VANGUARD Microarray Study

Li Shen, Jing Wang, and Kevin R. Coombes
Dept. of Bioinformatics and Computational Biology
The University of Texas M.D. Anderson Cancer Center (MDACC)

25 February 2011

Contents

1 Executive Summary

1.1 Statistical Methods

Microarray data were processed using the RMA algorithm as implemented in the aroma.affymetrix package of R. We use a mixed-effects model on the processed data to understand the expression patterns of each gene. the fixed effects represent

- Batch: array data were collected (over time) in two batches from two different laboratories. This effect is a nuisance and not one we really want to use to make inferences.

- DetMap: This is a detailed map of the site where each (bronchial brushing) sample was obtained. There are four levels of this variable, basically corresponding to its distance from the site of the primary tumor. The levels are ADJ (adjacent to the tumor), NON-ADJ (same half of the lung but not adjacent to the tumor), MC (main carina), and CONTRA (from the contralateral lung).
- Time.point: Samples were collected at four time points (0, 12, 24, or 36 months).
- We also allow an interaction term between DetMap and Time.point.

Random effects primarily account for the fact that we have multiple samples from the same patient (Case). We try to fit a model that includes a different starting point and slope over time for each case. For some probes, we are unable to fit a model that includes different slopes for each patient; in this case, we fall back to a single (base-level) random effect foe each case.

1.2 Results

- Batch is a dominant effect on gene expression (**Figure 1**).
- After adjusting for batch (as part of the mixed-effects model), both site and time are significant in some genes, with site more important than time (**Figure 1**).
- There is very weak (and perhaps no real) evidence that the interaction term is ever significant (**Figure 1**, **Figure 3**).

2 Loading the Data

2.1 R Packages

We start by loading the packages that will be needed for our analysis.

```
> require(nlme)
> library(lattice)
> library(RColorBrewer)
> library(ClassComparison)
> library(ClassDiscovery)
```

```
by using mclust, invoked on its own or through another package,
you accept the license agreement in the mclust LICENSE file
and at http://www.stat.washington.edu/mclust/license.txt
```

We modify some of the default display options for the trellis/lattice plots.

```
> x <- trellis.par.get("plot.symbol")
> x$pch <- 16
> x$col <- "#00aa60"
> trellis.par.set("plot.symbol", x)
> rm(x)
```

2.2 Affymetrix data

The processed Affymetrix data can be found in the following location:

```
> basedir <- ifelse(.Platform$OS == "windows", "//mdadqsfs02", "/data")
> datadir <- file.path(basedir, "bioinfo2", "Lung-HN", "Wistuba-VANGUARD",
+     "Analysis")
```

Here we load the data.

```
> load(file.path(datadir, "gExpr-RMA-Aroma.RData"))
```

In order to understand what is contained in the dataset, we explore the objects that we just loaded.

```
> ls()
```

```
[1] "basedir"  "datadir"  "gExpr"    "normData" "si"
```

```
> class(gExpr)
```

```
[1] "data.frame"
```

```
> dim(gExpr)
```

```
[1] 33252   396
```

```
> class(normData)
```

```
[1] "data.frame"
```

```
> dim(normData)
```

```
[1] 33252   391
```

```
> normData <- as.matrix(normData)
> class(si)
```

```
[1] "data.frame"
```

```
> dim(si)
```

```
[1] 391  61
```

```
> all(colnames(normData) == rownames(si))
```

```
[1] TRUE
```

2.3 Gene/Probe Annotations

Here we load the annotations for the probes on the ST 1.0 array.

```
> annot <- read.csv(file.path(datadir, "RNW",
+                             "Human Gene 1.0 ST annotations for Li Shen.csv"),
+                   header=TRUE, as.is=TRUE, na.strings=c("NA","Un", ""),
+                   row.names=1)
> all(rownames(normData) %in% rownames(annot))
```

```
[1] TRUE
```

```
> annot <- annot[rownames(normData),]
> annot$Symbol <- factor(annot$Symbol)
> annot$UGCluster <- factor(annot$UGCluster)
> annot$Chromosome <- factor(annot$Chromosome,
+                            levels=c(1:22, "X", "Y"))
> annot$Cytoband <- factor(annot$Cytoband)
> summary(annot)
```

```
     Name            Accession            UGCluster           Symbol
 Length:33252       Length:33252       Hs.559040:   26   LOC349196:   26
 Class :character   Class :character   Hs.199343:   13   DUX4     :   12
 Mode  :character   Mode  :character   Hs.196086:   12   FAM90A1  :   12
                                       Hs.460179:   12   MGC72080 :   12
                                       Hs.553518:   12   SMG1     :   12
                                       (Other)  :21386   (Other)  :21697
                                       NA's     :11791   NA's     :11481
   EntrezID           Chromosome       Cytoband          GO
 Min.   :        1   1      : 2241   6p21.3 :  356   Length:33252
 1st Qu.:     7166   19     : 1442   19p13.3:  210   Class :character
 Median :    51585   2      : 1385   16p13.3:  203   Mode  :character
 Mean   :  1588044   11     : 1374   19p13.2:  170
 3rd Qu.:   124778   6      : 1355   Xq28   :  133
 Max.   :100499221   (Other):13971   (Other):20629
 NA's   :    11481   NA's   :11484   NA's   :11551
```

2.4 Clinical Data

Now we clean up the clinical data.

```
> si$Batch <- factor(si$Batch)
> si$Gender <- factor(si$Gender)
> si$Diagnosis..Histology. <- factor(si$Diagnosis..Histology.)
> colnames(si)[19] <- "Histology"
```

```
> si$Off.study_Reason <- factor(si$Off.study_Reason)
> si$Differentiation <- factor(si$Differentiation)
> si$Leison.Site <- factor(si$Leison.Site)
> si$Anatomical_site <- factor(si$Anatomical_site)
> si$site.of.collection <- factor(si$site.of.collection)
> si$Contralateral <- factor(si$Contralateral)
> dmlev <- c("ADJ", "NON-ADJ", "MC", "CONTRA")
> si$DetMap <- factor(si$DetMap, levels = dmlev)
> mlev <- c("ADJ", "NON-ADJ", "MC")
> si$Map <- factor(si$Map, levels = mlev)
> si$Code.4.time.point <- factor(si$Code.4.time.point)
> si$Code.4.Site.of.collection <- factor(si$Code.4.Site.of.collection)
> si$pT <- factor(si$pT)
> si$pN <- factor(si$pN)
> si$Final.Pat.Stage <- factor(si$Final.Pat.Stage)
> si$EGFR.status <- factor(si$EGFR.status)
> si$KRAS.status <- factor(si$KRAS.status)
> si$V_Case.ID.Inclusion_number. <- factor(paste("P", si$V_Case.ID.Inclusion_number.,
+     sep = ""))
> colnames(si)[8] <- "Case"
> simplify <- c(2, 8, 27, 31, 34, 12, 14:20, 22, 24, 43:47, 13)
> rm(dmlev, mlev)
> summary(si[, simplify])
```

```
 Batch         Case          DetMap          Map         Time.point
 I : 71   P1     : 25   ADJ    : 62   ADJ    : 62   Min.   : 0.00
 II:320   P31    : 24   NON-ADJ:107   NON-ADJ:268   1st Qu.: 0.00
          P40    : 24   MC     : 60   MC     : 60   Median :12.00
          P44    : 24   CONTRA :161   NA's   :  1   Mean   :15.87
          P6     : 24   NA's   :  1                 3rd Qu.:24.00
          P18    : 23                               Max.   :36.00
          (Other):247
          Off.study_Reason Gender  DOB..DOBirth.       DOSurgery
                  : 17     F:134   Length:391         Length:391
 N                :332     M:257   Class :character   Class :character
 Y_died           : 25             Mode  :character   Mode  :character
 Y_recurrence lung: 17
```

```
 DOInclusion          Surgery                   Histology   Differentiation
 Length:391         Length:391         Adenocarcinoma:309   MOD     :180
 Class :character   Class :character   Squamous      : 82   MOD-POOR: 24
```

```
 Mode  :character   Mode  :character                        POOR    : 35
                                                           W       : 42
                                                           WELL    : 23
                                                           NA's    : 87

 Anatomical_site Contralateral    pT      pN      Final.Pat.Stage      EGFR.status
 LLL: 58               :  1    1A  :203   0:366   I  : 23         MUT del E19: 18
 LUL:102         CONTRA:161    1B  : 43   1: 25   IA :221         WT         : 71
 RLL: 80         IPSI  :169    2A  : 98           IB : 98         NA's       :302
 RML: 17         MC    : 60    2B  : 24           IIA: 49
 RUL:134                       NA's: 23


     KRAS.status
 MUT cod 12: 24
 WT        : 65
 NA's      :302
```

```
> tmp <- si[match(levels(as.factor(si$MRN)), si$MRN), c(8, 13)]
> tmp2 <- read.csv(file.path(datadir, "RNW", "Copy of VANGUARD_06012011_BRONCHIAL BRUSHES.csv")
+     header = TRUE, as.is = TRUE, na.strings = c("NA", "Un", ""), row.names = "MDAH")
> ci <- data.frame(tmp, Event = tmp2[match(tmp$MRN, rownames(tmp2)),
+     "Event"])
> ci <- ci[order(ci$Case), ]
> rownames(ci) <- ci$Case
> Event.col <- rep("Suspicion", nrow(ci))
> Event.col[which(ci$Event == "YES")] <- "Recurrence"
> Event.col[which(ci$Event == "NO")] <- "No"
> ci <- cbind(ci, Event.col)
> ci <- ci[, c(1, 4)]
> foo <- merge(si, ci, by = "Case")
> for (i in 1:nrow(foo)) {
+     w <- which(si$Experiment.Names == foo[i, "Experiment.Names"])
+     if (length(w) != 1)
+         stop("no unique match")
+     rownames(foo)[i] <- rownames(si)[w]
+ }
> foo <- foo[rownames(si), ]
> foo <- foo[, c(colnames(si), "Event.col")]
> all(si[, 1:61] == foo[, 1:61])

[1] NA
```

```
> si <- foo
> rm(foo, ci, Event.col, tmp, tmp2, i, w)
```

These colors will be used in some of the later plots.

```
> ev.col <- c(No = "white", Recurrence = "black", Suspicion = "gray")
> ev.colors <- ev.col[as.character(si$Event.col)]
> hist.col <- c(Adenocarcinoma = "orange", Squamous = "purple")
> hist.colors <- hist.col[as.character(si$Histology)]
> batch.col <- c(I = "cyan", II = "magenta")
> batch.colors <- batch.col[as.character(si$Batch)]
> site.col <- brewer.pal(5, "Reds")[2:5]
> names(site.col) <- levels(si$DetMap)
> site.col <- c(ADJ = "red", `NON-ADJ` = "gold", MC = "green", CONTRA = "blue")
> site.colors <- site.col[as.numeric(si$DetMap)]
> time.col <- brewer.pal(5, "Blues")[2:5]
> names(time.col) <- seq(0, 36, 12)
> time.colors <- time.col[1 + round(si$Time.point/12)]
> case.col <- c(brewer.pal(3, "Reds"), brewer.pal(3, "Blues"), brewer.pal(3,
+     "Greens"), brewer.pal(3, "Purples"), brewer.pal(3, "Greys")[2:3],
+     brewer.pal(12, "Paired")[11], brewer.pal(9, "Set1")[6:7], brewer.pal(8,
+         "Dark2")[4], "#88e1e1")
> names(case.col) <- levels(si$Case)
> case.colors <- case.col[as.numeric(si$Case)]
> colorfacs <- list(Case = list(fac = si$Case, col = case.col), Site = list(fac = si$DetMap,
+     col = site.col), Time = list(fac = factor(si$Time.point), col = time.col),
+     Batch = list(fac = si$Batch, col = batch.col), Histology = list(fac = si$Histology,
+         col = hist.col), Event = list(fac = si$Event.col, col = ev.col))
> cr <- colorRampPalette(c("white", brewer.pal(9, "Oranges")))
> tf <- function(x) x^0.15
```

3 Statistical Modeling

We use a mixed-effects model to understand the expression patterns of each gene. Fixed effects represent

- Batch: array data were collected (over time) in two batches from two different laboratories.
- DetMap: This is a detailed map of the site where each (bronchial bruishing) sample was obtained. There are four levels of this variable, basically corresponding to its distance from the site of the primary tumor. The levels are ADJ (adjacent to the tumor), NON-ADJ (same half of the lung but not adjacent to the tumor), MC (main carina), and CONTRA (from the contralateral lung).

- Time.point: Sample were collected at four time points (0, 12, 24, or 36 months).
- We also allow an interaction term between DetMap and Time.point.

Random effects primarily account for the fact that we have multiple samples from the same patient (Case). We try to fit a model that includes a different starting point and slope over time for each case. For some genes, we are unable to fit a model that includes different slopes for each patient; in this case, we fall back to a single (base-level) random effect for each case.

```
> gene.label <- function(gene) {
+     ifelse(is.na(annot[gene, "Symbol"]), rownames(annot)[gene], as.character(annot[gene,
+         "Symbol"]))
+ }
> f <- "modlist.rda"
> if (file.exists(f)) {
+     load(f)
+ } else {
+     modlist <- lapply(1:nrow(normData), function(x) 1)
+     for (gene in 1:nrow(normData)) {
+         gl <- gene.label(gene)
+         cat(gl, "\n", file = stderr())
+         pinfo <- 1:nrow(si)
+         pclin <- si[pinfo, simplify]
+         x <- normData[gene, pinfo]
+         tempd <- data.frame(si[, c(2, 8, 27, 34, 19)], Y = x)
+         foo <- na.omit(tempd)
+         foo$Time.point <- foo$Time.point/12
+         foo <- foo[order(foo$Time.point, foo$Case), ]
+         gd <- groupedData(Y ~ Time.point | Case, data = foo, outer = ~Histology)
+         mod6 <- try(lme(Y ~ Batch + DetMap * Time.point, data = gd,
+             random = ~Time.point | Case, method = "ML"))
+         if (inherits(mod6, "try-error")) {
+             mod6 <- (lme(Y ~ Batch + DetMap * Time.point, data = gd,
+                 random = ~1 | Case, method = "ML"))
+         }
+         modlist[[gene]] <- mod6
+     }
+     rm(gene, gl, pinfo, pclin, x, tempd, foo, gd, mod6)
+     save(modlist, file = f)
+ }
> rm(f)
```

3.1 Relative Importance of Fixed Effects

In the next block of code, we extract the p-values from the statistical models. To illustrate what we expect to get, we first show an example.

```
> x <- modlist[[1]]
> anova(x)
```

```
                  numDF denDF  F-value p-value
(Intercept)           1   363 7571.805  <.0001
Batch                 1   363   17.882  <.0001
DetMap                3   363    0.185  0.9068
Time.point            1   363    8.879  0.0031
DetMap:Time.point     3   363    2.944  0.0330
```

This example shows that we get separate p-values for each of the fixed effects included in the model, stored as the fourth column of the ANOVA table. So, we extract that column for each gene and save it.

```
> f <- "pvals.rda"
> if (file.exists(f)) {
+     load(f)
+ } else {
+     lap <- lapply(modlist, function(x) {
+         a <- anova(x)
+         a[, 4]
+     })
+     pvals <- matrix(unlist(lap), ncol = 5, byrow = TRUE)
+     a <- anova(modlist[[1]])
+     colnames(pvals) <- rownames(a)
+     rownames(pvals) <- rownames(normData)
+     pvals <- as.data.frame(pvals)
+     rm(lap, a, x)
+     save(pvals, file = f)
+ }
> rm(f)
```

We fit beta-uniform-mixture (BUM) models to the p-values for each of the four fundamental terms in the statistical model. We also plot histograms for the distributions of these p-values (**Figure 1**). It is clear that batch is an extremely large effect, being present in almost every gene. However, after adjusting for batch, both the site and the time produce clear signs of changing (for some genes) across the samples in a consistent manner, with site being slightly more important than time.

```
> bsite <- Bum(pvals$DetMap)
> countSignificant(bsite, alpha = 0.01)
```

```
[1] 1165
```

```
> btime <- Bum(pvals$Time.point)
> countSignificant(btime, alpha = 0.01)
```

```
[1] 348
```

```
> bbat <- Bum(pvals$Batch)
> countSignificant(bbat, alpha = 0.01)
```

```
[1] 25064
```

```
> binter <- Bum(pvals$"DetMap:Time.point")
> countSignificant(binter, alpha = 0.01)
```

```
[1] 0
```

3.2 Interaction Between Time and Site

Next, we would like to better understand the interaction term in the model. The histogram for the p-values associated with the interaction is a slightly odd shape, in that the standard BUM model clearly does not fit the distribution (**Figure 1**).

We start by asking whether the significance of site and time is correlated (tends to happen for the same genes) or independent. A smooth scatter plot of the logistically transsformed p-values strongly suggests that they are independent (**Figure 2**). Directly counting the overlap at a 5% significance level agrees with this assessment.

```
> ss <- countSignificant(bsite, alpha = 0.05)
> ss
```

```
[1] 4686
```

```
> tt <- countSignificant(btime, alpha = 0.05)
> tt
```

```
[1] 1395
```

```
> observed <- sum(selectSignificant(bsite, alpha = 0.05) & selectSignificant(btime,
+     alpha = 0.05))
> expected <- ss * tt/nrow(normData)
> round(c(OBS = observed, EXP = expected))
```

```
OBS EXP
203 197
```

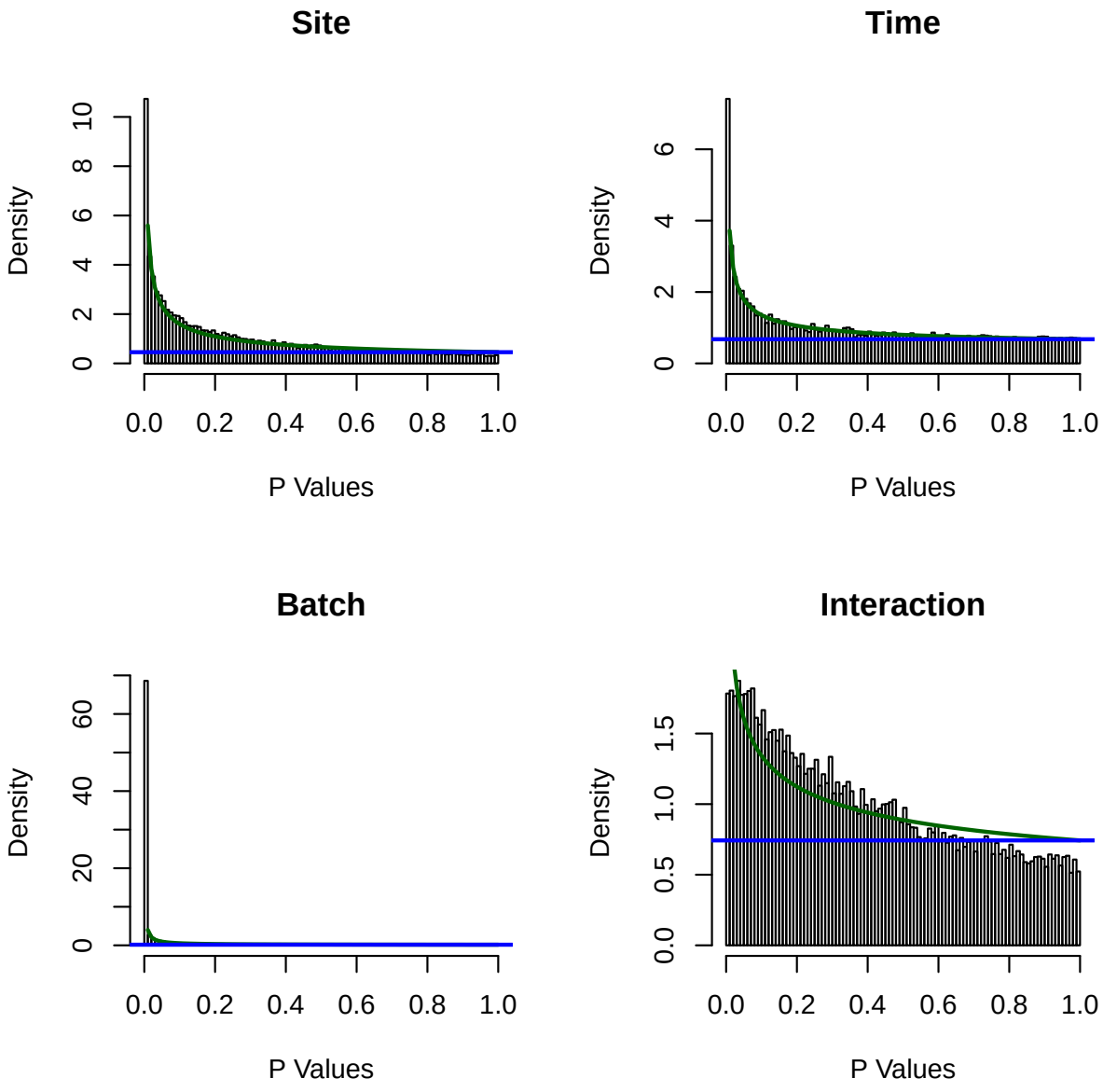

Figure 1: Histograms of p-values for the fixed effects.

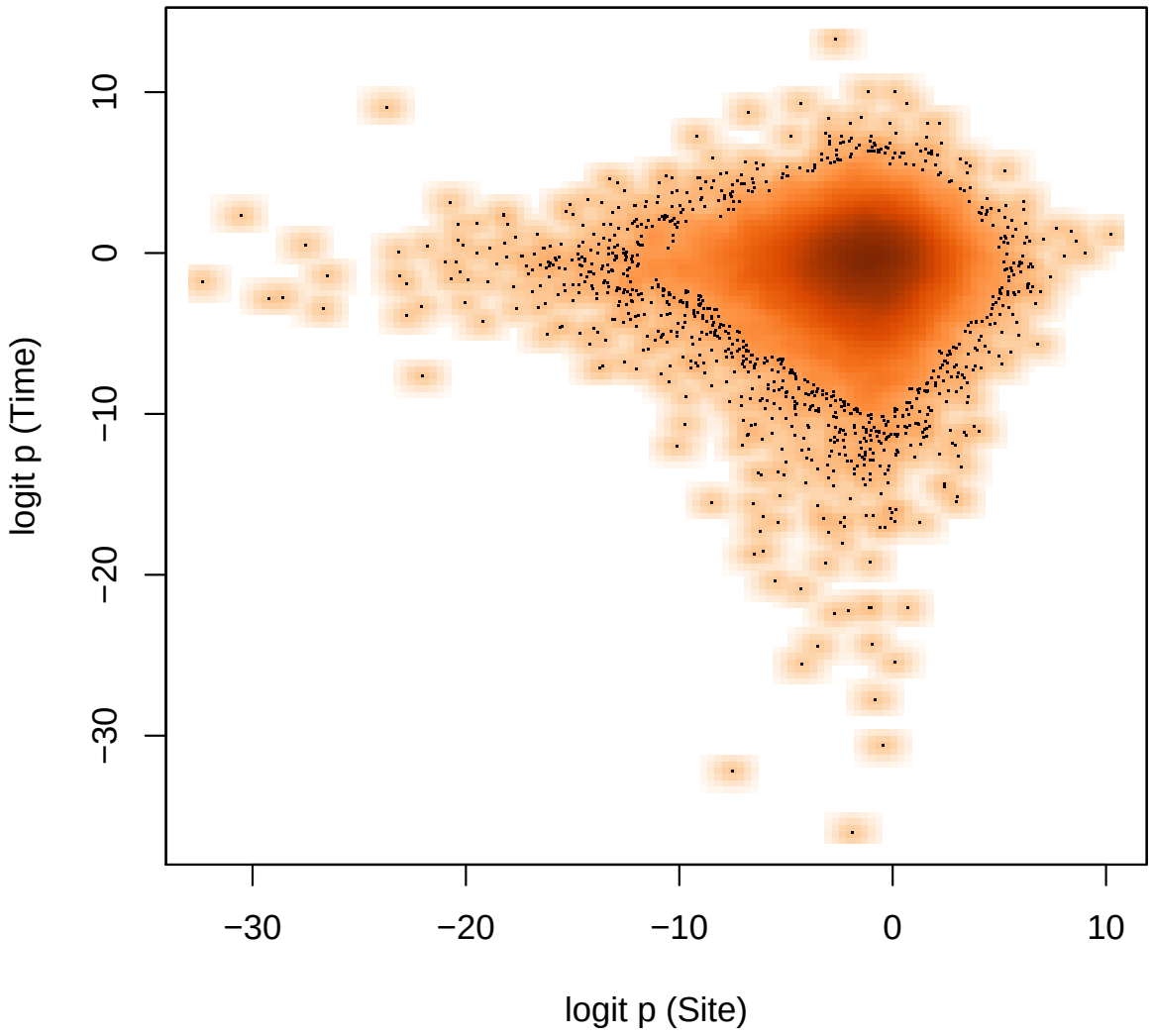

Figure 2: Smoothed scatter plot logistically transformed p-values from site and time.

Now we restrict to the set of genes where there is some very weak evidence that both time and site have significant effects. There are about 3000 probes for which both time and site have $p < 0.20$.

```
> cutter <- 0.2
> onesig <- selectSignificant(bsite, alpha = cutter) & selectSignificant(btime,
+     alpha = cutter)
> sum(onesig)

[1] 3062
```

We can fit a BUM model to the interaction p-values associated with this subset of probes. We still get a fairly small number of genes, even with a 30% FDR.

```
> bp <- Bum(pvals$"DetMap:Time.point"[onesig])
> countSignificant(bp, alpha = 0.3)

[1] 25
```

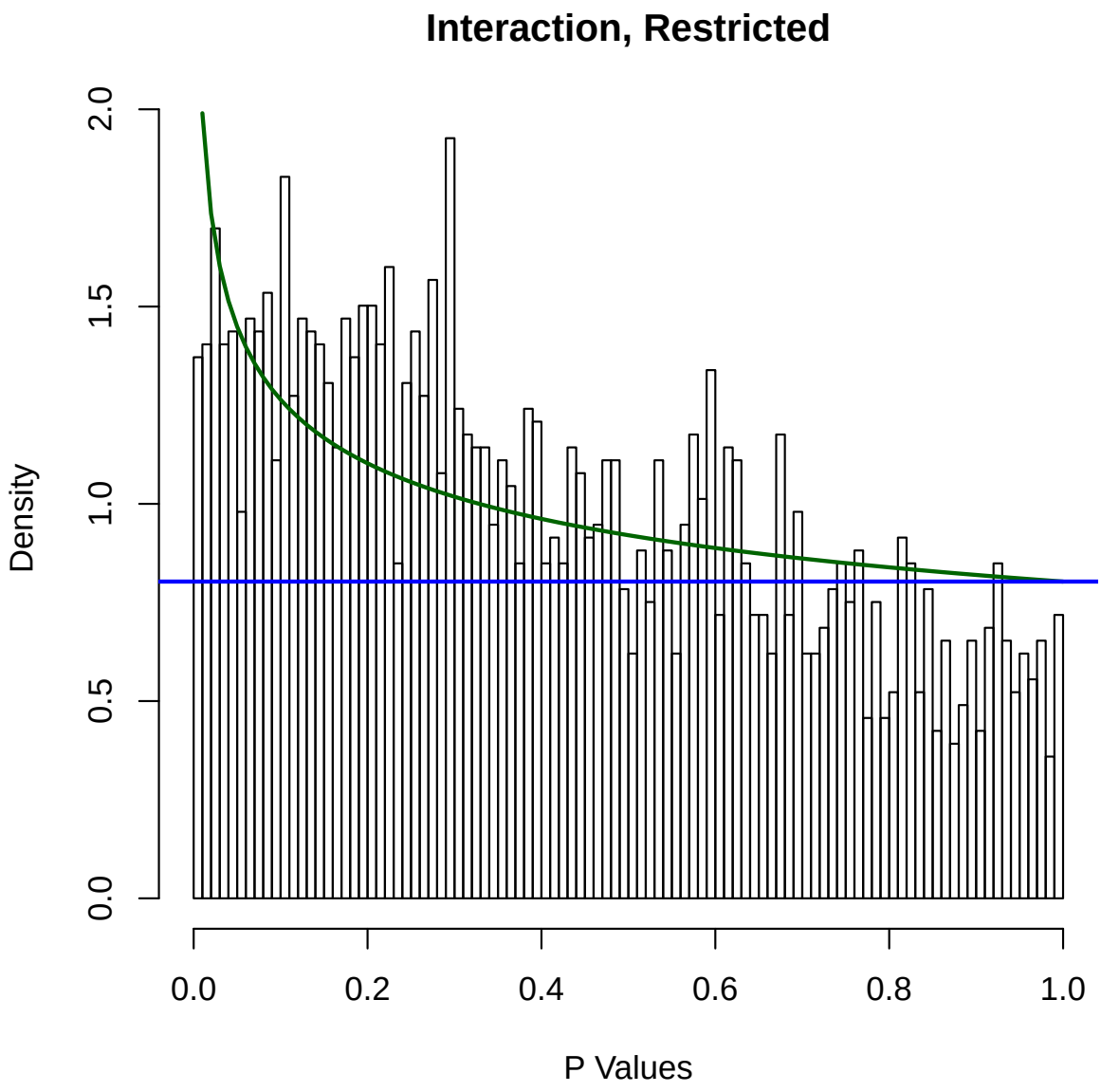

Figure 3: Histogram of p-values for the interaction between time and site, restricted to genes where both main effects show a trend toward significance.

3.3 Random Effects

In our model, we tried to fit a random intercept and a random slope (for trends over time) for each case. For some genes, we were unable to estimate all of these coefficients, and were forced to drop the slope terms for individual cases. The next block of code collects the indicators that separate genes into two categories deopending on whether or not we could add the random-effects slopes into the model.

```
> ref <- unlist(lapply(modlist, function(x) {
+     dim(ranef(x))[2]
+ }))
> table(ref)

ref
    1     2
 7046 26206
```

We were able to include random-effects slopes for 26206 genes and were unable to do so for the remaining 7046 genes. A slightly larger percentage of fixed-effects time parameters are significant when we cannot fit a random slope model (**Figure 4**).

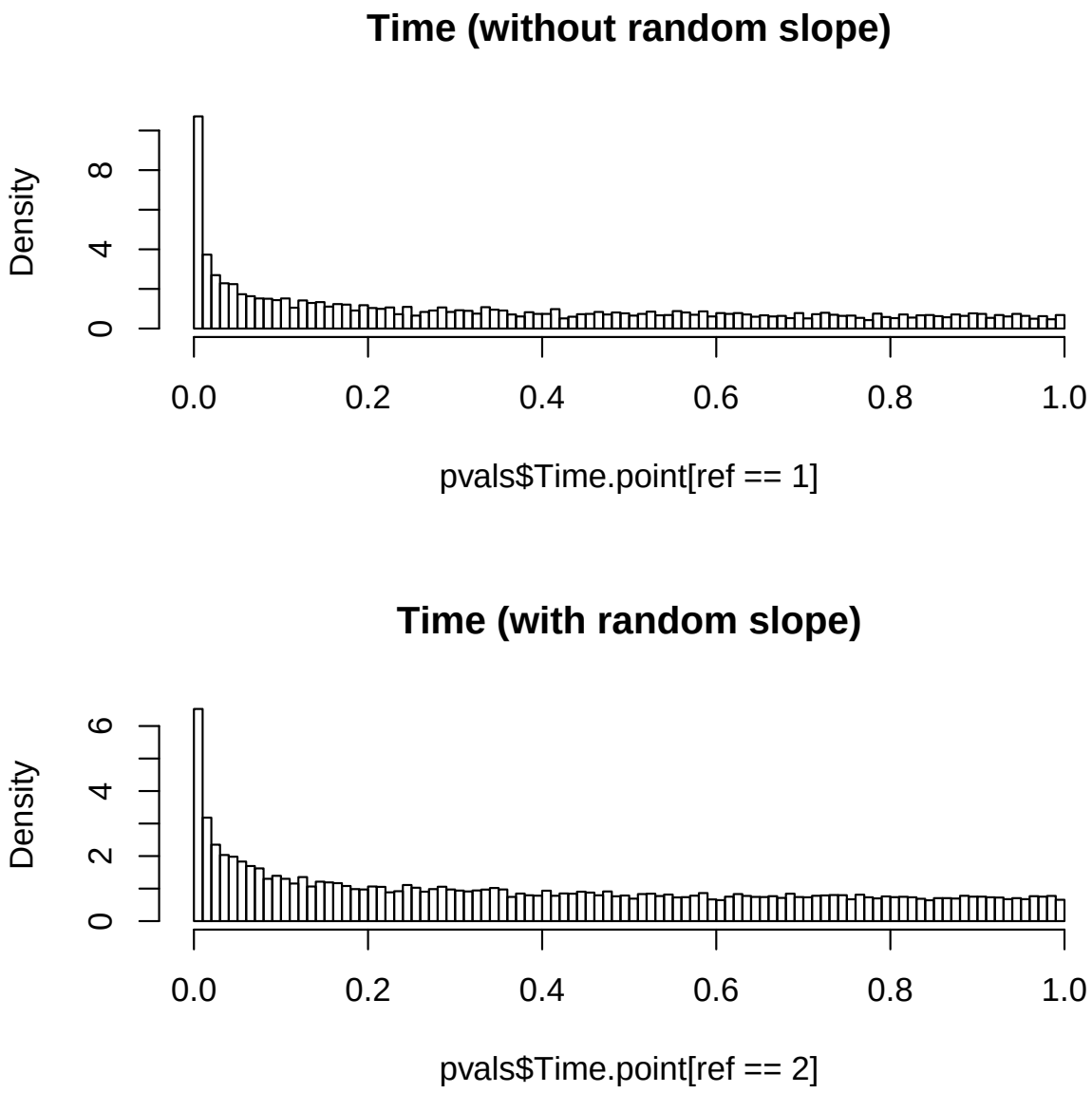

Figure 4: Histograms of the p-values for significant change over time depending on whether or not the model was able to fit different slopes for different patients.

4 Batch Correction

In order to generate additional plots, we need to adjust for the batch effects that are imposed on almost all genes. The information that we need to make this adjustment is already contained in the (fixed-effects) coefficients in the statistical models that we computed for each gene. For example,

```
> x <- modlist[[1]]
> fixef(x)
```

```
             (Intercept)                  BatchII            DetMapNON-ADJ
             6.115740148              0.291131961              0.079230372
                DetMapMC            DetMapCONTRA               Time.point
             0.117021531              0.430656803              0.287538998
DetMapNON-ADJ:Time.point      DetMapMC:Time.point  DetMapCONTRA:Time.point
             0.003167197             -0.071842053             -0.250375185
```

The next block of code extracts all of the fixed-effects coefficients from the statistical models.

```
> f <- "fixcoef.rda"
> if (file.exists(f)) {
+     load(f)
+ } else {
+     fip <- lapply(modlist, fixef)
+     fixcoef <- matrix(unlist(fip), ncol = 9, byrow = TRUE)
+     colnames(fixcoef) <- names(fixef(x))
+     rownames(fixcoef) <- rownames(normData)
+     fixcoef <- as.data.frame(fixcoef)
+     rm(fip, x)
+     save(fixcoef, file = f)
+ }
> rm(f)
```

Now we use the batch coefficients to adjust the data.

```
> adjData <- normData
> temp <- sweep(adjData[, si$Batch == "I"], 1, fixcoef$BatchII, "+")
> adjData[, si$Batch == "I"] <- temp
```

4.1 Genes That Are Different By Site

We start by selecting the genes that are signifcantly different between sites, based on a 1% false discovery rate (FDR).

```
> ssel <- selectSignificant(bsite, alpha = 0.01)
> site.specific <- adjData[ssel, ]
```

Now we cluster the samples using these genes (**Figure 5**).

```
> ssc <- hclust(distanceMatrix(site.specific, "pearson"), "ward")
```

There are more "adjacent" samples in the left branch and more "main carina" and "contralateral" samples in the right branch of the dendorgram; this difference is statistically significant.

```
> table(cutree(ssc, k = 2))

  1   2
148 243

> tab.site <- table(cutree(ssc, k = 2), si$DetMap)
> tab.site

    ADJ NON-ADJ  MC CONTRA
  1  35      44  19     49
  2  27      63  41    112

> round(tab.site[1, ]/apply(tab.site, 2, sum) * 100, 1)

    ADJ NON-ADJ      MC  CONTRA
   56.5    41.1    31.7    30.4

> fisher.test(tab.site)

        Fisher's Exact Test for Count Data

data:  tab.site
p-value = 0.002749
alternative hypothesis: two.sided
```

We also want to cluster the genes that differ between sites. With both genes and samples clustered, we can construct a heatmap (**Figure 6**). The patterns in the heatmap suggest that there are at least eight different gene expression patterns, which are indicated by the colorbar along the left side.

```
> ggc <- hclust(distanceMatrix(t(site.specific), "pearson"), "ward")
> scut <- cutree(ggc, k = 8)
> scut.colors <- brewer.pal(8, "Dark2")[scut]
> sclass <- as.numeric(ssel)
> sclass[ssel] <- scut
> table(sclass)

sclass
    0     1     2     3     4     5     6     7     8
32087   263    38   130   348    96   115   133    42
```

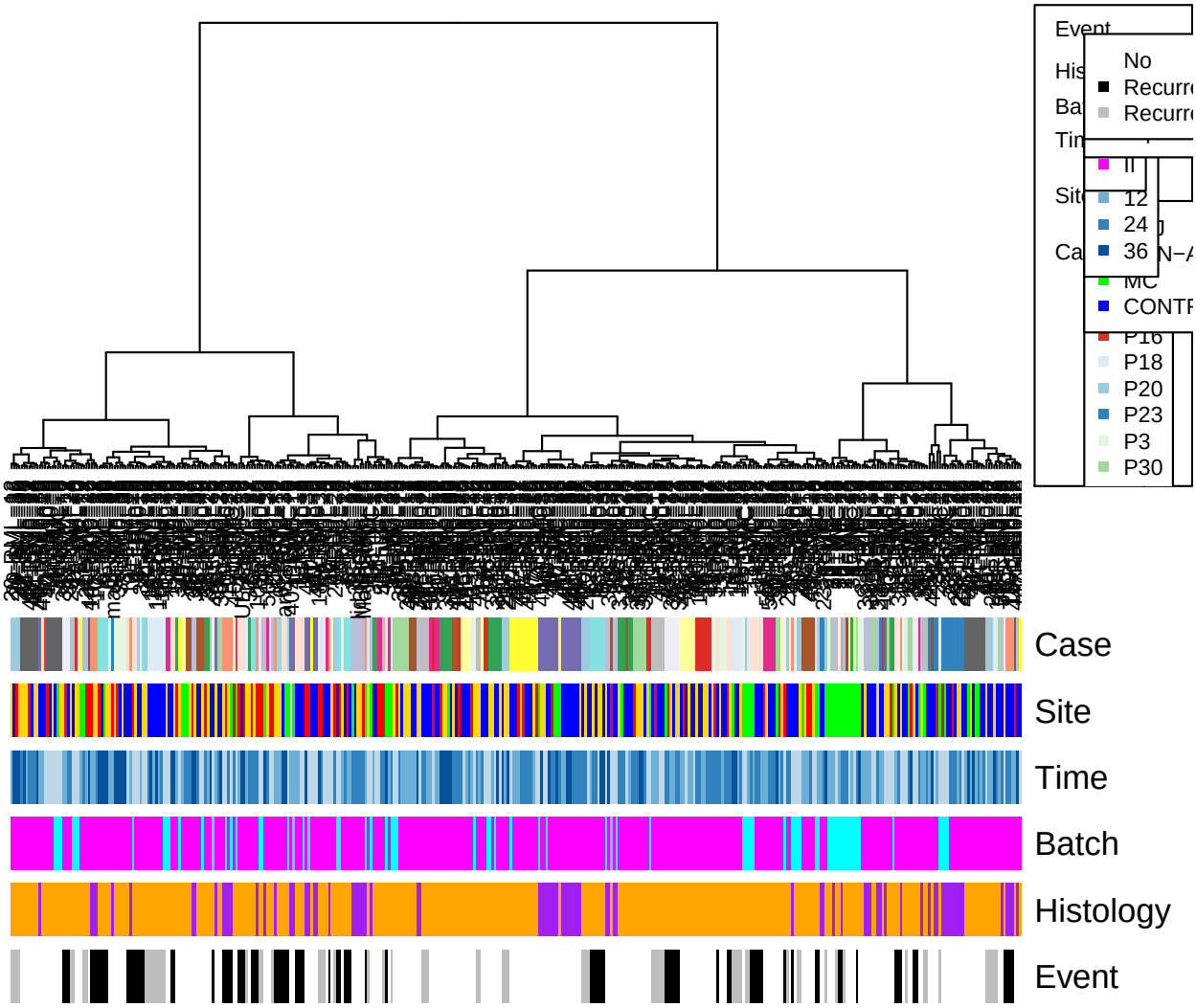

Figure 5: Hierarchical clustering of samples (using Pearson correlation and Ward's linkage) based on genes that are significantly different between **sites**.

```
> ssite <- t(scale(t(site.specific)))
> ssite[ssite > 5] <- 5
> ssite[ssite < -5] <- -5
```

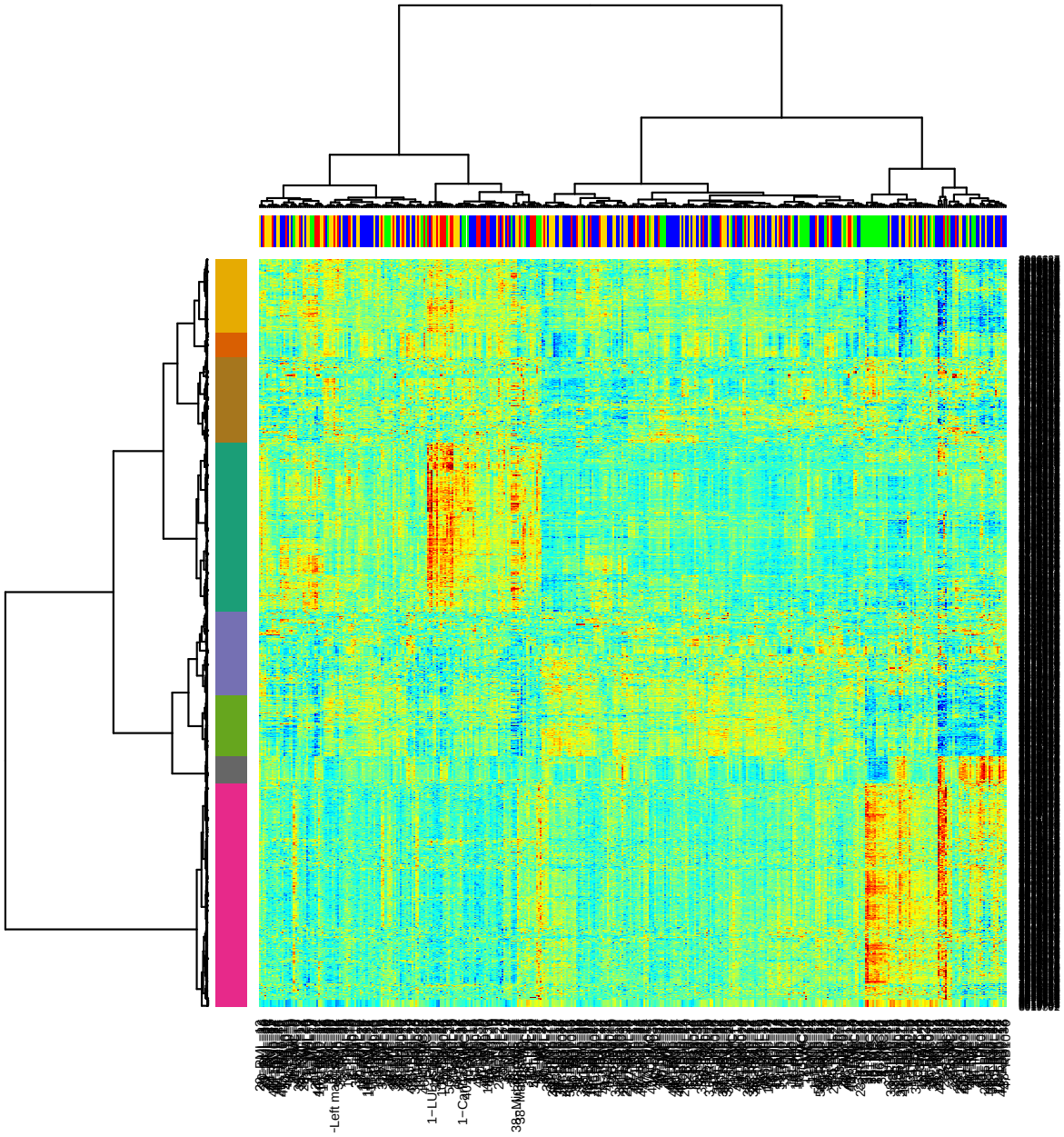

Figure 6: Two-dimensional clustering heatmap image of the genes selected because they differ by **site**. Top colorbar indicates site as in the previous plot. Left colorbar uses this clustering to define 8 types of gene expression patterns.

4.2 Genes That Change Over Time

In this section, we study genes that change (linearly) over time, regardless of the site. We start by selecting such genes with FDR equal to 5%.

```
> tsel <- selectSignificant(btime, alpha = 0.05)
> time.lapse <- adjData[tsel, ]
```

Next, we cluster the samples using these genes (**Figure 7**). The main branches are clearly unbalanced with respect to time; the starting time is much more likely to occur in the right-hand branch and the final time point is much more likely to appear in the left hand branch. In fact, over time, the samples seem to be moving from the right to the left branch.

```
> sc <- hclust(distanceMatrix(time.lapse, "pearson"), "ward")

> table(cutree(sc, k = 2))

  1   2
222 169

> tab.time <- table(cutree(sc, k = 2), si$Time.point)
> tab.time

     0 12 24 36
  1 45 56 70 51
  2 64 52 43 10

> round(tab.time[1, ]/apply(tab.time, 2, sum) * 100, 1)

   0   12   24   36
41.3 51.9 61.9 83.6

> fisher.test(tab.time)

        Fisher's Exact Test for Count Data

data:  tab.time
p-value = 4.146e-07
alternative hypothesis: two.sided
```

This effect becomes even more pronounced if we cut the tree slightly lower.

```
> table(cutree(sc, k = 3))

  1   2   3
222 124  45
```

```
> tab.time2 <- table(cutree(sc, k = 3), si$Time.point)
> tab.time2

     0 12 24 36
  1 45 56 70 51
  2 38 39 37 10
  3 26 13  6  0

> chisq.test(tab.time2)

        Pearson's Chi-squared test

data:  tab.time2
X-squared = 44.1381, df = 6, p-value = 6.94e-08
```

We also want to cluster the genes that differ between time. With both genes and samples clustered, we can construct a heatmap (**Figure 8**). The patterns in the heatmap suggest that there are at least eight different gene expression patterns, which are indicated by the colorbar along the left side.

```
> gc <- hclust(distanceMatrix(t(time.lapse), "pearson"), "ward")
> tcut <- cutree(gc, k = 8)
> tcut.colors <- brewer.pal(8, "Dark2")[tcut]
> tclass <- as.numeric(tsel)
> tclass[tsel] <- tcut
> table(tclass)

tclass
    0     1     2     3     4     5     6     7     8
31857    53   159   183   258   259    94   219   170

> stime <- t(scale(t(time.lapse)))
> stime[stime > 5] <- 5
> stime[stime < -5] <- -5
```

Actually, we only find two reasl classes: the ones that show an increase (top branch in **Figure 8**) and the ones that show a decrease (bottom branch in **Figure 8**) over time.

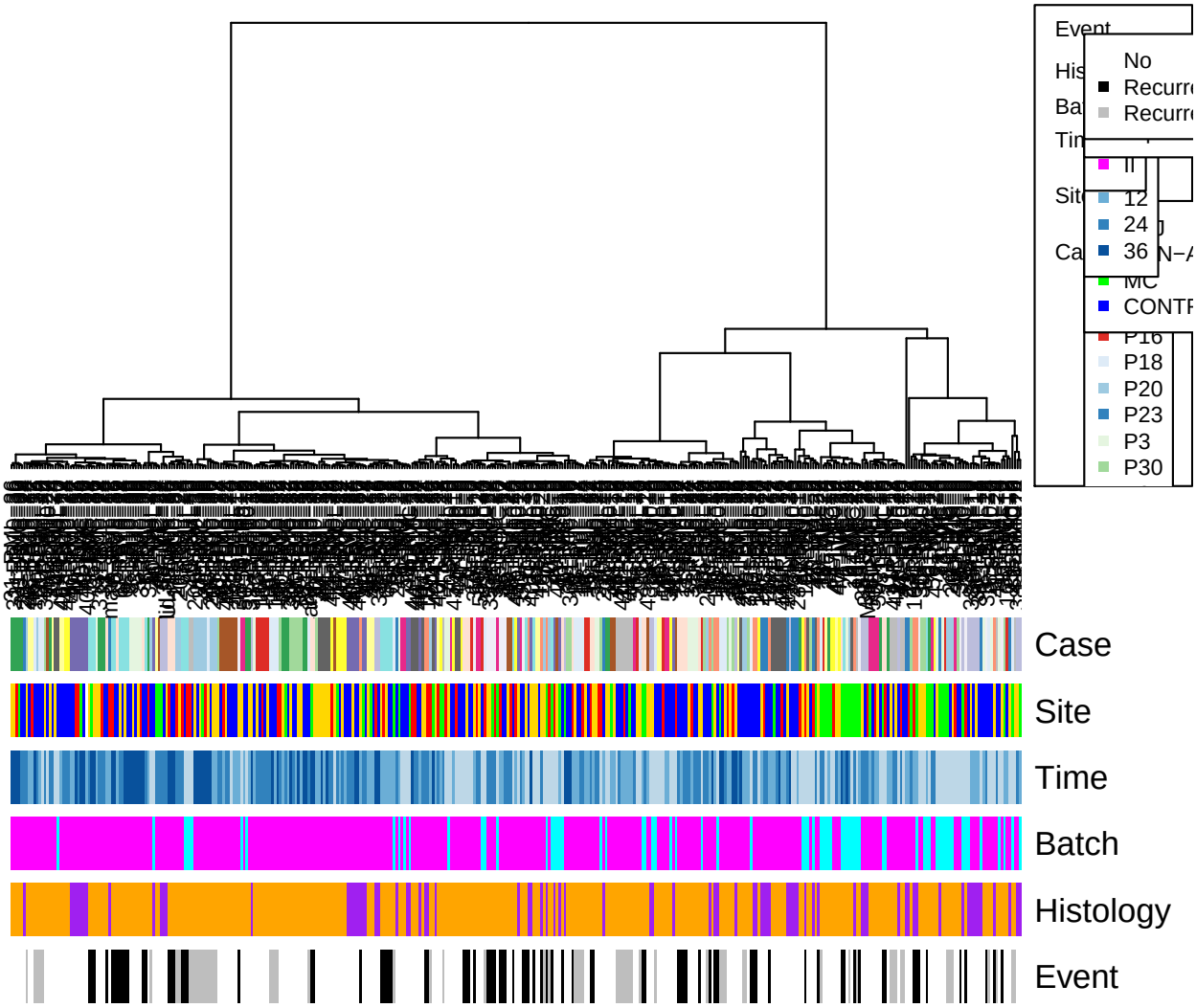

Figure 7: Hierarchical clustering of samples (using Pearson correlation and Ward's linkage) based on genes that are significantly different between **time points**.

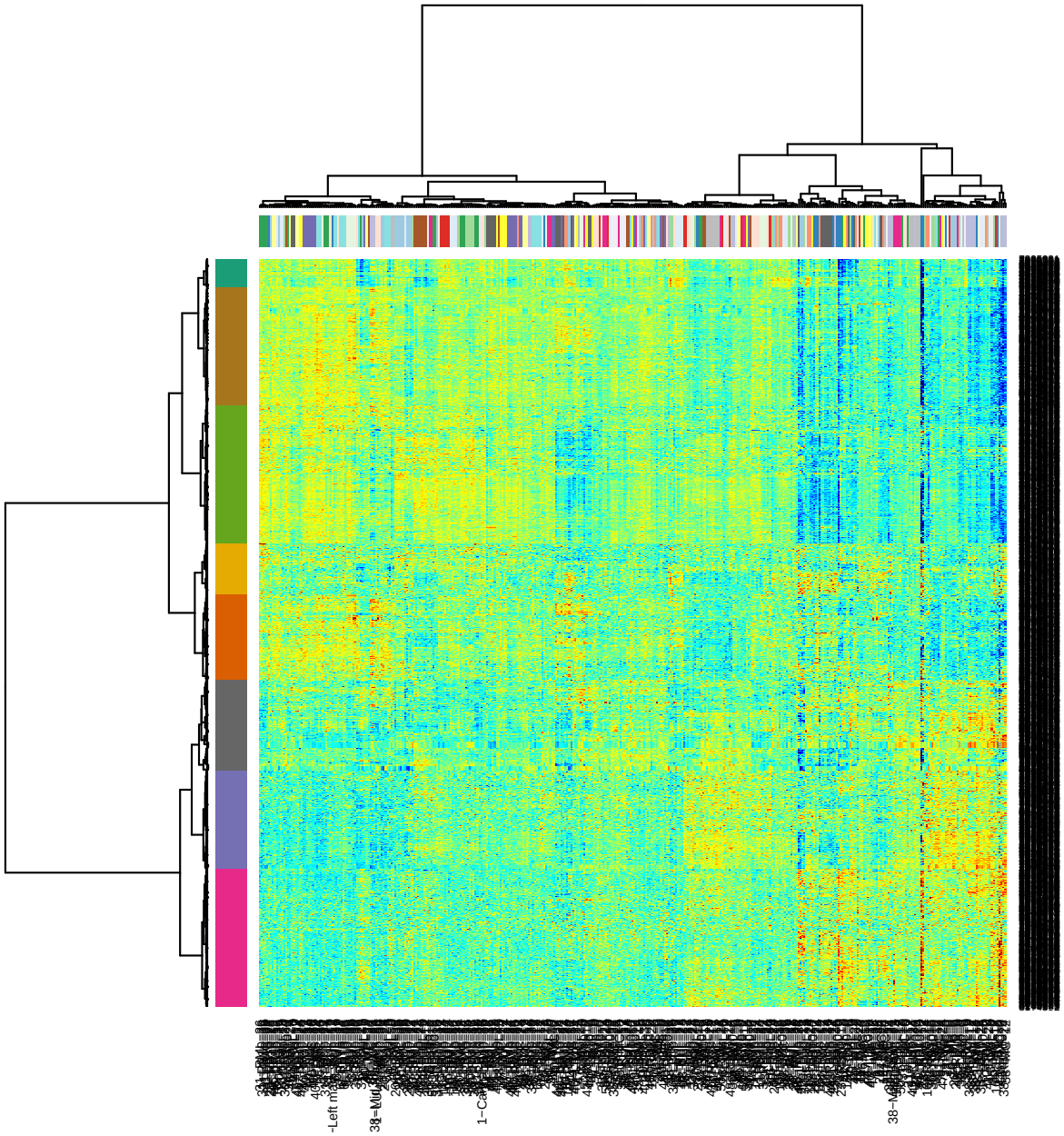

Figure 8: Two-dimensional clustering heatmap image of the genes selected because they differ by **time point**. Top colorbar indicates time, as in the previous plot. Left colorbar uses this clustering to define 8 types of gene expression patterns.

Appendix

This analysis was run in the following directory:

```
> getwd()

[1] "o:/Lung-HN/KRC-Analyses"
```

Note that '//mdadqsfs02/bioinfo2 is the standard insititutional location for storing data and analyses; 'O: is the name given to that location on this machine.

This analysis was run in the following software environment:

```
> sessionInfo()

R version 2.12.0 (2010-10-15)
Platform: x86_64-pc-mingw32/x64 (64-bit)

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] ClassDiscovery_2.10.2  mclust_3.4.8           cluster_1.13.1
 [4] ClassComparison_2.12.0 PreProcess_2.10.1      oompaBase_2.12.0
 [7] Biobase_2.10.0         RColorBrewer_1.0-2     lattice_0.19-13
[10] nlme_3.1-97

loaded via a namespace (and not attached):
[1] grid_2.12.0       KernSmooth_2.23-4
```