Microarray segmentation methods significantly influence data precision

Ahmed Ahmed James D. Brenton

February 12, 2004

1 Data description

Expression microarrays containing 6528 pairs of duplicate cDNA spots were used (Cancer Research UK DNA Microarray Facility at the Institute of Cancer Research; CR-UK DMF Human 6.5k genome-wide array). All microarrays used were from the same printing batch. Total RNA was obtained from the cell line HCT116 and an isogenic daughter line with a targeted disruption of the *EP300* gene derived by homologous recombination (Iyer *et al*, submitted). Total RNA was used for reverse transcription and indirect labeling with Cy3 and Cy5 dyes (Amersham) using random hexamers. Measurements of the amount of purified cDNA and Cy3/Cy5 incorporation were made before hybridization using the Nanodrop ND-1000 spectrophotometer (Nanodrop Technologies, Inc.). Two sets of experiments (A and B) were carried out, using 6 slides in each with a balanced dye-swap design (three slides for each dye). Experiment A and B were identical but used 10 µg and 15 µg total RNA for labelling for each hybridization. The names of the slides and the dyes used for each slide are as follows:

Experiment	Slide ID	p300 knock-out	HCT116
A	3084	Cy3	Cy5
	3089	Cy3	Cy5
	3097	Cy3	Cy5
	3092	Cy5	Cy3
	3093	Cy5	Cy3
	3069	Cy5	Cy3
B	3025	Cy3	Cy5
	3033	Cy3	Cy5
	3054	Cy3	Cy5
	3034	Cy5	Cy3
	3036	Cy5	Cy3
	3037	Cy5	Cy3

Segmentation was performed using QuantArray (Perkin Elmer) and GenePix Pro 4.1 (Axon Instruments, Inc.) software. All three methods of segmentation available within the QuantArray package were evaluated. When using the segmentation software, channel 1 was always assigned to cy3 regardless of the

sample, knock-out or wild type, used. The suffixes A, F, H and G were used, following the slide ID, to name the raw data files generated by the adaptive, fixed circle, histogram and genepix segmentation methods respectively.

All statistical analysis was conducted using the R environment [**?**] and the R package "Statistics for Microarray Analysis" [**?**]. We start the analysis by startting the sma library then loading the data files into R.

```
> library(sma)
> for (i in 1:length(dir(pattern = "....[AFHG]"))) {
+     data <- read.table(dir(pattern = "....[AFHG]")[i], sep = "\t",
+         header = TRUE)
+     assign(dir(pattern = "....[AFHG]")[i], data)
+ }
```

We generate objects according to the experiment used, A or B, and the segmentation method. For experiment A we build `expA.A`, `expA.F`, `expA.H`, `expA.G` for adaptive, fixedcircle, histogram and genepix respectively:

```
> expA.A <- c("3084A", "3089A", "3097A", "3092A", "3093A", "3069A")
> expA.F <- c("3084F", "3089F", "3097F", "3092F", "3093F", "3069F")
> expA.H <- c("3084H", "3089H", "3097H", "3092H", "3093H", "3069H")
> expA.G <- c("3084G", "3089G", "3097G", "3092G", "3093G", "3069G")
```

We do the same for experiment B:

```
> expB.A <- c("3025A", "3033A", "3054A", "3034A", "3036A", "3037A")
> expB.F <- c("3025F", "3033F", "3054F", "3034F", "3036F", "3037F")
> expB.H <- c("3025H", "3033H", "3054H", "3034H", "3036H", "3037H")
> expB.G <- c("3025G", "3033G", "3054G", "3034G", "3036G", "3037G")
```

Note that the first half of each of the vector of names represents slides where the test in the experiment, in this case the knock-out clone, was labeled by cy3 and the the control, in this case the wild type clone, labeled by cy5.

2 Statistical methods

Log intensity ratios for each spot were obtained, as a measure of the differential expression between the samples, with and without background subtraction. All spots from each microarray were included in the analysis. Data precision was initially assessed by using correlation coefficients.

2.1 The `get.cor` function

To allow for iterating over the entire data set and to investigate the effect of various normalization methods and background subtraction we use a function `get.cor`.

2.1.1 Description

This function extracts the log2 ratios for all the slides used as well as correlations between replicates within each slide and between different slides.

2.1.2 Usage

get.cor(x, BG.subtract=TRUE, normalization)

2.1.3 Arguments

x:	a vector the names of the raw data objects.
BG.subtract:	a logical argument meaning; 'should the back ground values be subtracted?'.
normalization:	the method of normalization to be used; 'n' for no normalization, 's' for scaled normalization, 'm' for median normalization, 'p' for print-tip group lowess normalization and 'l' for global 'lowess' normalization.

2.1.4 Details

In the function we build a list containing the raw data R, red=Channel 2, Rb, background for channel 2, G ,green= Channel 1, and Gb, background for channel 1. Note that, for consistency, we have always put cy3, G, in Channel 1 regardless of the sample used. Also note that by not specifying the background data, Rb and Gb, no background subtraction is performed.

We then specify the grid information for the images as follows; nspot.r= the number rows in a block, nspot.c= the number of columns within a block, ngrid.r= the number of rows of blocks and ngrid.c= the number of columns of blocks.

We use the normalization function in the sma package.

We then correct the sign for the ratios for the first half of the elements in (x) to account for the dye-swapping.

We assign the ratios of all the slides to an object called M. Note that the number of columns in M represents the number of elements in (x). In other words the number of slides tested.

We define index objects rep.a and rep.b for the first and second replicates in each slide.

To obtain the within-slide correlations we compute the correlation coefficients using the `cor` function in R.

For between-slidecorrelations we obtain the correlations for M using the the `cor` function.

2.1.5 Value

within:	a vector containing the within-slide correlations.
between:	a vector containing the between-slidecorrelations.

ratios: a data frame containing the log2 ratios of knock-out over wild-type for each spot in a slide. The number of columns correspond to the number of slides and the number of rows to the total number of spots in a slide.

2.1.6 Script

```
> get.cor <- function(x, BG.subtract = TRUE, normalization) {
+     correlations <- list(within = NULL, between = NULL, ratios = NULL)
+     res <- list(R = NULL, G = NULL, Rb = NULL, Gb = NULL)
+     if (BG.subtract == TRUE)
+         for (i in (1:length(x))) {
+             y <- get(x[i])
+             res$R <- cbind(res$R, y$ch2.Intensity)
+             res$G <- cbind(res$G, y$ch1.Intensity)
+             res$Rb <- cbind(res$Rb, y$ch2.Background)
+             res$Gb <- cbind(res$Gb, y$ch1.Background)
+         }
+     else for (i in (1:length(x))) {
+         y <- get(x[i])
+         res$R <- cbind(res$R, y$ch2.Intensity)
+         res$G <- cbind(res$G, y$ch1.Intensity)
+     }
+     data.grid <- list(nspot.r = 16, nspot.c = 34, ngrid.r = 12,
+         ngrid.c = 2)
+     data.ma <- stat.ma(res, data.grid, norm = normalization)
+     halfn = length(x)/2
+     data.ma$M[, c(1:halfn)] <- (data.ma$M[, c(1:halfn)]) * -1
+     M <- data.ma$M
+     rep.a <- rep(c(TRUE, FALSE), 12, each = 544)
+     rep.b <- rep(c(FALSE, TRUE), 12, each = 544)
+     cors <- NULL
+     for (i in 1:ncol(M)) {
+         y <- cor(M[rep.a, i], M[rep.b, i], use = "pairwise.complete.obs")
+         cors <- c(cors, y)
+     }
+     cors.between <- cor(M, use = "pairwise.complete.obs")
+     upper.diagonal <- col(cors.between) > row(cors.between)
+     cors.between <- cors.between[upper.diagonal]
+     correlations$within <- c(correlations$within, cors)
+     correlations$between <- c(correlations$between, cors.between)
+     correlations$ratios <- cbind(correlations$ratio, M)
+     correlations
+ }
```

2.2 Obtaining the data

We then obtain the data for the different methods of segmentation for experiment A:

First, with background subtraction and no normalization:

```
> dataA.A.BGS.n <- get.cor(expA.A, BG.subtract = TRUE, normalization = "n")
> dataA.F.BGS.n <- get.cor(expA.F, BG.subtract = TRUE, normalization = "n")
> dataA.H.BGS.n <- get.cor(expA.H, BG.subtract = TRUE, normalization = "n")
> dataB.A.BGS.n <- get.cor(expB.A, BG.subtract = TRUE, normalization = "n")
> dataB.F.BGS.n <- get.cor(expB.F, BG.subtract = TRUE, normalization = "n")
> dataB.H.BGS.n <- get.cor(expB.H, BG.subtract = TRUE, normalization = "n")
```

Then without background subtraction

```
> dataA.A.NoBGS.n <- get.cor(expA.A, BG.subtract = FALSE, normalization = "n")
> dataA.F.NoBGS.n <- get.cor(expA.F, BG.subtract = FALSE, normalization = "n")
> dataA.H.NoBGS.n <- get.cor(expA.H, BG.subtract = FALSE, normalization = "n")
> dataB.A.NoBGS.n <- get.cor(expB.A, BG.subtract = FALSE, normalization = "n")
> dataB.F.NoBGS.n <- get.cor(expB.F, BG.subtract = FALSE, normalization = "n")
> dataB.H.NoBGS.n <- get.cor(expB.H, BG.subtract = FALSE, normalization = "n")
```

Then we repeat all the process with scaled normalization:

```
> dataA.A.BGS.s <- get.cor(expA.A, BG.subtract = TRUE, normalization = "s")
> dataA.F.BGS.s <- get.cor(expA.F, BG.subtract = TRUE, normalization = "s")
> dataA.H.BGS.s <- get.cor(expA.H, BG.subtract = TRUE, normalization = "s")
> dataB.A.BGS.s <- get.cor(expB.A, BG.subtract = TRUE, normalization = "s")
> dataB.F.BGS.s <- get.cor(expB.F, BG.subtract = TRUE, normalization = "s")
> dataB.H.BGS.s <- get.cor(expB.H, BG.subtract = TRUE, normalization = "s")
> dataA.A.NoBGS.s <- get.cor(expA.A, BG.subtract = FALSE, normalization = "s")
> dataA.F.NoBGS.s <- get.cor(expA.F, BG.subtract = FALSE, normalization = "s")
> dataA.H.NoBGS.s <- get.cor(expA.H, BG.subtract = FALSE, normalization = "s")
> dataB.A.NoBGS.s <- get.cor(expB.A, BG.subtract = FALSE, normalization = "s")
> dataB.F.NoBGS.s <- get.cor(expB.F, BG.subtract = FALSE, normalization = "s")
> dataB.H.NoBGS.s <- get.cor(expB.H, BG.subtract = FALSE, normalization = "s")
```

2.3 Extracting the correlations to data frames

We then build a data frame of the results of within-slide correlations and the identifiers as follows; the first column contains the correlation values, the second contains the segmentation method and the third contains the DNA content whether Low, in experiment A, or High, in experiment B:

```
> Results.within.BGS <- data.frame(c(dataA.A.BGS.n$within, dataA.F.BGS.n$within,
+     dataA.H.BGS.n$within, dataB.A.BGS.n$within, dataB.F.BGS.n$within,
+     dataB.H.BGS.n$within), factor(rep(c("A", "F", "H"), each = 6,
+     2)), factor(rep(c("Low", "High"), each = 18)))
> names(Results.within.BGS) <- c("Correlations", "Method", "DNA")
```

And with no background subtraction:

```
> Results.within.NoBGS <- data.frame(c(dataA.A.NoBGS.n$within,
+     dataA.F.NoBGS.n$within, dataA.H.NoBGS.n$within, dataB.A.NoBGS.n$within,
+     dataB.F.NoBGS.n$within, dataB.H.NoBGS.n$within), factor(rep(c("A",
+     "F", "H"), each = 6, 2)), factor(rep(c("Low", "High"), each = 18)))
> names(Results.within.NoBGS) <- c("Correlations", "Method", "DNA")
```

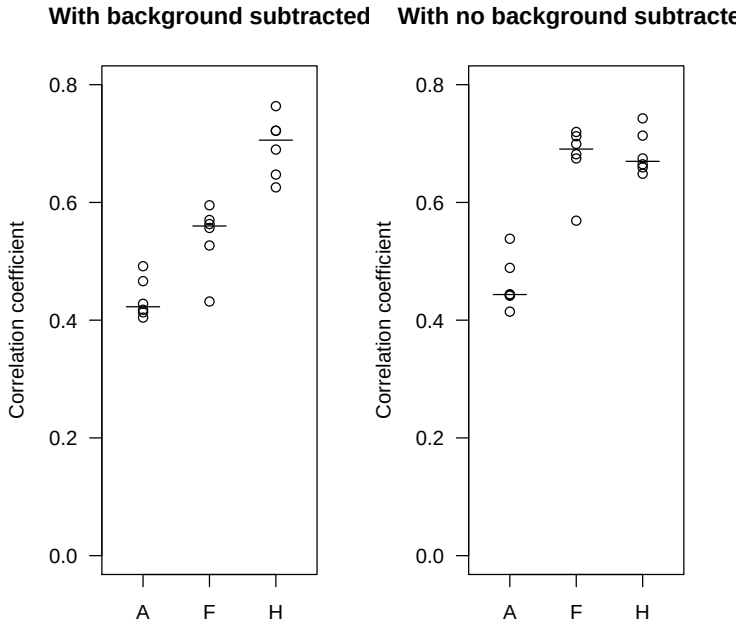
We do the same for between-slidecorrelations and we add a fourth column to identify whether the dyes used for the two slide where the same, S, or different, D. Note that for between-slidecomparisons we used normalized data:

```
> Results.between.BGS <- data.frame(c(dataA.A.BGS.s$between, dataA.F.BGS.s$between,
+     dataA.H.BGS.s$between, dataB.A.BGS.s$between, dataB.F.BGS.s$between,
+     dataB.H.BGS.s$between), factor(rep(c("A", "F", "H"), each = 15,
+     2)), factor(rep(c("Low", "High"), each = 45)), factor(rep(c("S",
+     "D", "S", "D", "S"), c(3, 6, 1, 3, 2))))
> names(Results.between.BGS) <- c("Correlations", "Method", "DNA",
+     "Dye")
> Results.between.NoBGS <- data.frame(c(dataA.A.NoBGS.s$between,
+     dataA.F.NoBGS.s$between, dataA.H.NoBGS.s$between, dataB.A.NoBGS.s$between,
+     dataB.F.NoBGS.s$between, dataB.H.NoBGS.s$between), factor(rep(c("A",
+     "F", "H"), each = 15, 2)), factor(rep(c("Low", "High"), each = 45)),
+     factor(rep(c("S", "D", "S", "D", "S"), c(3, 6, 1, 3, 2))))
> names(Results.between.NoBGS) <- c("Correlations", "Method", "DNA",
+     "Dye")
```

3 Segmentation method significantly influences within-slide correlations

To investigate whether the segmentation method was an important determinant of within-slide correlations, we first plot the within-slide correlations data categorized by the segmentation method. We use a function to plot the medians of the different categories on the dot plots as follows.

```
> stripchart.plot.median.by.factor <- function(x, y, z) {
+     for (i in 1:length(x)) {
+         lines(c(x[i] - z, x[i] + z), c(y[i], y[i]))
+     }
+ }
```



We then turn the graphics window off.

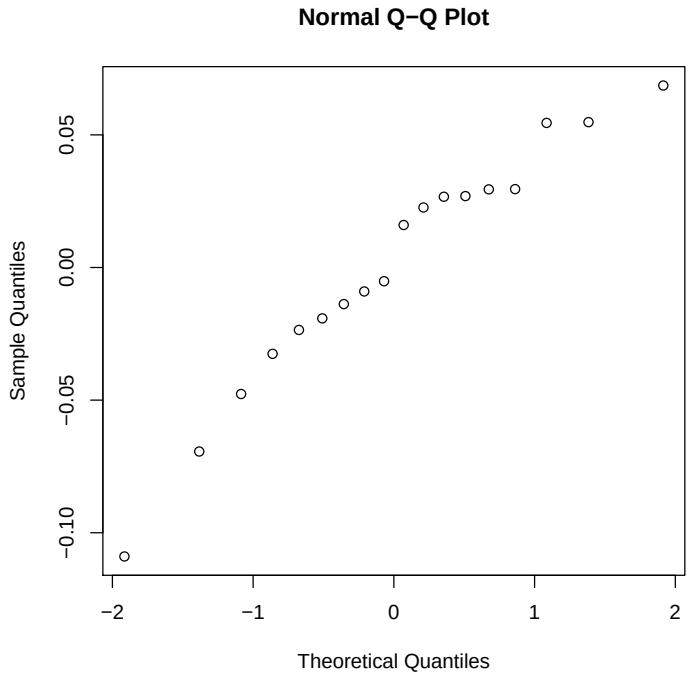
```
> dev.off()

null device
          1
```

The dot plot shows differences in the correlations between the three methods of segmentation, A, F and H. To investigate whether these differences were significant we used a one-way ANOVA which is suitable for investigating data grouped into more than two categories. Applying a simple t test would not account for the problem of multiple testing of differences between the means of categories (A−F, A−H and F−H). First, one-way ANOVA analysis was performed on the background subtracted within-slide correlations for experiment A (first 18 rows):

```
> ANOVA.within.one.BGS <- lm(Correlations[1:18] ~ Method[1:18],
+     Results.within.BGS)
> summary(ANOVA.within.one.BGS)

Call:
lm(formula = Correlations[1:18] ~ Method[1:18], data = Results.within.BGS)

Residuals:
      Min        1Q    Median        3Q       Max
-0.108935 -0.022447  0.005427  0.028829  0.068617
```

```
Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.43690    0.01996  21.886 8.52e-13 ***
Method[1:18]F  0.10376    0.02823   3.675  0.00225 **
Method[1:18]H  0.25810    0.02823   9.142 1.60e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.0489 on 15 degrees of freedom
Multiple R-Squared: 0.8495,        Adjusted R-squared: 0.8294
F-statistic: 42.33 on 2 and 15 DF,  p-value: 6.791e-07
```

We found a significant difference ($p < 0.001$) between the three methods of segmentation. Note that the R-squared test equals 0.84 indicating a good fit. We also perform a diagnostic QQ plot to examine the fitness of the model to the data. We find a reasonable fit of the quantiles.



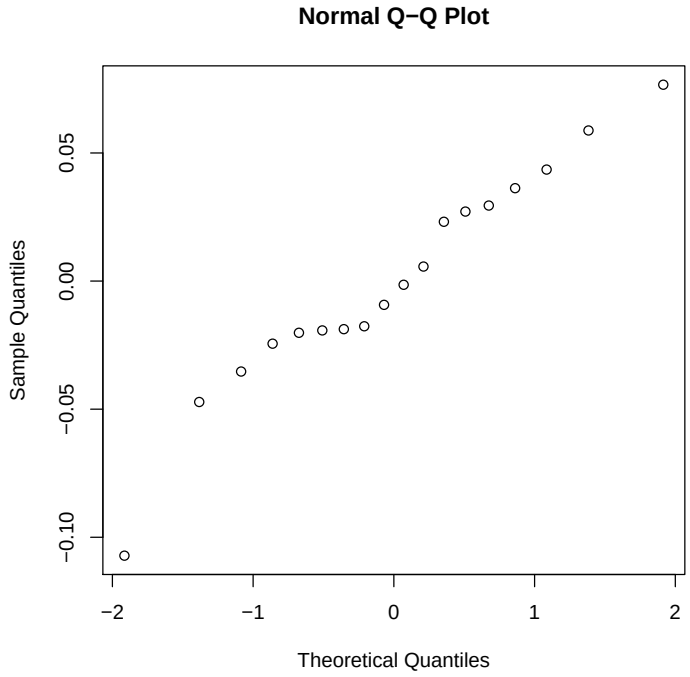
We then use the Tukey HSD test to investigate the significance level at each level of comparison.

```
> TukeyHSD(aov(Correlations[1:18] ~ Method[1:18], Results.within.BGS))

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Correlations[1:18] ~ Method[1:18], data = Results.within.BGS)
```

```
$"Method[1:18]"
         diff        lwr       upr
F-A 0.1037579 0.03042833 0.1770874
H-A 0.2581018 0.18477229 0.3314314
H-F 0.1543440 0.08101442 0.2276735
```

We found that the confidence intervals at each level of comparison do not include zero indicating that the Null hypothesis of equal category means is rejected. We then do the same analysis on the data with no background subtraction and do the diagnostic plot. We find a reasonable fit for the quantiles.

```
> ANOVA.within.one.NoBGS <- lm(Correlations[1:18] ~ Method[1:18],
+     Results.within.NoBGS)
> summary(ANOVA.within.one.NoBGS)

Call:
lm(formula = Correlations[1:18] ~ Method[1:18], data = Results.within.NoBGS)

Residuals:
      Min        1Q    Median        3Q       Max
-0.107173 -0.019949 -0.005356  0.028892  0.076647

Coefficients:
              Estimate Std. Error t value Pr(>|t|)
(Intercept)    0.46172    0.01880  24.564 1.58e-13 ***
Method[1:18]F  0.21448    0.02658   8.068 7.76e-07 ***
Method[1:18]H  0.22234    0.02658   8.364 4.96e-07 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.04604 on 15 degrees of freedom
Multiple R-Squared: 0.8573,        Adjusted R-squared: 0.8382
F-statistic: 45.05 on 2 and 15 DF,  p-value: 4.560e-07
```




```
> TukeyHSD(aov(Correlations[1:18] ~ Method[1:18], Results.within.NoBGS))

  Tukey multiple comparisons of means
    95% family-wise confidence level

Fit: aov(formula = Correlations[1:18] ~ Method[1:18], data = Results.within.NoBGS)

$"Method[1:18]"
           diff         lwr        upr
F-A 0.214476387  0.14542797 0.28352481
H-A 0.222336495  0.15328807 0.29138492
H-F 0.007860108 -0.06118831 0.07690853
```

The overall difference between the methods remained ($p < 0.001$) although the difference between the histogram and fixed circle method was no longer significant.

4 Histogram segmentation gives lower pixel-to-pixel variability

We hypothesized that the better precision for the histogram method was because of less variability in pixel intensity, as fluctuations in pixel values have been shown to increase noise [**?**]. The histogram method summarizes centiles of pixel intensities obtained from a square centered around the true spot. From this it follows that a narrow window of centiles will reduce the within-spot variability

as compared to the other methods of segmentation used here. We therefore calculated the coefficient of variability (CV) for foreground and background pixels for each feature in experiments A and B in both Cy3 and Cy5 channels.

4.1 Categorizing the data

In order to obtain the CV values we created a data object containing all the names of the raw data objects created by the three different methods of quantarray segmentation:

```
> All.slides.methods <- c(expA.A, expA.F, expA.H, expB.A, expB.F,
+     expB.H)
```

We build the identifier for the interaction between the method of segmentation and the DNA content:

```
> Method.content <- factor(rep(c("LowA", "LowF", "LowH", "HighA",
+     "HighF", "HighH"), each = 6))
```

4.2 The `get.cv` function

4.2.1 Description

This is a function to compute the spot CV values.

4.2.2 Usage

get.cv(x)

4.2.3 Arguments

x:	an object that has the names of the raw data objects.

4.2.4 Details

We first get the median coefficient of variability (CV) values for each slide. First the median CV for cy3, channel 1, by dividing the standard deviation (Std.Dev) over the mean intensity derived from the total number of pixels per spot. We then get the median of the CVs for all the spots on the slide. We do the same for the background and for cy5.

4.2.5 Value

A list containing the following:

Cy3.foreground:	the medians for all the CVs of cy3 foreground of all spots in a slide.
Cy3.background:	the medians for all the CVs of cy3 background of all spots in a slide.

Cy5.foreground: the medians for all the CVs of cy5 foreground of all spots in a slide.

Cy5.background: the medians for all the CVs of cy5 background of all spots in a slide.

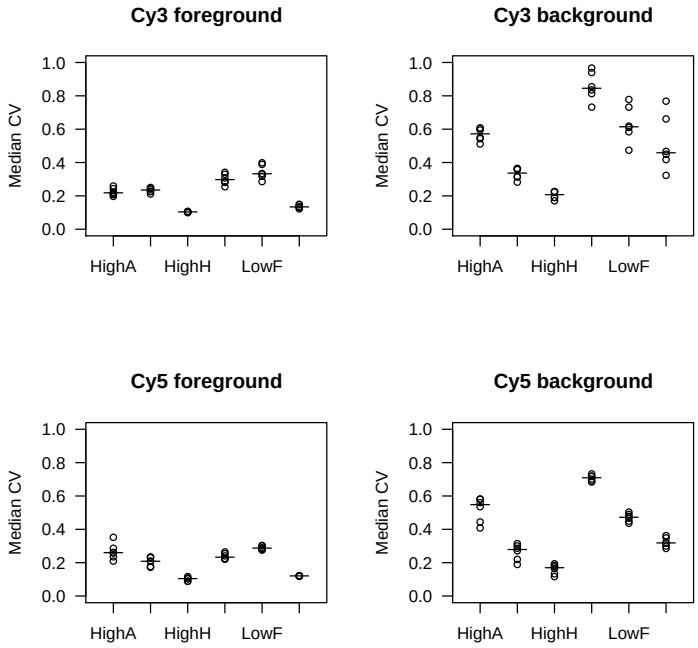
4.2.6 Script

```
> get.cv <- function(x) {
+     Median.CV.Values <- list(Cy3.foreground = NULL, Cy3.background = NULL,
+         Cy5.foreground = NULL, Cy5.Background = NULL)
+     for (i in 1:length(x)) {
+         data <- get(x[i])
+         CV.cy3.fore <- data$ch1.Intensity.Std.Dev/data$ch1.Intensity
+         median.cvs.cy3.fore <- median(CV.cy3.fore, na.rm = TRUE)
+         CV.cy3.back <- data$ch1.Background.Std.Dev/data$ch1.Background
+         median.cvs.cy3.back <- median(CV.cy3.back, na.rm = TRUE)
+         CV.cy5.fore <- data$ch2.Intensity.Std.Dev/data$ch2.Intensity
+         median.cvs.cy5.fore <- median(CV.cy5.fore, na.rm = TRUE)
+         CV.cy5.back <- data$ch2.Background.Std.Dev/data$ch2.Background
+         median.cvs.cy5.back <- median(CV.cy5.back, na.rm = TRUE)
+         Median.CV.Values$Cy3.foreground <- c(Median.CV.Values$Cy3.foreground,
+             median.cvs.cy3.fore)
+         Median.CV.Values$Cy3.background <- c(Median.CV.Values$Cy3.background,
+             median.cvs.cy3.back)
+         Median.CV.Values$Cy5.foreground <- c(Median.CV.Values$Cy5.foreground,
+             median.cvs.cy5.fore)
+         Median.CV.Values$Cy5.background <- c(Median.CV.Values$Cy5.background,
+             median.cvs.cy5.back)
+         Median.CV.Values
+     }
+ }
```

5 Results of analysis of CV values

We get the median CV values for all the data set and plot the results as follows:

```
> Median.CVs <- get.cv(All.slides.methods)
> attach(Median.CVs)
```




```
> detach(Median.CVs)
```

The histogram method had the lowest CV values in both foreground and background.

6 Dye-swapping confounds the precision of between-slide comparisons

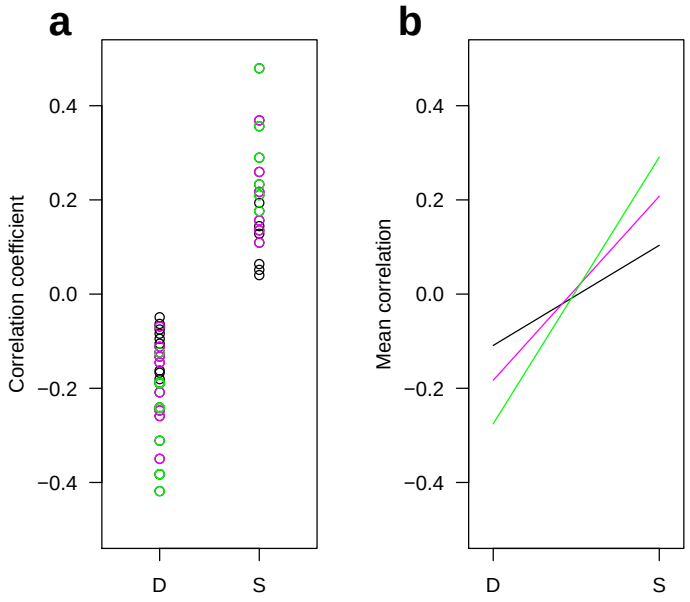
We next studied the effect of segmentation on between-slide variability by deriving a matrix of correlations for all possible pair-wise comparisons between the slides for each method (fifteen comparisons for each of three segmentation methods). A one-way ANOVA was conducted as above.

```
> ANOVA.between.one.BGS <- lm(Correlations[1:45] ~ Method[1:45],
+     Results.between.BGS)
> summary(ANOVA.between.one.BGS)

Call:
lm(formula = Correlations[1:45] ~ Method[1:45], data = Results.between.BGS)

Residuals:
     Min       1Q   Median       3Q      Max
-0.36937 -0.14291 -0.06007  0.16832  0.52833

Coefficients:
```

```
                Estimate Std. Error t value Pr(>|t|)
(Intercept)    -0.023981   0.058585  -0.409    0.684
Method[1:45]F -0.002681   0.082852  -0.032    0.974
Method[1:45]H -0.024950   0.082852  -0.301    0.765

Residual standard error: 0.2269 on 42 degrees of freedom
Multiple R-Squared: 0.002596,        Adjusted R-squared: -0.0449
F-statistic: 0.05466 on 2 and 42 DF,  p-value: 0.9469
```

In contrast to our results from within-slide comparisons, no significant differences in the correlations were found. The correlation coefficients between slides with dye swapping were mostly negative, indicating low overall repeatability of the data. As dye swapping would be expected to alter correlations between slides, we reanalyzed the data by restricting the comparisons to those between replicates in which cDNA probes had been labelled with the same fluors. We plot the effect of dye swapping on correlations using the interaction plot:

```
> low <- Results.between.BGS[1:45, ]
```




```
> dev.off()

null device
          1
```

As expected, comparisons between replicates with the same dyes had higher correlations than between slides with swapped dyes. However, the beneficial effect of histogram segmentation on correlation was observed for slides with the same dye.

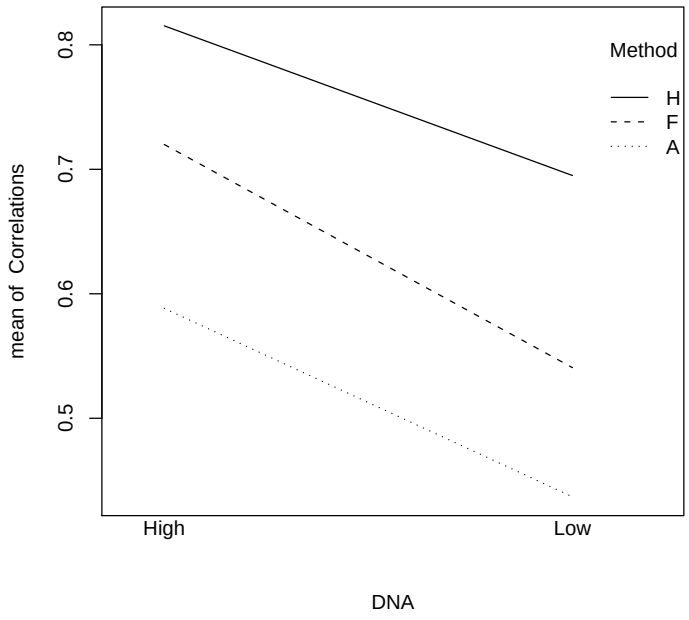
7 Precision is determined by choice of segmentation method and amount of labelled probe

7.1 The effect of segmentation method and amount of labelled probe on within-slide correlations

The impact of the quantity of RNA used on the overall precision of microarray data has been previously reported [**?**]. In experiment *A*, we labelled 10 µg of total RNA for each slide which yielded a median of 2.1 µg (IQR 1.1–3.1) of cDNA probe after purification, and incorporated a median of 151 pmol (IQR 104–206) of each fluor. In order to identify whether the low between-slide correlations were caused by inadequate specific activity of our samples, we repeated the experiment using starting material of 15 µg of total RNA for each sample (experiment *B*; median labelled cDNA 5.5 µg, IQR 4.6–6.6, median incorporation of each fluor 463 pmol, IQR 384–534).

First, we look at the effect of the two variables on the within-slide correlations using an interaction plot.

```
> attach(Results.within.BGS)
```




```
> detach(Results.within.BGS)
```

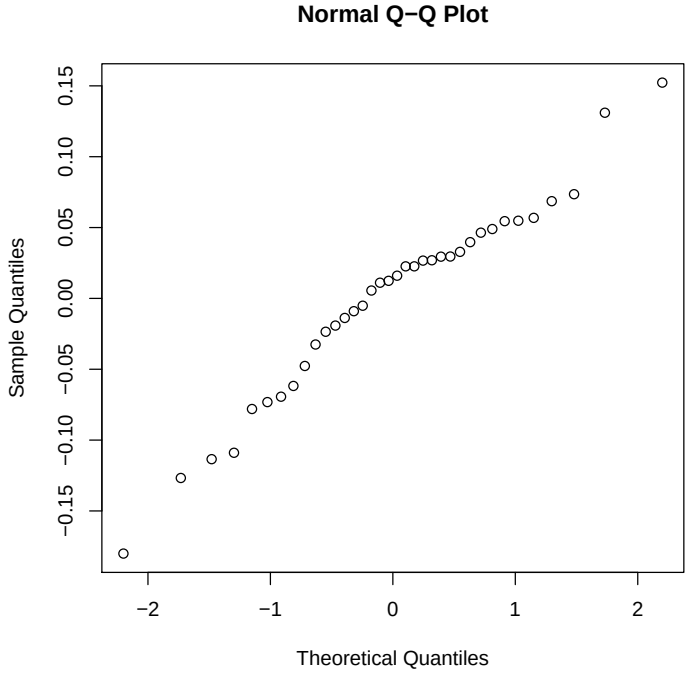
The figure indicates a lack of interaction between the two variables and suggests that differences exist between the categories within each variable, low and high for the DNA content and A, F, H for the segmentation method.

In order to investigate whether these differences were significant, we performed a two-way ANOVA for within-slide correlations to examine the effect of

DNA content and the method of segmentation. Note that we fit the interaction term in the model to examine for interactions.

```
> ANOVA.within.two.BGS <- lm(Correlations ~ Method * DNA, Results.within.BGS)
> anova(ANOVA.within.two.BGS)

Analysis of Variance Table

Response: Correlations
           Df  Sum Sq Mean Sq F value    Pr(>F)
Method      2 0.35305 0.17652 31.2396 4.636e-08 ***
DNA         1 0.20333 0.20333 35.9831 1.400e-06 ***
Method:DNA  2 0.00523 0.00262  0.4629    0.6339
Residuals  30 0.16952 0.00565
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

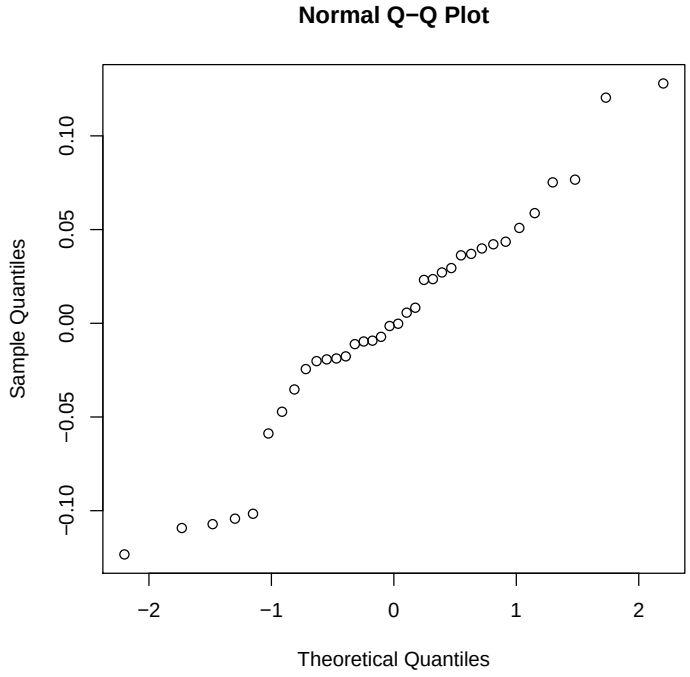
The amount of labelled sample and method of segmentation independently and significantly ($p < 0.001$) influenced correlations. There was no significant interaction between the two variables. We then perform a diagnostic QQ plot to examine how the models fits the data and find a good fit.



We repeat the same process with no background subtraction

```
> ANOVA.within.two.NoBGS <- lm(Correlations ~ Method * DNA, Results.within.NoBGS)
> anova(ANOVA.within.two.NoBGS)
```

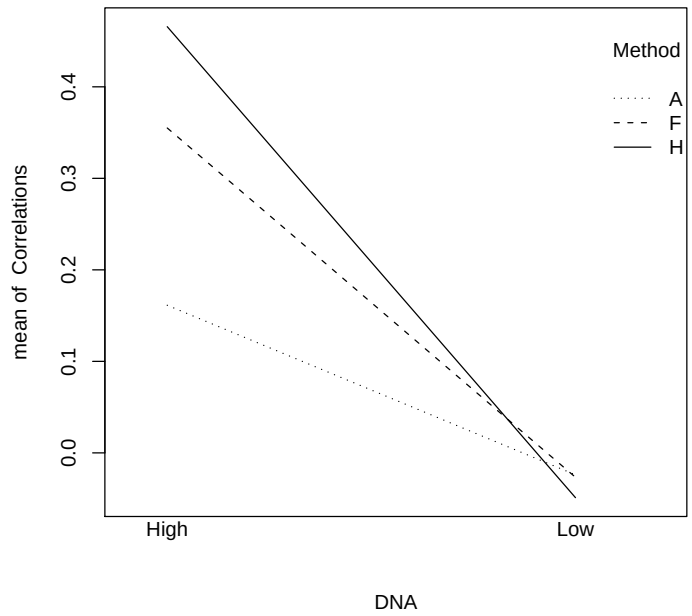
```
Analysis of Variance Table

Response: Correlations
           Df   Sum Sq  Mean Sq F value    Pr(>F)
Method      2 0.303638 0.151819 35.4856 1.241e-08 ***
DNA         1 0.244684 0.244684 57.1916 1.969e-08 ***
Method:DNA  2 0.007234 0.003617  0.8454    0.4394
Residuals  30 0.128350 0.004278
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



7.2 The effect of segmentation method and amount of labelled probe on between-slide correlations

We then plot the effect of the two variables on between-slide correlations

```
> attach(Results.between.BGS)
```

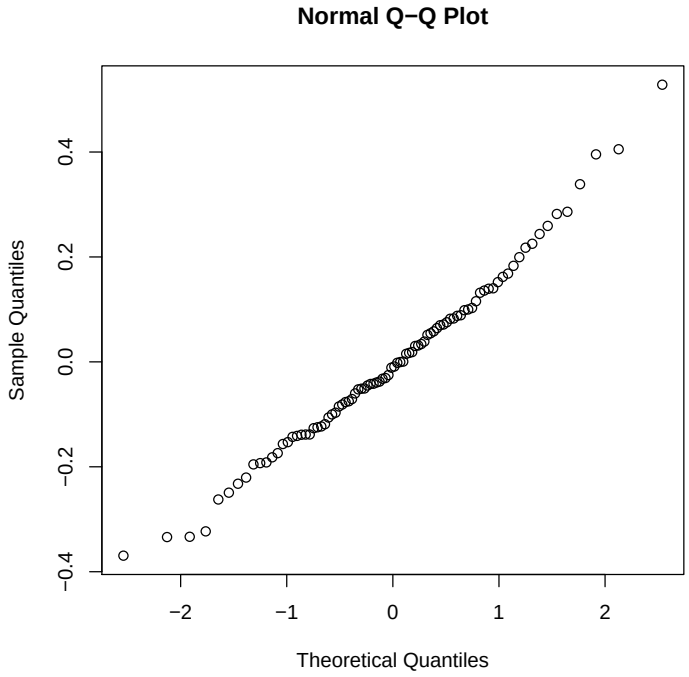



```
> detach(Results.between.BGS)
> dev.off()

null device
          1
```

The plot shows obvious interaction between the two variables at the low DNA concentration level. In order to formally assess the apparent differences between categories within each variable we perform a two-way ANOVA.

```
> ANOVA.between.two.BGS <- lm(Correlations ~ Method * DNA, Results.between.BGS)
> anova(ANOVA.between.two.BGS)

Analysis of Variance Table

Response: Correlations
           Df  Sum Sq Mean Sq F value    Pr(>F)
Method      2 0.30598 0.15299  4.9495  0.009289 **
DNA         1 2.92465 2.92465 94.6172 2.047e-15 ***
Method:DNA  2 0.41177 0.20588  6.6607  0.002065 **
Residuals  84 2.59647 0.03091
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```

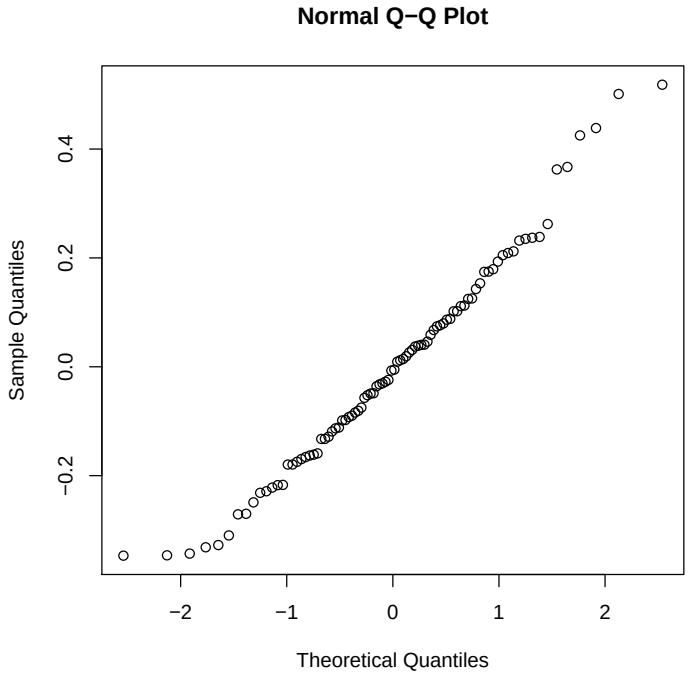
Again we perform a diagnostic QQ plot and find a good quantile fit.



For between-slide comparisons, using a larger amount of labelled sample significantly ($p < 0.001$) improved the correlations independently of the segmentation method used. However significant interaction was observed between the amount of labelled sample and the segmentation method. Therefore, while there was no advantage for any segmentation method when low amounts of labelled sample were used, there were marked differences for the methods when using higher amounts.

We examine the effects with no background subtraction as follows:

```
> ANOVA.between.two.NoBGS <- lm(Correlations ~ Method * DNA, Results.between.NoBGS)
> anova(ANOVA.between.two.NoBGS)

Analysis of Variance Table

Response: Correlations
           Df Sum Sq Mean Sq F value    Pr(>F)
Method      2 0.3305  0.1652  4.0781  0.020402 *
DNA         1 3.7414  3.7414 92.3384 3.529e-15 ***
Method:DNA  2 0.4459  0.2230  5.5030  0.005678 **
Residuals  84 3.4035  0.0405
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1
```



8 Coefficient of repeatability confirms higher precision

A low value for the correlation coefficient does not necessarily mean low repeatability as the correlation coefficient is not a measure of sameness [**?**]. Previous reports have shown discrepancies between correlation coefficients and repeatability coefficients [**?**, **?**]. In order to confirm our findings, we repeated the analysis using the coefficient of repeatability (CR) values to compare between the three different methods of segmentation and included a fourth proprietary method encoded within the GenePix software package.

8.1 Data processing for the Genepix segmentation

We focus the attention on the data from experiment B to avoid the effect of DNA content. We only examine the data with no background subtraction to examine the effect independent of background estimation.

8.2 The `Get.ratios.genepix` function

8.2.1 Description

A function to automate the process of obtaining the normalized log2 ratios for the genepix method.

8.2.2 Usage

Get.ratios.genepix(x)

8.2.3 Arguments

x:	an object containing the names of the genepix raw data objects.

8.2.4 Details

For each raw data object we get the intensity values for cy5, Red= F635.Mean, and for cy3, Green= F532.Mean. We apply the grid information. We use scaled normalization as above.

8.2.5 Value

A data frame containing the normalized log2 ratios of each spot. The number of columns corresponds to the number of elements in x.

8.2.6 Script

```
> Get.ratios.genepix <- function(x) {
+     res <- list(R = NULL, G = NULL, Rb = NULL, Gb = NULL)
+     for (i in (1:length(x))) {
+         y <- get(x[i])
+         res$R <- cbind(res$R, y$F635.Mean)
+         res$G <- cbind(res$G, y$F532.Mean)
+         res
+     }
+     data.grid <- list(nspot.r = 16, nspot.c = 34, ngrid.r = 12,
+         ngrid.c = 2)
+     data.ma <- stat.ma(res, data.grid, norm = "s")
+     halfn = length(x)/2
+     data.ma$M[, c(1:halfn)] <- (data.ma$M[, c(1:halfn)]) * -1
+     data.ma$M
+ }
```

8.2.7 Obtaining the ratios

```
> Res.genepix <- Get.ratios.genepix(expB.G)
```

8.3 The get.sigma function

8.3.1 Description

A function that calculates the coeffecient of repeatability (CR) values as described by Jenssen2002.

8.3.2 Usage

get.sigma(x,arrays)

8.3.3 Arguments

x:	a three-dimensional array having 2 columns, by 6528 rows, by the number of slides in an experiment as a third dimension.
arrays:	the number of slides in an experiment

8.3.4 Details

The function computes the CR value for each spot by getting the square root of the mean of sum of squares for replicates across the slides. First the mean of the ratios of any two replicates on a slide:

```
> spot.mean <- function(x) {
+     y <- mean(x, na.rm = T)
+ }
```

Where x is a vector containing two elements representing the log2 ratios of a duplicate spot on a slide.

Then the sum of squares for a duplicate spot pair on a slide:

```
> sum.sq.spot <- function(x) {
+     square.difference.spot <- NULL
+     for (i in 1:length(x)) {
+         y <- x[i] - (spot.mean(x))
+         y <- y^2
+         square.difference.spot <- c(square.difference.spot, y)
+     }
+     sum.square.difference.spot <- sum(square.difference.spot)
+     sum.square.difference.spot
+ }
```

where x is a vector containing two elements representing the log2 ratios of a duplicate spot on a slide.

First we get the difference between the ratio of each spot and the mean of the two spot ratios.

Then we get the sum of the differences

We repeat the same process for each duplicate on a slide:

```
> sum.sq.allspots <- function(x) {
+     ss.allspots <- NULL
+     for (i in 1:nrow(x)) {
+         y <- x[i, ]
+         y <- sum.sq.spot(y)
+         ss.allspots <- c(ss.allspots, y)
+         ss.allspots
+     }
+ }
```

Where x is a matrix of two columns, representing the two replicates in a slide, and 6528 rows, representing the different clones on the slide.

Finally, we obtain the CR values by computing the square root of the mean of sum of squares for replicates across the slides.

8.3.5 Value

An object containing the CR values for each spot in the object x.

8.3.6 Script

```
> get.sigma <- function(x, arrays) {
+     CR.values <- NULL
+     for (i in 1:arrays) {
+         y <- x[, , i]
+         y <- sum.sq.allspots(y)
+         CR.values <- cbind(CR.values, y)
+         CR.values
+     }
+     res <- apply(CR.values, 1, mean, na.rm = T)
+     res <- sqrt(res)
+     res
+ }
```

8.4 Obtaining the CR values

We convert the data to a three dimensional matrix for the different segmentation method.

8.4.1 The adaptive method

```
> rep.a <- rep(c(TRUE, FALSE), 12, each = 544)
> rep.b <- rep(c(FALSE, TRUE), 12, each = 544)
> Ratio.adaptive <- NULL
> for (i in 1:6) {
+     y <- cbind(dataB.A.NoBGS.s$ratios[rep.a, i], dataB.A.NoBGS.s$ratios[rep.b,
+         i])
+     Ratio.adaptive <- cbind(Ratio.adaptive, y)
+ }
> dim(Ratio.adaptive) <- c(6528, 2, 6)
```

8.4.2 The fixed circle method

```
> Ratio.fixed <- NULL
> for (i in 1:6) {
+     y <- cbind(dataB.F.NoBGS.s$ratios[rep.a, i], dataB.F.NoBGS.s$ratios[rep.b,
+         i])
+     Ratio.fixed <- cbind(Ratio.fixed, y)
+ }
> dim(Ratio.fixed) <- c(6528, 2, 6)
```

8.4.3 The histogram method

```
> Ratio.histogram <- NULL
> for (i in 1:6) {
+     y <- cbind(dataB.H.NoBGS.s$ratios[rep.a, i], dataB.H.NoBGS.s$ratios[rep.b,
+         i])
+     Ratio.histogram <- cbind(Ratio.histogram, y)
+ }
> dim(Ratio.histogram) <- c(6528, 2, 6)
```

8.4.4 The genepix method

```
> Ratio.genepix <- NULL
> for (i in 1:6) {
+     y <- cbind(Res.genepix[rep.a, i], Res.genepix[rep.b, i])
+     Ratio.genepix <- cbind(Ratio.genepix, y)
+ }
> dim(Ratio.genepix) <- c(6528, 2, 6)
```

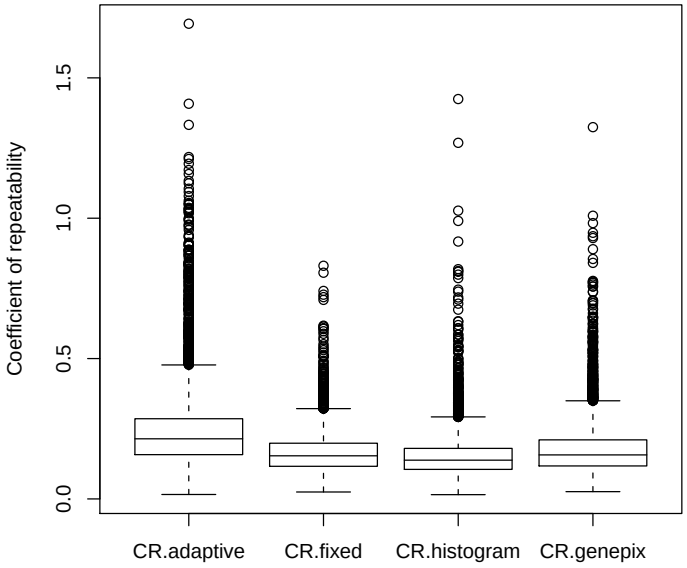
8.4.5 We then obtain the CR values for each method of segmentation.

```
> CR.adaptive <- get.sigma(Ratio.adaptive, 6)
> CR.fixed <- get.sigma(Ratio.fixed, 6)
> CR.histogram <- get.sigma(Ratio.histogram, 6)
> CR.genepix <- get.sigma(Ratio.genepix, 6)
```

8.4.6 We then calculate the median values for each method and plot the data.

```
> apply(cbind(CR.adaptive, CR.fixed, CR.histogram, CR.genepix),
+     2, median, na.rm = T)
```

```
 CR.adaptive     CR.fixed CR.histogram   CR.genepix
   0.2141851    0.1536131    0.1380435    0.1568353
```



The box plots for the sigma factors obtained for each feature from the slides from experiment B (no background subtraction) showed that the histogram methods had the lowest median CR value.

9 Estimating the differentially expressed genes

The probability of a gene to be differentially expressed is dependent on the variability of the data for that gene across the replicates of an experiment [**?**]. It follows from our findings that the segmentation method could have a direct effect on the number of differentially expressed genes identified. In order to test this assumption we used a Bayesian method to estimate the number of differentially expressed genes at a p value of 0.01 from the data set of experiment B [**?**].

We take account of each the duplicate clones on a slide in estimating significance. To achieve this we reorder the clones so that duplicates are sequentially in pairs as this is required by the stat.bayes function in sma.

9.1 The order.clones function

9.1.1 Description

A function to reorder the clones within a slide so that duplicates are listed sequentially.

9.1.2 Usage

order.clones(x)

9.1.3 Arguments

x: a matrix of 13056 rows corresponding to all the clones on a slide. The number of columns corresponds to the number of slides in an experiment.The data in x are the log2 ratios.

y: the number of clones on each subgrid of a microarray.

9.1.4 Details

We exclude any clones with a replicate that has a missing value in the experiment and reorder the matrix so that duplicate clones in a slide are sequential:

9.1.5 Script

```
> order.clones <- function(x, y) {
+     rep.a <- rep(c(TRUE, FALSE), each = y)
+     rep.b <- rep(c(FALSE, TRUE), each = y)
+     replicates <- cbind(x[rep.a, ], x[rep.b, ])
+     replicates <- na.omit(replicates)
+     reordered.clones <- NULL
+     for (i in 1:ncol(x)) {
+         y <- c(rbind(replicates[, i], replicates[, i + ncol(x)]))
+         reordered.clones <- cbind(reordered.clones, y)
+         reordered.clones
+     }
+ }
```

9.2 Identification of differential expression

We then compute the probabilities for differential expression:

```
> adaptive.bayesian <- stat.bayesian(M = order.clones(dataB.A.NoBGS.s$ratios,
+     y = 544), nb = 6, nw = 2)
> fixed.bayesian <- stat.bayesian(M = order.clones(dataB.F.NoBGS.s$ratios,
+     y = 544), nb = 6, nw = 2)
> histogram.bayesian <- stat.bayesian(M = order.clones(dataB.H.NoBGS.s$ratios,
+     y = 544), nb = 6, nw = 2)
> genepix.bayesian <- stat.bayesian(M = order.clones(Res.genepix,
+     y = 544), nb = 6, nw = 2)
```

The number of genes with a log odds ratio of more than zero for each method was as follows:

```
> length(which(adaptive.bayesian$lods > 0))

[1] 345

> length(which(fixed.bayesian$lods > 0))

[1] 967
```

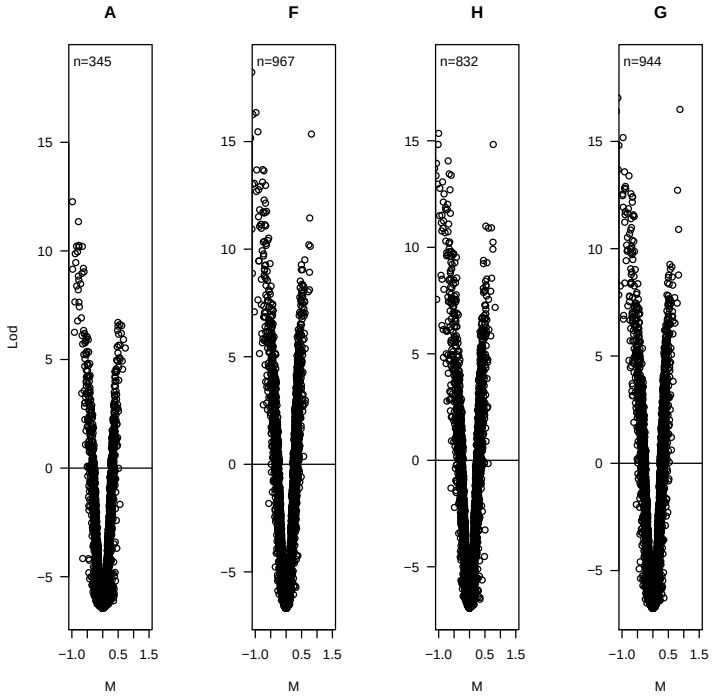
```
> length(which(histogram.bayesian$lods > 0))
```

```
[1] 832
```

```
> length(which(genepix.bayesian$lods > 0))
```

```
[1] 944
```

We plot the volcano plot for each method of segmentation with the log2 ratios (M) on the x axis and the lods, probabilities of differential expression, on the y axis.



References

[1] Ihaka, R. and Gentleman, R. (1996) R: A language for data analysis and graphics. *J. Comp. Graph. Stat.*, **5**, 299–314.

[2] Dudoit, S., Yang, Y. H., Callow, M. J. and Speed, T. P. (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sin.*, **12**, 111–139.

[3] Brown, C. S., Goodwin, P. C. and Sorger, P. K. (2001) Image metrics in the statistical analysis of DNA microarray data. *Proc. Natl. Acad Sci U.S.A.*, **98**, 8944–9.

[4] Yue, H., Eastman, P. S., Wang, B. B., Minor, J., Doctolero, M. H., Nuttall, R. L., Stack, R., Becker, J. W., Montgomery, J. R., Vainer, M. and Johnston, R. (2001) An evaluation of the performance of cDNA microarrays

for detecting changes in global mRNA expression. *Nucleic Acids Res.*, **29**, E41–1.

[5] Bland, J. M. and Altman, D. G. (1986) Statistical methods for assessing agreement between two methods of clinical measurement. *Lancet*, **1**, 307–10.

[6] Jenssen, T. K., Langaas, M., Kuo, W. P., Smith-Sorensen, B., Myklebost, O. and Hovig, E. (2002) Analysis of repeatability in spotted cDNA microarrays. *Nucleic Acids Res.*, **30**, 3235–44.

[7] Lee, M. L., Kuo, F. C., Whitmore, G. A. and Sklar, J. (2000) Importance of replication in microarray gene expression studies: statistical methods and evidence from repetitive cDNA hybridizations. *Proc. Natl. Acad. Sci. U.S.A.*, **97**, 9834–9.

[8] Lönnstedt, I. and Speed, T. (2002) Replicated microarray data. *Stat. Sin.*, **12**, 31–46.