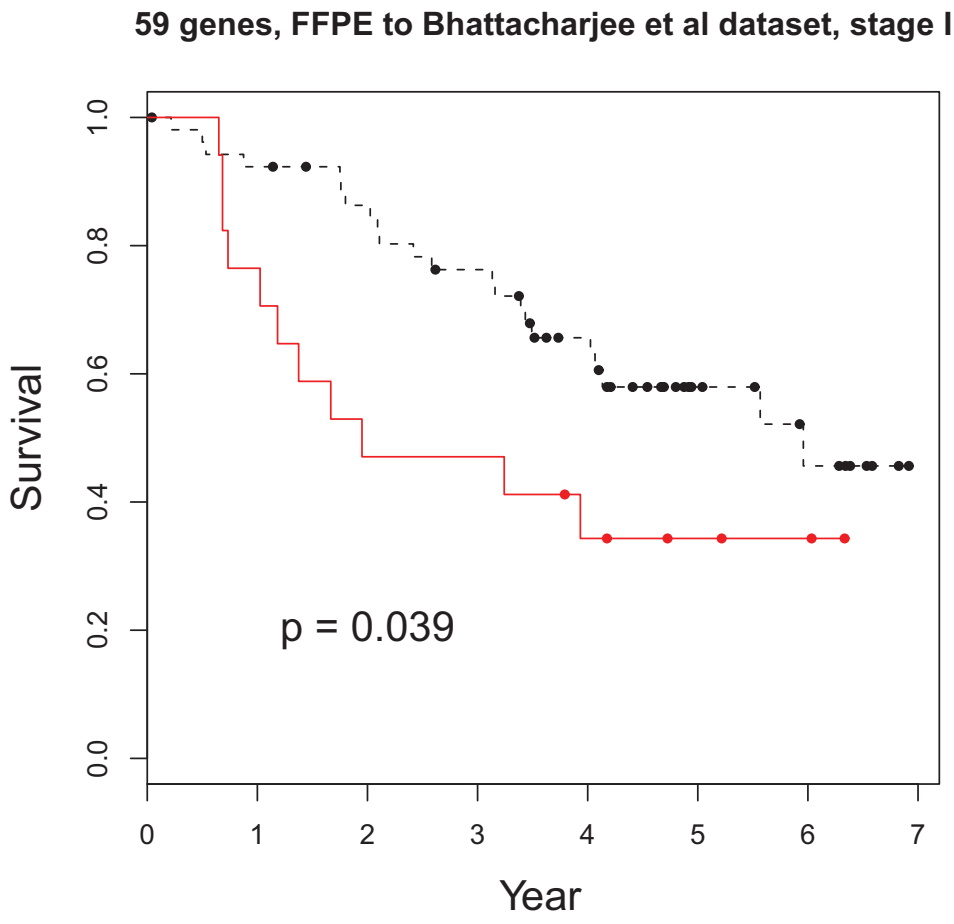
Kaplan Meier plots of survival for high and low risk group in Bhattacharjee et al dataset predicted from 59-gene signature

Predict the lung cancer stage I patients' survival in Bhattacharjee et al dataset.

```
> subset <- which(pnas.clin$year < 7 & pnas.clin$stage == 1)
> surv.fit <- survfit(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)

Call: survfit(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
```

```
fit$v.pred.1df=1      53    53      53     22   5.96    4.07      NA
fit$v.pred.1df=2      17    17      17     11   1.95    1.18      NA
```

```
> logrank <- survdiff(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> logrank

Call:
survdiff(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                  N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 53       22    26.65     0.812      4.25
fit$v.pred.1df=2 17       11     6.35     3.409      4.25

 Chisq= 4.3  on 1 degrees of freedom, p= 0.0392

> pv <- pchisq(logrank$chisq, 1, lower.tail = F)

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bhattacharjee et al dataset, stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bhattacharjee et al dataset, stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```



Kaplan Meier plots of survival for high and low risk group of stage I patients in Bhattacharjee et al dataset predicted from 59-gene signature

```
> subset <- which(pnas.clin$year < 7 & pnas.clin$stage == 1)
> summary(coxph(Surv(pnas.clin$year, pnas.clin$dead) ~ risk + gender +
+     Smoking, data = pnas.clin, subset = subset))

Call:
coxph(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ risk +
    gender + Smoking, data = pnas.clin, subset = subset)

  n= 70

              coef  exp(coef)   se(coef)      z Pr(>|z|)
risk     0.4410060  1.5542701  0.1991547  2.214   0.0268 *
genderM  0.2638568  1.3019417  0.3582436  0.737   0.4614
Smoking -0.0005453  0.9994548  0.0056967 -0.096   0.9237
```

```
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

        exp(coef) exp(-coef) lower .95 upper .95
risk       1.5543     0.6434    1.0520     2.296
genderM    1.3019     0.7681    0.6451     2.627
Smoking    0.9995     1.0005    0.9884     1.011

Rsquare= 0.073   (max possible= 0.973 )
Likelihood ratio test= 5.28  on 3 df,   p=0.1524
Wald test            = 5.12  on 3 df,   p=0.1634
Score (logrank) test = 5.2  on 3 df,   p=0.1576
```

```
> pnas.clin.sub <- pnas.clin[!is.na(pnas.clin$year + pnas.clin$dead),
+     ]
> table(pnas.clin.sub$stage)

 1  2  3  4
76 24 10  3

> table(pnas.clin$stage)

 1  2  3  4
76 24 10  3
```

Build prodiction model from Consortium data and predict the lung cancer patients' survival in Bhattacharjee et al dataset.

```
> pnas.clin <- read.csv("clin.Bhattacharjee.csv", row.names = 1)
> pnas.expr <- read.csv("expr.Bhattacharjee.csv", row.names = 1)
> dim(pnas.expr)

[1]  45 203

> pnas.clin$year <- pnas.clin$Survival/12
> table(pnas.clin$dead, pnas.clin$Censor)

       censored Censored NA- uncensored Uncensored
  0  0       42       12   0          0          0
  1  0        0        0   0         51         20

> pnas.expr <- pnas.expr[, rownames(pnas.clin)]
> which(rownames(pnas.clin) != colnames(pnas.expr))

integer(0)

> beer.clin.dat <- read.csv("Consortium.clin.csv", row.names = 1)
> beer.expr <- read.csv("Consortium.expr.csv", row.names = 1)
> beer.clin <- beer.clin.dat[colnames(beer.expr), ]
> which(rownames(beer.clin) != colnames(beer.expr))
```

```
integer(0)

> dim(beer.clin)

[1] 442  10

> dim(beer.expr)

[1] 1012  442

> g59 <- read.csv("59 genes.csv", row.names = 1)
> head(g59, 2)

            Accession Symbol ffpe.coef   ffpe.HR    ffpe.pv  beer.coef   beer.HR
200080_s_at NM_002107  H3F3A -1.194011 0.3030034 0.03408396 -0.5612282 0.5705079
200775_s_at NM_031263 HNRNPK  1.172992 3.2316484 0.01769802  0.5485713 1.7307785
               beer.pv
200080_s_at 0.01900127
200775_s_at 0.01928427

> beer.expr <- beer.expr[rownames(g59), ]
> dim(beer.expr)

[1]  59 442

> rownames(beer.expr) <- g59$Symbol
> cb.expr <- data.frame(beer.expr[rownames(pnas.expr), ], pnas.expr)
> dim(cb.expr)

[1]  45 645

> site <- c(rep("beer", dim(beer.expr)[2]), rep("pnas", dim(pnas.expr)[2]))
> cb.expr[] <- normalize.quantiles(as.matrix(cb.expr))
> beer.expr <- cb.expr[, site == "beer"]
> pnas.expr <- cb.expr[, site != "beer"]

> ind1 <- which(!is.na(beer.clin$year))
> length(ind1)

[1] 440

> data.train <- list(x = beer.expr[, ind1], y = beer.clin$year[ind1],
+     censoring.status = beer.clin$death[ind1], featurenames = NULL)
> train.obj <- superpc.train(data.train, type = "survival")
> head(pnas.clin)

     Survival Sex Smoking Summary.Stage     Censor stage dead     year
A7       14.1   M     2.5          <NA> Uncensored    NA    1 1.175000
A8       72.4   F    40.0            IA   Censored     1    0 6.033333
A9       66.2   M    40.0            IA   Censored     1    0 5.516667
A12      21.9   F    75.0           IIB Uncensored     2    1 1.825000
A13      49.6   M    25.0            IA Uncensored     1    1 4.133333
A195     76.6   f    10.0            IB   Censored     1    0 6.383333
```

```
> data.test <- list(x = pnas.expr, y = pnas.clin$year, censoring.status = pnas.clin$dead,
+     featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> table(fit$v.pred.1df)

  1   2
115  88

> risk <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "continuous")$v.pred.1df
> pnas.clin$gender <- NA
> pnas.clin$gender[pnas.clin$Sex == "M"] <- "M"
> pnas.clin$gender[pnas.clin$Sex == "m"] <- "M"
> pnas.clin$gender[pnas.clin$Sex == "F"] <- "F"
> pnas.clin$gender[pnas.clin$Sex == "f"] <- "F"
> table(pnas.clin$gender)

 F  M
72 53

> subset <- which(pnas.clin$year < 7)
> length(which(!is.na(pnas.clin$year + pnas.clin$dead + pnas.clin$stage)))

[1] 113

> length(which(pnas.clin$year < 7))

[1] 117

> summary(coxph(Surv(pnas.clin$year, pnas.clin$dead) ~ risk + gender +
+     Smoking, data = pnas.clin, subset = subset))

Call:
coxph(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ risk +
    gender + Smoking, data = pnas.clin, subset = subset)

  n=115 (2 observations deleted due to missingness)

              coef  exp(coef)   se(coef)      z Pr(>|z|)
risk     1.0040959  2.7294384  0.3925910  2.558   0.0105 *
genderM  0.1048533  1.1105477  0.2518452  0.416   0.6772
Smoking -0.0004529  0.9995472  0.0037032 -0.122   0.9027
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

        exp(coef) exp(-coef) lower .95 upper .95
risk       2.7294     0.3664    1.2644     5.892
genderM    1.1105     0.9005    0.6779     1.819
Smoking    0.9995     1.0005    0.9923     1.007
```

```
Rsquare= 0.054   (max possible= 0.993 )
Likelihood ratio test= 6.39  on 3 df,   p=0.09423
Wald test            = 6.59  on 3 df,   p=0.08615
Score (logrank) test = 6.65  on 3 df,   p=0.08397
```

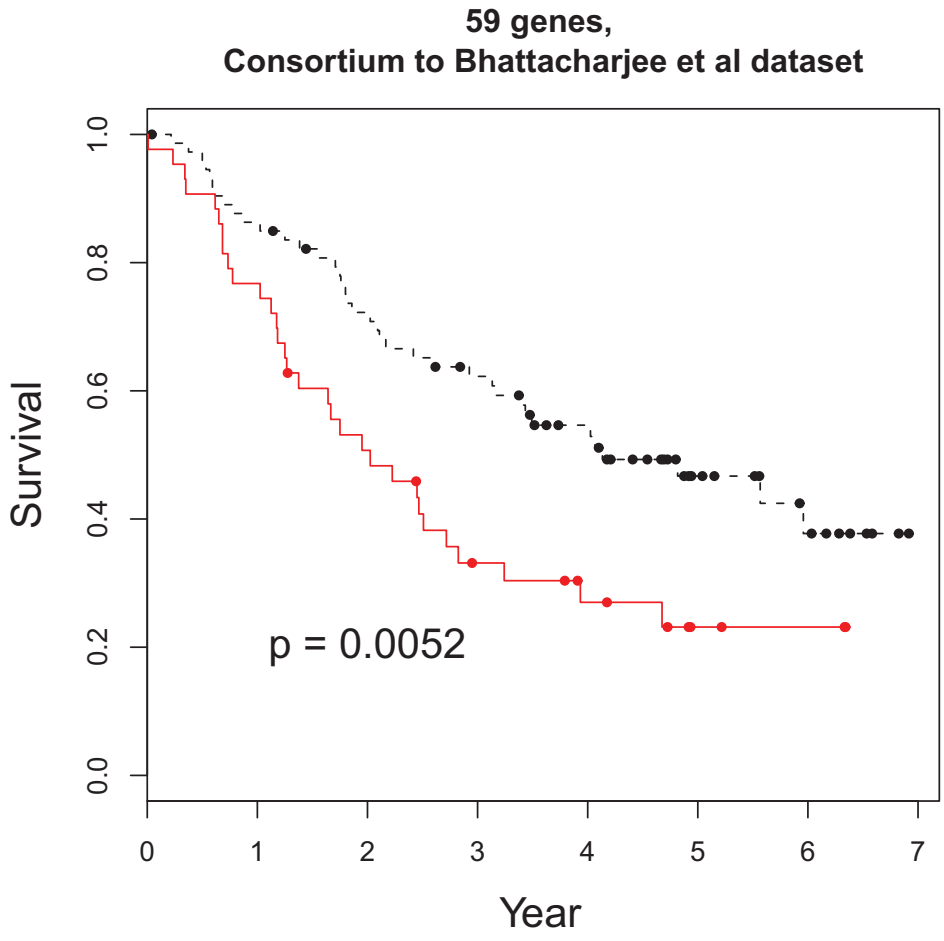
```
> surv.fit <- survfit(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)
```

```
Call: survfit(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      74    74      74     38   4.13    3.16      NA
fit$v.pred.1df=2      43    43      43     31   2.02    1.27    3.24
```

```
> logrank <- survdiff(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
```

```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bhattacharjee et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```

```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bhattacharjee et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```

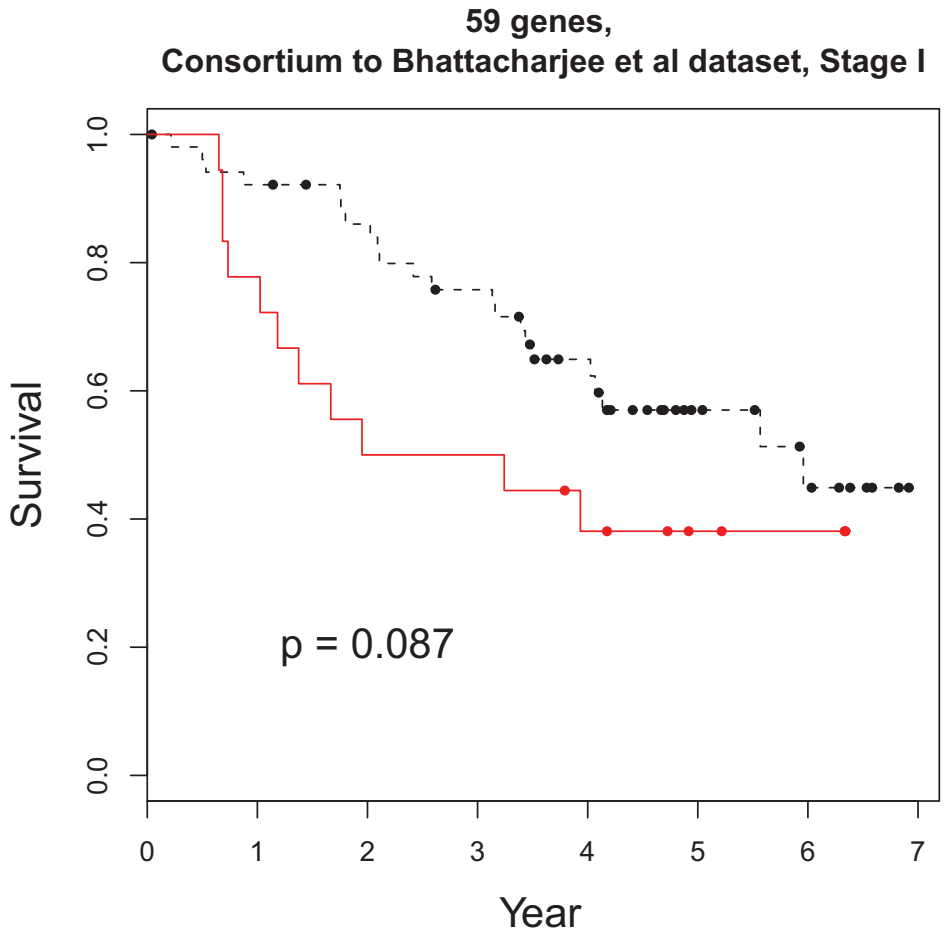


Kaplan Meier plots of survival for high and low risk group in Bhattacharjee et al dataset predicted from 59-gene signature

Now predict the survival time for stage I patients only.

```
> subset <- which(pnas.clin$year < 7 & pnas.clin$stage == 1)
> summary(coxph(Surv(pnas.clin$year, pnas.clin$dead) ~ risk + gender +
+     Smoking, data = pnas.clin, subset = subset))

Call:
coxph(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ risk +
    gender + Smoking, data = pnas.clin, subset = subset)

  n= 70

             coef exp(coef)  se(coef)      z Pr(>|z|)
risk     1.233445  3.433036  0.633187  1.948   0.0514 .
```

```
genderM  0.275303  1.316929  0.362705  0.759   0.4478
Smoking -0.001319  0.998681  0.005815 -0.227   0.8205
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

        exp(coef) exp(-coef) lower .95 upper .95
risk       3.4330     0.2913    0.9925    11.875
genderM    1.3169     0.7593    0.6469     2.681
Smoking    0.9987     1.0013    0.9874     1.010

Rsquare= 0.055   (max possible= 0.973 )
Likelihood ratio test= 3.98  on 3 df,   p=0.2632
Wald test            = 4.07  on 3 df,   p=0.2536
Score (logrank) test = 4.1  on 3 df,   p=0.2505

> surv.fit <- survfit(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)

Call: survfit(formula = Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      52    52      52     22   5.96    4.02      NA
fit$v.pred.1df=2      18    18      18     11   2.60    1.18      NA

> logrank <- survdiff(Surv(pnas.clin$year, pnas.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> pv <- pchisq(logrank$chisq, 1, lower.tail = F)

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bhattacharjee et al dataset, Stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bhattacharjee et al dataset, Stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```



Kaplan Meier plots of survival for high and low risk group of stage I patients in Bhattacharjee et al dataset predicted from 59-gene signature

4 Predict lung cancer patients survival in Bild et al dataset

In this section, we use the 59-gene signature developed from FFPE and Consortium datasets to prediction the lung cancer prognosis in Bild et al (Nature 2006) dataset.

Read FFPE data.

```
> ffpe.clin <- read.csv("FFPE.clin.csv", row.names = 1)
> head(ffpe.clin)
```

```
             Specimen.Number SPOR.N      Histology Final.Pat.Stage Time_to_Progression
AGR08.564.CEL            MDA5   1724       Squamous             IIA           2.3846680
AGR08.567.CEL            MDA9   1663 Adenocarcinoma              IB           2.2642026
AGR08.568.CEL           MDA10   1623       Squamous            IIIB           0.3778234
AGR08.571.CEL           MDA14   1576 Adenocarcinoma             IIB           2.8199863
```

```
AGR08.574.CEL           MDA18   1547 Adenocarcinoma             IIB           2.8966461
AGR08.576.CEL           MDA20   1537 Adenocarcinoma              IB           2.1300479
             Progression Death_Time Death_Event stage
AGR08.564.CEL           0   2.384668           0     2
AGR08.567.CEL           0   2.264203           0     1
AGR08.568.CEL           1   1.404517           1     3
AGR08.571.CEL           0   2.819986           0     2
AGR08.574.CEL           1   2.896646           0     2
AGR08.576.CEL           0   2.130048           0     1
```

```
> dim(ffpe.clin)
```

```
[1] 55  9
```

```
> clin <- ffpe.clin
> ffpe.fitted <- read.csv("selected_data_norm_b0.csv", row.names = 1)
> dim(ffpe.fitted)
```

```
[1] 1400   55
```

```
> expr <- ffpe.fitted[, rownames(clin)]
> dim(expr)
```

```
[1] 1400   55
```

```
> all(rownames(clin) == colnames(expr))
```

```
[1] TRUE
```

Read Bild data. The expression data has been reduced to only the 59 probes that are corresponding the 59-gene signature.

```
> duke.clin <- read.csv("Bild clin from database.csv", row.names = 1)
> duke.expr <- read.csv("./Bild et al/GSE3141_series_matrix.csv", row.names = 1)
> range(duke.expr)
```

```
[1]      0.1 302457.4
```

```
> gc()
```

```
         used (Mb) gc trigger  (Mb) max used  (Mb)
Ncells  430779 11.6     984024  26.3   667722  17.9
Vcells 6668965 50.9   22283078 170.1 25012849 190.9
```

```
> duke.expr[] <- log2(duke.expr)
> hist(as.matrix(duke.expr))
> range(duke.expr)
```

```
[1] -3.321928 18.206372
```

```
> head(duke.clin)
```

```
         Stage      Histology death month
GSM70202     I       Squamous     1  23.6
GSM70203    II Adenocarcinoma     0  41.7
GSM70219  <NA> Adenocarcinoma     1  39.0
GSM70204     I Adenocarcinoma     0  41.6
GSM70158    IV       Squamous     1   5.3
GSM70194    IA Adenocarcinoma     0  51.2
```

```
> dim(duke.clin)
```

```
[1] 111   4
```

```
> dim(duke.expr)
```

```
[1] 54675   111
```

```
> duke.clin <- duke.clin[colnames(duke.expr), ]
> duke.clin$year <- duke.clin$month/12
> duke.clin$dead <- duke.clin$death
> all(rownames(duke.clin) == colnames(duke.expr))
```

```
[1] TRUE
```

```
> duke.clin$stage <- NA
> table(duke.clin$Stage)
```

```
       I   IA   IB   II  IIA  IIB IIIA   IV
   2   7   29   26    1    4   11    4    1
```

```
> duke.clin$stage[duke.clin$Stage %in% c("I", "IA", "IB")] <- 1
> duke.clin$stage[duke.clin$Stage %in% c("II", "IIA", "IIB")] <- 2
> duke.clin$stage[duke.clin$Stage %in% c("IIIA")] <- 3
> duke.clin$stage[duke.clin$Stage %in% c("IV")] <- 4
> table(duke.clin$stage, duke.clin$Stage, useNA = "ifany")
```

```
          I IA IB II IIA IIB IIIA IV <NA>
  1     0  7 29 26  0   0   0    0  0    0
  2     0  0  0  0  1   4  11    0  0    0
  3     0  0  0  0  0   0   0    4  0    0
  4     0  0  0  0  0   0   0    0  1    0
  <NA>  2  0  0  0  0   0   0    0  0   26
```

Align and normalize FFPE and Bild expression datasets.

```
> g59 <- read.csv("59 genes.csv", row.names = 1)
> head(g59, 2)
```

```
            Accession Symbol ffpe.coef   ffpe.HR    ffpe.pv  beer.coef   beer.HR
200080_s_at NM_002107  H3F3A -1.194011 0.3030034 0.03408396 -0.5612282 0.5705079
200775_s_at NM_031263 HNRNPK  1.172992 3.2316484 0.01769802  0.5485713 1.7307785
               beer.pv
200080_s_at 0.01900127
200775_s_at 0.01928427
```

```
> ffpe.expr <- expr[rownames(g59), ]
> dim(ffpe.expr)

[1] 59 55

> range(ffpe.expr)

[1]  3.942112 14.821264

> head(rownames(duke.expr))

[1] "1007_s_at" "1053_at"   "117_at"    "121_at"    "1255_g_at" "1294_at"

> length(intersect(rownames(ffpe.expr), rownames(duke.expr)))

[1] 59

> duke.expr <- duke.expr[rownames(ffpe.expr), ]
> all(rownames(ffpe.expr) == rownames(duke.expr))

[1] TRUE

> cb.expr <- data.frame(ffpe.expr, duke.expr)
> dim(cb.expr)

[1]  59 166

> site <- c(rep("FFPE", dim(ffpe.expr)[2]), rep("duke", dim(duke.expr)[2]))
> cb.expr[] <- normalize.quantiles(as.matrix(cb.expr))
> ffpe.expr <- cb.expr[, site == "FFPE"]
> duke.expr <- cb.expr[, site != "FFPE"]
```

Build prodiction model from FFPE data and predict the lung cancer patients' survival in Bild et al dataset.

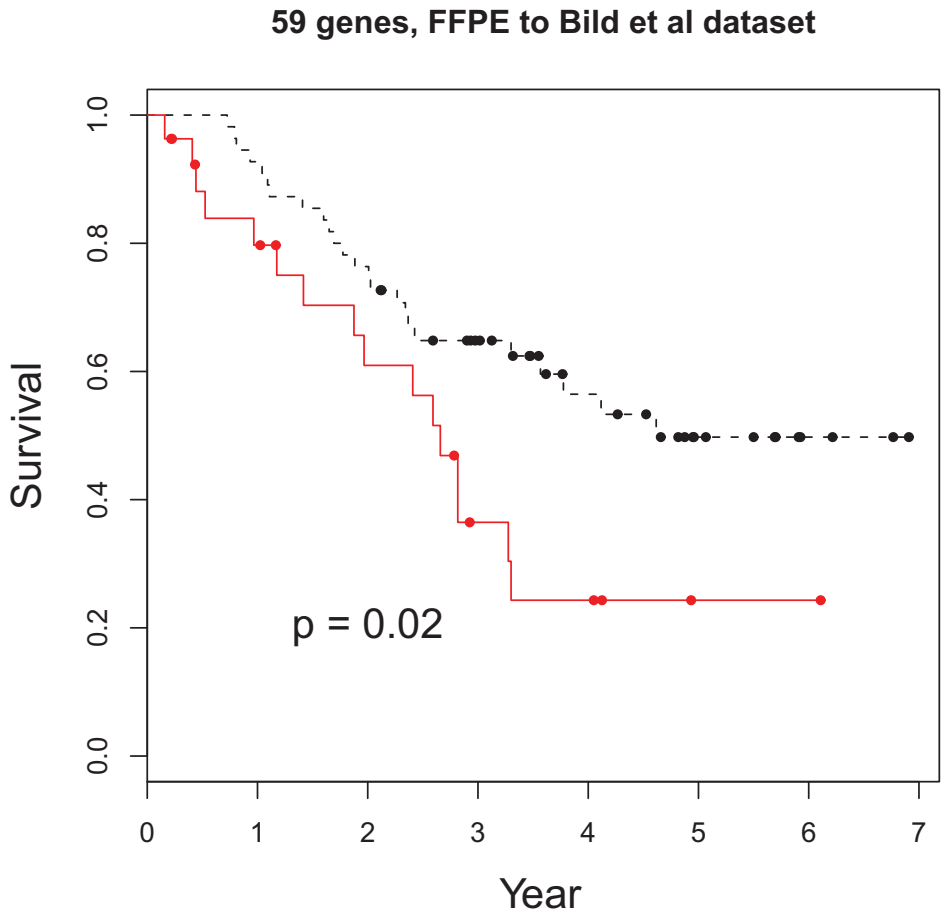
```
> data.train <- list(x = ffpe.expr, y = ffpe.clin$Death_Time, censoring.status = ffpe.clin$Death_Event,
+     featurenames = rownames(ffpe.expr))
> train.obj <- superpc.train(data.train, type = "survival")
> data.test <- list(x = duke.expr, y = duke.clin$year, censoring.status = duke.clin$dead,
+     featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> table(fit$v.pred.1df)

 1  2
75 36

> attributes(fit)

$names
[1] "v.pred"          "u"               "d"               "which.features"
[5] "v.pred.1df"      "n.components"    "coef"            "call"
[9] "prediction.type"
```

```
> fit$coef

   score.1    score.2    score.3
-2.4476833  0.7506039  0.4273745

> fit$d

[1] 23.55970 10.73353 10.23920

> um <- data.frame(fit$u)
> rownames(um) <- names(which(fit$which.features))
> write.csv(um, "U matrix new.csv")
> risk <- NULL
> risk <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "continuous")$v.pred.1df
> hist(unlist(risk))
> subset <- which(duke.clin$year < 7 & (!is.na(duke.clin$stage)))
> summary(coxph(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset))

Call:
coxph(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

  n= 82

                 coef exp(coef) se(coef)     z Pr(>|z|)
fit$v.pred.1df 0.7444    2.1051   0.3256 2.286   0.0222 *
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

               exp(coef) exp(-coef) lower .95 upper .95
fit$v.pred.1df     2.105      0.475     1.112     3.985

Rsquare= 0.058   (max possible= 0.979 )
Likelihood ratio test= 4.89  on 1 df,   p=0.02697
Wald test            = 5.23  on 1 df,   p=0.02223
Score (logrank) test = 5.47  on 1 df,   p=0.01939

> surv.fit <- survfit(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)

Call: survfit(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      55    55      55     24   4.62    3.57      NA
fit$v.pred.1df=2      27    27      27     16   2.66    1.88      NA
```

```
> logrank <- survdiff(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> logrank

Call:
survdiff(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 55       24    30.27      1.30      5.41
fit$v.pred.1df=2 27       16     9.73      4.04      5.41

 Chisq= 5.4  on 1 degrees of freedom, p= 0.02

> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
> cox.fit <- coxph(Surv(duke.clin$year, duke.clin$dead) ~ risk + duke.clin$Histology,
+     subset = subset)
> summary(cox.fit)

Call:
coxph(formula = Surv(duke.clin$year, duke.clin$dead) ~ risk +
    duke.clin$Histology, subset = subset)

  n= 82

                               coef exp(coef) se(coef)      z Pr(>|z|)
risk                         0.2909    1.3376   0.1293  2.249   0.0245 *
duke.clin$HistologySquamous -0.4171    0.6589   0.3583 -1.164   0.2443
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

                            exp(coef) exp(-coef) lower .95 upper .95
risk                            1.338     0.7476    1.0381     1.724
duke.clin$HistologySquamous     0.659     1.5176    0.3265     1.330

Rsquare= 0.06   (max possible= 0.979 )
Likelihood ratio test= 5.04  on 2 df,   p=0.08054
Wald test            = 5.08  on 2 df,   p=0.07902
Score (logrank) test = 5.04  on 2 df,   p=0.08028

> head(duke.clin)

         Stage      Histology death month      year dead stage
GSM70125    IA       Squamous     0  57.8 4.8166667    0     1
GSM70126    IA       Squamous     0  66.0 5.5000000    0     1
GSM70127  <NA> Adenocarcinoma     0  59.8 4.9833333    0    NA
GSM70128   IIB       Squamous     0  14.0 1.1666667    0     2
GSM70129  IIIA Adenocarcinoma     1  11.6 0.9666667    1     3
GSM70130    IA Adenocarcinoma     0  34.8 2.9000000    0     1
```

```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bild et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```


```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bild et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```

Kaplan Meier plots of survival for high and low risk group in Bild et al dataset predicted from 59-gene signature

Predict the lung cancer stage I patients' survival in Bild et al dataset.

```
> subset <- which(duke.clin$year < 7 & duke.clin$stage == 1)
> surv.fit <- survfit(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)
```

```
Call: survfit(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      40    40      40     15     NA    4.12      NA
fit$v.pred.1df=2      21    21      21     12   2.82    2.41      NA
```

```
> logrank <- survdiff(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> logrank
```

```
Call:
survdiff(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 40       15    19.96      1.23      4.85
fit$v.pred.1df=2 21       12     7.04      3.50      4.85

 Chisq= 4.9  on 1 degrees of freedom, p= 0.0276
```

```
> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
> table(duke.clin$stage)
```

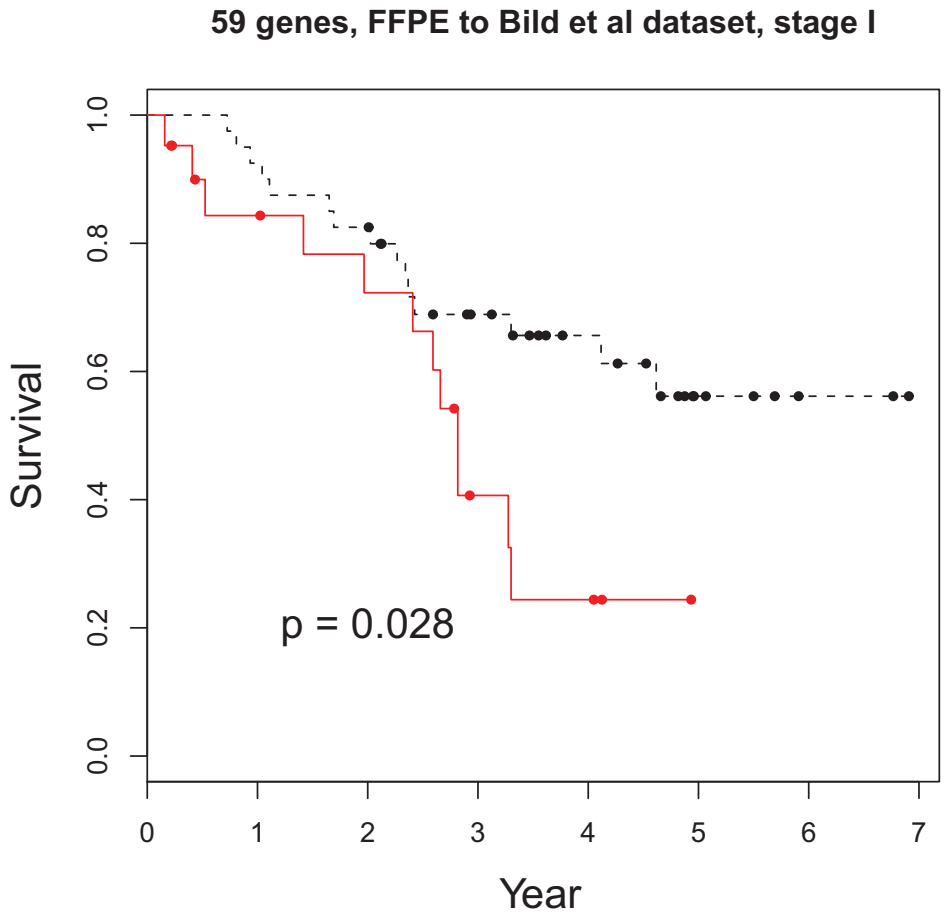
```
 1  2  3  4
62 16  4  1
```

```
> cox.fit <- coxph(Surv(duke.clin$year, duke.clin$dead) ~ risk + duke.clin$Histology,
+     subset = subset)
> summary(cox.fit)
```

```
Call:
coxph(formula = Surv(duke.clin$year, duke.clin$dead) ~ risk +
    duke.clin$Histology, subset = subset)

  n= 61

                               coef exp(coef) se(coef)      z Pr(>|z|)
risk                         0.3059    1.3578   0.1540  1.986    0.047 *
duke.clin$HistologySquamous -0.4209    0.6565   0.4554 -0.924    0.355
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

                            exp(coef) exp(-coef) lower .95 upper .95
risk                           1.3578     0.7365    1.0040     1.836
duke.clin$HistologySquamous    0.6565     1.5233    0.2689     1.603

Rsquare= 0.062   (max possible= 0.961 )
Likelihood ratio test= 3.93  on 2 df,   p=0.1405
Wald test            = 3.97  on 2 df,   p=0.1375
Score (logrank) test = 3.95  on 2 df,   p=0.1385
```

```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bild et al dataset, stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```


```
> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, FFPE to Bild et al dataset, stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```

Kaplan Meier plots of survival for high and low risk group of Stage I in Bild et al dataset predicted from 59-gene signature

Build the prediction model using the Consortium data, and predict the lung cancer pateints's survival in Bild et al dataset.

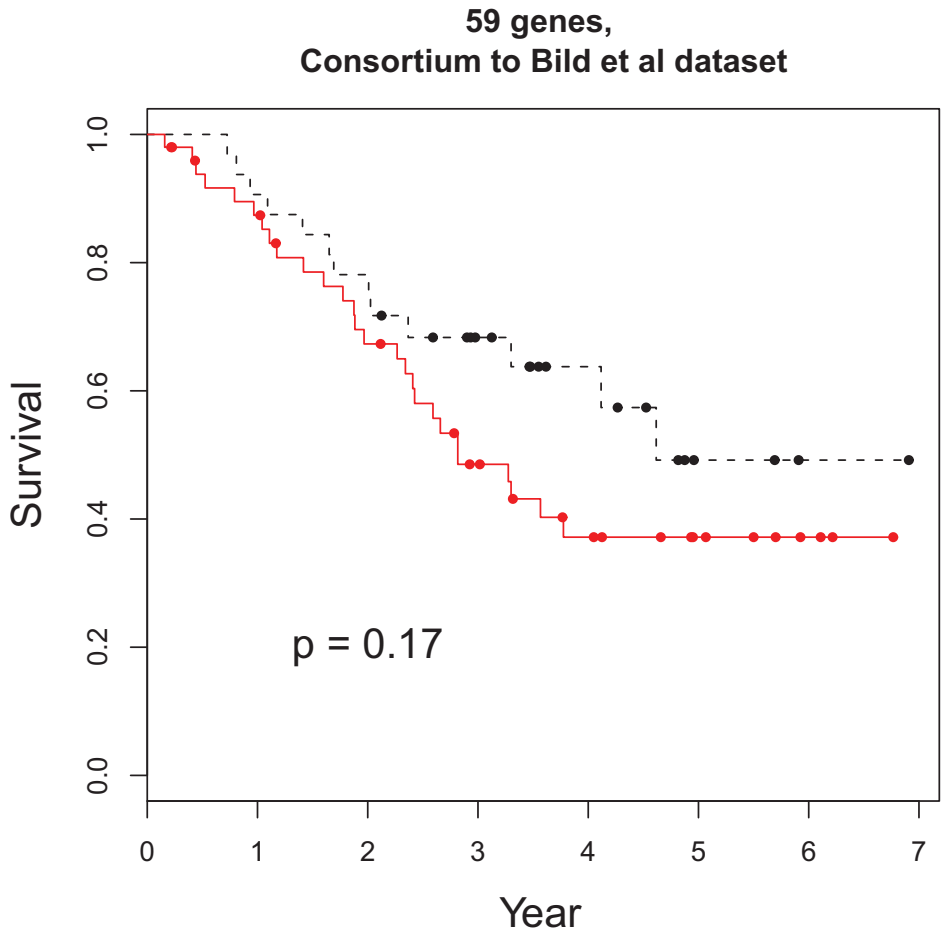
```
> beer.clin.dat <- read.csv("Consortium.clin.csv", row.names = 1)
> beer.expr <- read.csv("Consortium.expr.csv", row.names = 1)
> head(beer.clin.dat)
```

```
                    MICROARRAY dat.SITE death month      year GENDER AGE_AT_DIAGNOSIS
CL2004110909AA   CL2004110909AA     DFCI     0   110 9.1666667 Female              55
CL2004111002AA   CL2004111002AA     DFCI     0    98 8.1666667 Female              41
CL2004111003AA   CL2004111003AA     DFCI     0   110 9.1666667   Male              47
CL20041110100AA CL20041110100AA     DFCI     0    66 5.5000000   Male              73
CL20041110102AA CL20041110102AA     DFCI     1    29 2.4166667 Female              63
CL20041110103AA CL20041110103AA     DFCI     1     7 0.5833333   Male              72
                smoking stage adj_chemo_YN
CL2004110909AA        1     1            0
CL2004111002AA        1     1            0
CL2004111003AA        1     1            1
CL20041110100AA       0     1           NA
CL20041110102AA       1     2           NA
CL20041110103AA       0     1           NA

> beer.clin <- beer.clin.dat[colnames(beer.expr), ]
> which(rownames(beer.clin) != colnames(beer.expr))

integer(0)

> dim(beer.clin)

[1] 442  10

> dim(beer.expr)

[1] 1012  442

> beer.expr <- beer.expr[rownames(g59), ]
> dim(beer.expr)

[1]  59 442

> cb.expr <- data.frame(beer.expr, duke.expr)
> dim(cb.expr)

[1]  59 553

> site <- c(rep("beer", dim(beer.expr)[2]), rep("duke", dim(duke.expr)[2]))
> cb.expr[] <- normalize.quantiles(as.matrix(cb.expr))
> beer.expr <- cb.expr[, site == "beer"]
> duke.expr <- cb.expr[, site != "beer"]

> ind1 <- which(beer.clin$year < 7)
> length(ind1)

[1] 362

> data.train <- list(x = beer.expr[, ind1], y = beer.clin$year[ind1],
+     censoring.status = beer.clin$death[ind1], featurenames = NULL)
> train.obj <- superpc.train(data.train, type = "survival")
> head(duke.clin)
```

```
         Stage      Histology death month      year dead stage
GSM70125    IA       Squamous     0  57.8 4.8166667    0     1
GSM70126    IA       Squamous     0  66.0 5.5000000    0     1
GSM70127  <NA> Adenocarcinoma     0  59.8 4.9833333    0    NA
GSM70128   IIB       Squamous     0  14.0 1.1666667    0     2
GSM70129  IIIA Adenocarcinoma     1  11.6 0.9666667    1     3
GSM70130    IA Adenocarcinoma     0  34.8 2.9000000    0     1
```

```
> data.test <- list(x = duke.expr, y = duke.clin$year, censoring.status = duke.clin$dead,
+     featurenames = NULL)
> fit <- superpc.predict(train.obj, data.train, data.test, threshold = 1,
+     prediction.type = "discrete")
> table(fit$v.pred.1df)

 1  2
38 73

> subset <- which(duke.clin$year < 7 & (!is.na(duke.clin$stage)))
> length(subset)

[1] 82

> surv.fit <- survfit(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)

Call: survfit(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
fit$v.pred.1df=1      32    32      32     13   4.62    3.30      NA
fit$v.pred.1df=2      50    50      50     27   2.82    2.41      NA

> logrank <- survdiff(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> logrank

Call:
survdiff(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 32       13     17.3     1.080      1.91
fit$v.pred.1df=2 50       27     22.7     0.825      1.91

 Chisq= 1.9  on 1 degrees of freedom, p= 0.167

> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
> cox.fit <- coxph(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df +
+     duke.clin$stage, subset = subset)
> summary(cox.fit)
```

```
Call:
coxph(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df +
    duke.clin$stage, subset = subset)

  n= 82

                  coef exp(coef) se(coef)     z Pr(>|z|)
fit$v.pred.1df  0.3449    1.4118   0.3426 1.007   0.3141
duke.clin$stage 0.5636    1.7569   0.2559 2.202   0.0276 *
---
Signif. codes:  0 Ś***Š 0.001 Ś**Š 0.01 Ś*Š 0.05 Ś.Š 0.1 Ś Š 1

                exp(coef) exp(-coef) lower .95 upper .95
fit$v.pred.1df      1.412     0.7083    0.7214     2.763
duke.clin$stage     1.757     0.5692    1.0640     2.901

Rsquare= 0.073   (max possible= 0.979 )
Likelihood ratio test= 6.21  on 2 df,   p=0.04477
Wald test            = 6.77  on 2 df,   p=0.0339
Score (logrank) test = 7.1  on 2 df,   p=0.02878

> gc()

         used (Mb) gc trigger  (Mb) max used  (Mb)
Ncells 380760 10.2     984024  26.3   984024  26.3
Vcells 483253  3.7   29563329 225.6 36937229 281.9

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bild et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bild et al dataset")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```



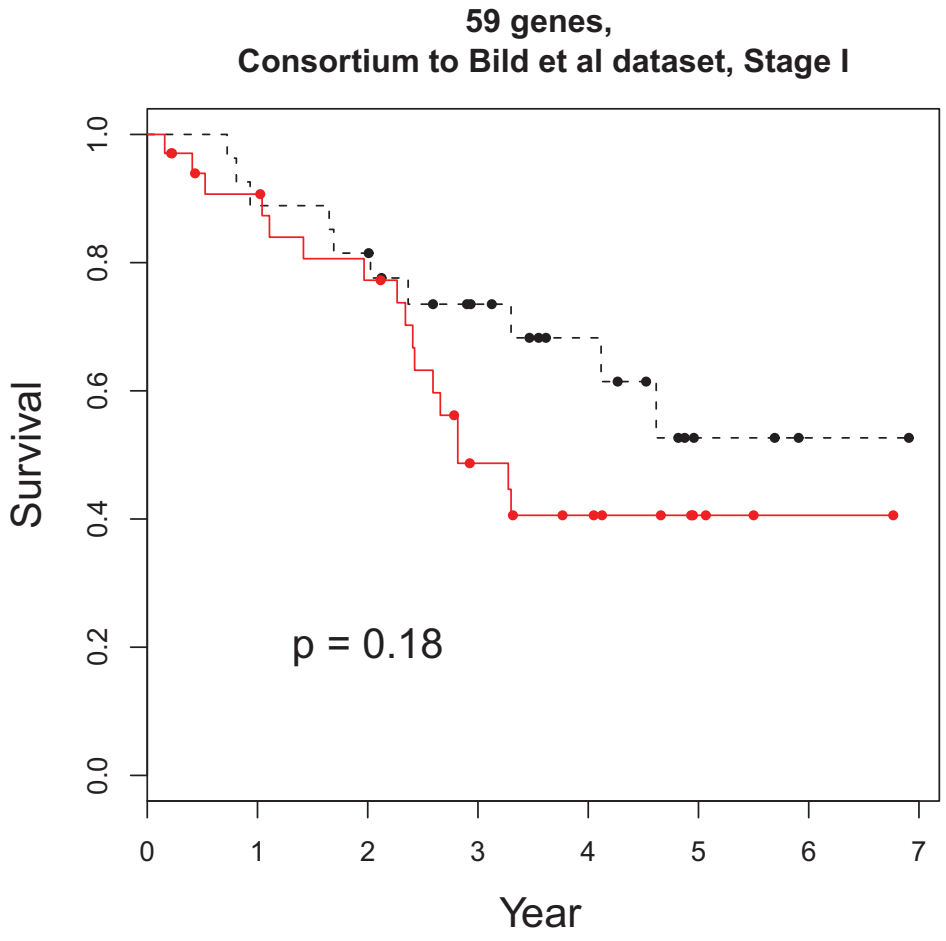
Kaplan Meier Plot of survival for the high and low risk groups in Bild et al dataset, predicted from the 59 gene-signature.

Now predict the stage I patients only.

```
> subset <- which(duke.clin$year < 7 & duke.clin$stage == 1)
> length(subset)

[1] 61

> surv.fit <- survfit(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> print(surv.fit)

Call: survfit(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                records n.max n.start events median 0.95LCL 0.95UCL
```

```
fit$v.pred.1df=1      27    27      27     10     NA    4.12      NA
fit$v.pred.1df=2      34    34      34     17   2.82    2.43      NA
```

```
> logrank <- survdiff(Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
+     subset = subset)
> logrank

Call:
survdiff(formula = Surv(duke.clin$year, duke.clin$dead) ~ fit$v.pred.1df,
    subset = subset)

                 N Observed Expected (O-E)^2/E (O-E)^2/V
fit$v.pred.1df=1 27       10     13.5     0.886      1.78
fit$v.pred.1df=2 34       17     13.5     0.880      1.78

 Chisq= 1.8  on 1 degrees of freedom, p= 0.182

> pv <- pchisq(logrank$chisq, 1, lower.tail = F)
> gc()

         used (Mb) gc trigger  (Mb) max used  (Mb)
Ncells 380654 10.2     984024  26.3   984024  26.3
Vcells 483180  3.7   23650663 180.5 36937229 281.9

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bild et al dataset, Stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)

> plot(surv.fit, col = 1:2, lty = c(2, 1), xlab = "Year", ylab = "Survival",
+     mark = 20, cex.lab = 1.5, main = "59 genes, \nConsortium to Bild et al dataset, Stage I")
> text(2, 0.2, pv.expr(pv), cex = 1.5)
```



Kaplan Meier plots of survival for high and low risk group of stage I patients in Bild et al dataset predicted from 59-gene signature

5 Appendix

This computation was performed in the following environment:

```
> sessionInfo()

R version 2.10.0 (2009-10-26)
i386-pc-mingw32

locale:
[1] LC_COLLATE=English_United States.1252  LC_CTYPE=English_United States.1252
[3] LC_MONETARY=English_United States.1252 LC_NUMERIC=C
[5] LC_TIME=English_United States.1252
```

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

other attached packages:
[1] preprocessCore_1.8.0 affy_1.24.2          Biobase_2.6.1        superpc_1.06
[5] survival_2.35-8

loaded via a namespace (and not attached):
[1] affyio_1.14.0
```