Estimating copy number for Affymetrix 6.0 with the crlmm Package

Rob Scharpf

February, 2009

1 Estimating copy number

General requirements: in addition to the R packages indicated below, processing a large number of samples requires a high-end computer with a large amount of RAM. Currently, the maximum number of samples that can be genotyped at once is approximately 2000 and this requires roughly 32G of RAM.

Suggested work flow. I typically submit code for preprocessing and genotyping as a batch job to a computing cluster. When the preprocessing is complete, I pick a chromosome and work interactively from the cluster to calculate copy number and make some of the suggested diagnostic plots.

1.1 Preprocess and genotype the samples

First, load the required libraries.

```
R> library(crlmm)
R> library(genomewidesnp6Crlmm)
R> library(ellipse)
```

Specify an output directory and provide the complete path for the CEL file names.

```
R> outdir <- "/thumper/ctsa/snpmicroarray/rs/data/gain/bipolar/GRU-EA"
R> datadir <- "/thumper/ctsa/snpmicroarray/GAIN/Bipolar/GRU-EA"
R> fns <- list.files(datadir, pattern = ".CEL", full.names = TRUE)
```

Genotype the samples with CRLMM. Submit this job to the cluster and save the output to `outdir`.

```
R> res2 <- crlmm(filenames = fns, save.it = TRUE,
     intensityFile = file.path(outdir, "intensities.rda"))
R> save(res2, file = file.path(outdir, "res2.rda"))
```

Quantile normalize the nonpolymorphic probes and save the output:

```
R> res3 <- cnrma(fns)
R> save(res3, file = file.path(outdir, "res3.rda"))
```

1.2 Copy number

Copy number can be assessed one chromosome at a time. Here we specify chromosome 15 and and load a list of indices used to subset the files saved in the previous section. The first element in the list correspond to indices of polymorphic probes on chromosome 15; the second element corresponds to indices of nonpolymorphic probes on chromosome 15.

```
R> CHR <- 15
R> CHR_INDEX <- paste(CHR, "index", sep = "")
R> data(list = CHR_INDEX, package = "genomewidesnp6Crlmm")
R> str(index)
```

```
List of 2
 $ snps: int [1:26074] 18301 18302 18303 18304 18305 18306 18307 18308 18309 18310 ...
 $ nps : int [1:26690] 722819 722817 722818 722823 722820 722821 722822 722827 722824 722825
```

Next we load 3 files that were saved from the preprocessing step and then subset these lists using the above indices to extract the preprocessed intensities and genotypes needed for estimating copy number. Specifically, we require 6 items:

- quantile-normalized A intensities (I1 x J)
- quantile-normalized B intensities (I1 x J)
- quantile-normalized intensities from nonpolymorphic (NP) probes (I2 x J)
- genotype calls (I1 x J)
- confidence scores of the genotype calls (I1 x J)
- signal to noise ratio (SNR) of the samples (J)

These items are extracted as follows:

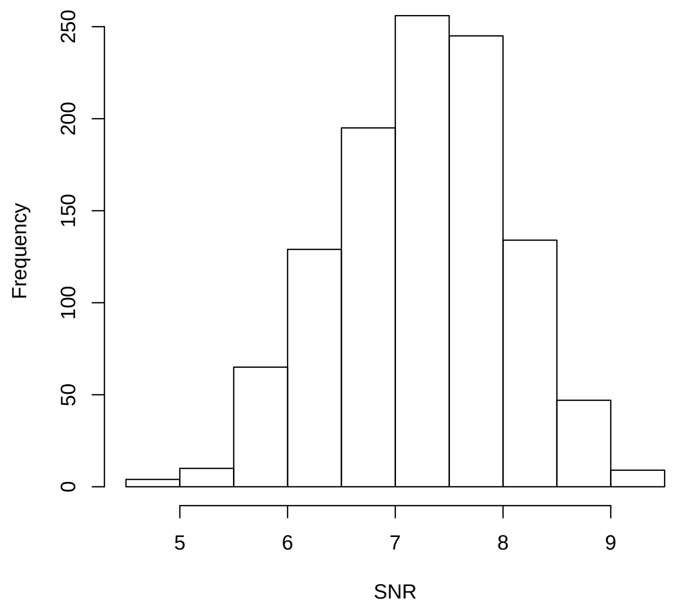
```
R> load(file.path(outdir, "intensities.rda"))
R> load(file.path(outdir, "res2.rda"))
R> load(file.path(outdir, "res3.rda"))
```

Depending on the number of samples, the above objects can be large and take a while to load. For the figures in this vignette, I am loading only 5000 SNP-level summaries from chromosome 22 (28MB) and the quantile-normalized intensities for 5000 nonpolymorphic probes on chromosome 22 (8 MB total).

```
R> A <- res$A
R> B <- res$B
R> calls <- res2$calls
R> conf <- res2$conf
R> SNR <- res2$SNR
R> NP <- res3$NP
R> rm(res, res2, res3)
R> gc()
```

Make a histogram of the signal to noise ratio for these samples:

```
R> hist(SNR, xlab = "SNR", main = "")
```



We suggest excluding samples with a signal to noise ratio less than 5. We then extract the probes (rows) that are on chromosome 15 and the samples (columns) that have a suitable signal to noise ratio.

```
R> A <- A[index[[1]], SNR > 5]
R> B <- B[index[[1]], SNR > 5]
R> calls <- calls[index[[1]], SNR > 5]
R> conf <- conf[index[[1]], SNR > 5]
R> NP <- NP[index[[2]], SNR > 5]
R> rm(res, res2, res3)
R> gc()
```

We want to adjust for batch effects. In our experience, the chemistry plate is a good surrogate for batch, although this should be assessed on a study by study basis. For samples processed by Broad, the plate is indicated by a capitalized five-letter word and is part of the sample name. It is important that the plate (batch) variable is ordered the same as filenames in the preprocessed data. For instance, to indicate plate in this example one can use the command

```
R> sns <- colnames(calls)
R> sns[1]
```

```
[1] "10056_INGLE_g_Plate_No._27_GenomeWideSNP_6_H09_50844.CEL"
```

```
R> plates <- sapply(sns, function(x) strsplit(x,
     "_")[[1]][2])
R> table(plates)
```

```
plates
ANNUL ARDOR BENCH BLOOD CHOMP CORSE CRAVE EMEUS FLIPS GRAPY
   34     1    55    46    34    45    48    47    37    20
IMAGE INERT INGLE INNED JOYED KRAAL MAXES MBIRA MOTET RIYAL
   54    25    53     5    34    21    38    49    39    46
SAPID SEELY SHARD STAYS THYME TOWNY TREYS TWAES VULGO
   46    12    46    14    50    62    36    39    54
```

We are now ready to estimate the copy number for each batch. In the current version of this package, one specifies an environment to which intermediate R objects for copy number estimation are stored. Allele-specific estimates of copy number are also stored in this environment.

```
R> e1 <- new.env()
R> computeCnBatch(A = A, B = B, calls = calls, conf = conf,
     NP = NP, plate = plates, envir = e1, chrom = CHR,
     DF.PRIOR = 75)
```

The DF.PRIOR indicates how much we will shrink SNP-specific estimates of the variance and correlation.

```
R> copyA <- get("CA", e1)
R> copyB <- get("CB", e1)
R> copyT <- (copyA + copyB)/100
R> copyT[copyT < 0] <- 0
R> copyT[copyT > 6] <- 6
```

2 Suggested plots

One sample at a time

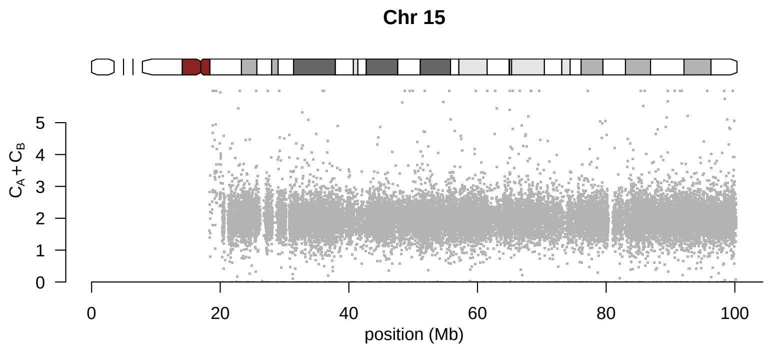


Figure 1: Total copy number (y-axis) for chromosome 22 plotted against physical position (x-axis) for one sample.

```
R> data(snpProbes, package = "genomewidesnp6Crlmm")
R> data(cnProbes, package = "genomewidesnp6Crlmm")
R> position <- snpProbes[match(rownames(calls), rownames(snpProbes)),
     "position"]
```

To plot physical position versus copy number for the first sample:

```
R> require(SNPchip)
R> par(las = 1)
R> plotCytoband(as.character(CHR), ylim = c(0, 7),
     cytoband.ycoords = c(6.5, 7), label.cytoband = FALSE,
     main = "Chr 15")

NULL

R> points(position, copyT[, 1], pch = ".", cex = 2,
     , xaxt = "n", col = "grey70")
R> axis(1, at = pretty(range(position)), labels = pretty(range(position))/1e+06)
R> axis(2, at = 0:5, labels = 0:5)
R> mtext("position (Mb)", 1, line = 2)
R> mtext(expression(C[A] + C[B]), 2, line = 2, las = 3)
```

One SNP at a time

```
R> tau2A <- get("tau2A", e1)
R> tau2B <- get("tau2B", e1)
R> sig2A <- get("sig2A", e1)
R> sig2B <- get("sig2B", e1)
R> nuA <- get("nuA", e1)
```

```
R> phiA <- get("phiA", e1)
R> nuB <- get("nuB", e1)
R> phiB <- get("phiB", e1)
R> corr <- get("corr", e1)
R> corrA.BB <- get("corrA.BB", e1)
R> corrB.AA <- get("corrA.BB", e1)
R> A <- get("A", e1)
R> B <- get("B", e1)
```

Here, we plot the prediction regions for total copy number 2 and 3 for the first plate. Black plotting symbols are estimates from the first plate; light grey are points from other plates. (You could also draw prediction regions for 0-4 copies, but it gets crowded). Notice that there is little evidence of a plate effect for this SNP.

```
R> par(las = 1)
R> p <- 1
R> J <- grep(unique(plates)[p], sns)
R> ylim <- c(6.5, 13)
R> I <- which(phiA > 10 & phiB > 10)
R> i <- I[1]
R> plot(log2(A[i, ]), log2(B[i, ]), pch = as.character(calls[i,
     ]), col = "grey60", cex = 0.9, ylim = ylim,
     xlim = ylim, xlab = "A", ylab = "B")
R> points(log2(A[i, J]), log2(B[i, J]), col = "black",
     pch = as.character(calls[i, J]))
R> for (CT in 2:3) {
     if (CT == 2)
         ellipse.col <- "brown"
     else ellipse.col <- "purple"
     for (CA in 0:CT) {
         CB <- CT - CA
         A.scale <- sqrt(tau2A[i, p] * (CA == 0) +
             sig2A[i, p] * (CA > 0))
         B.scale <- sqrt(tau2B[i, p] * (CB == 0) +
             sig2B[i, p] * (CB > 0))
         scale <- c(A.scale, B.scale)
         if (CA == 0 & CB > 0)
             rho <- corrA.BB[i, p]
         if (CA > 0 & CB == 0)
             rho <- corrB.AA[i, p]
         if (CA > 0 & CB > 0)
             rho <- corr[i, p]
         lines(ellipse(x = rho, centre = c(log2(nuA[i,
             p] + CA * phiA[i, p]), log2(nuB[i,
             p] + CB * phiB[i, p])), scale = scale),
             col = ellipse.col, lwd = 2)
```

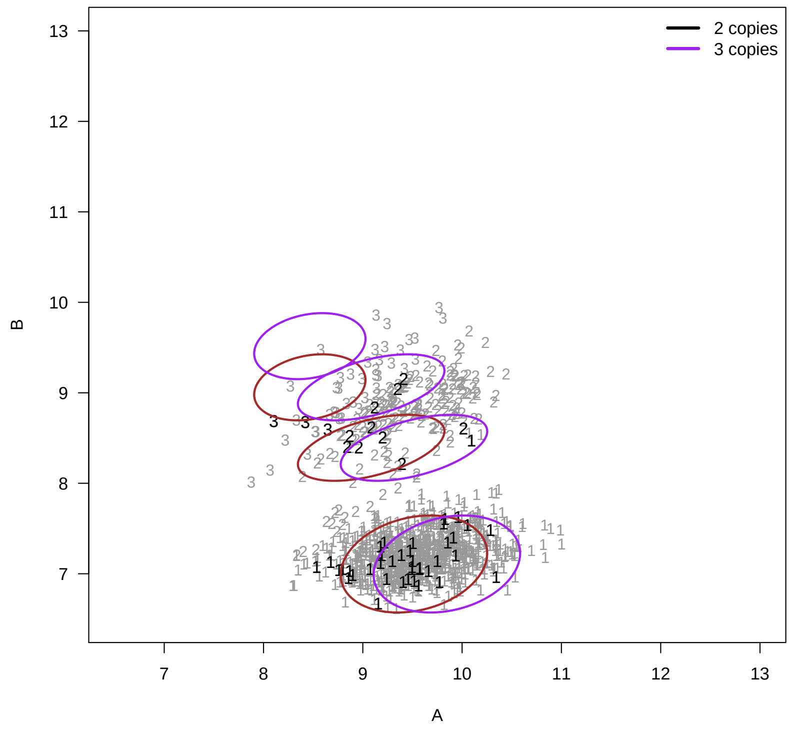


Figure 2: Prediction regions for copy number 2 and 3 for one SNP. The plate effects are negligible for this SNP and we only plot the prediction region for the first plate.

```
     }
 }
R> legend("topright", lwd = 3, col = c("black", "purple"),
     legend = c("2 copies", "3 copies"), bty = "n")
```

Here is one way to identify a SNP with a large plate effect – look for shifts in the prediction regions (here I'm lucking for large shifts in the A direction).

```
R> shiftA <- nuA + 2 * phiA
R> maxA <- apply(shiftA, 1, max, na.rm = T)
R> minA <- apply(shiftA, 1, min, na.rm = T)
R> d <- maxA - minA
R> hist(d)
R> index <- which(d > 3000)[2]
R> plate1 <- which(shiftA[index, ] == maxA[index])
```

```
R> plate2 <- which(shiftA[index, ] == minA[index])
```

Now plot the predictions for the two plates. All the ellipses are prediction regions for copy number 2. Note that while I looked for shifts in the A direction, the shift often occurs in both the B and A dimensions.

```
R> par(las = 1)
R> i <- index
R> J1 <- grep(unique(plates)[plate1], sns)
R> J2 <- grep(unique(plates)[plate2], sns)
R> ylim <- c(6.5, 13)
R> plot(log2(A[i, ]), log2(B[i, ]), pch = as.character(calls[i,
     ]), col = "grey60", cex = 0.9, ylim = ylim,
     xlim = ylim, xlab = "A", ylab = "B")
R> points(log2(A[i, J1]), log2(B[i, J1]), col = "brown",
     pch = as.character(calls[i, J1]))
R> points(log2(A[i, J2]), log2(B[i, J2]), col = "blue",
     pch = as.character(calls[i, J2]))
R> p <- plate1
R> CT <- 2
R> ellipse.col <- "brown"
R> for (CA in 0:CT) {
     CB <- CT - CA
     A.scale <- sqrt(tau2A[i, p] * (CA == 0) +
         sig2A[i, p] * (CA > 0))
     B.scale <- sqrt(tau2B[i, p] * (CB == 0) +
         sig2B[i, p] * (CB > 0))
     scale <- c(A.scale, B.scale)
     lines(ellipse(x = corr[i, p] * (CA > 0) *
         (CB > 0), centre = c(log2(nuA[i, p] +
         CA * phiA[i, p]), log2(nuB[i, p] + CB *
         phiB[i, p])), scale = scale), col = ellipse.col,
         lwd = 2)
 }
R> ellipse.col <- "blue"
R> p <- plate2
R> for (CA in 0:CT) {
     CB <- CT - CA
     A.scale <- sqrt(tau2A[i, p] * (CA == 0) +
         sig2A[i, p] * (CA > 0))
     B.scale <- sqrt(tau2B[i, p] * (CB == 0) +
         sig2B[i, p] * (CB > 0))
     scale <- c(A.scale, B.scale)
     lines(ellipse(x = corr[i, p] * (CA > 0) *
         (CB > 0), centre = c(log2(nuA[i, p] +
         CA * phiA[i, p]), log2(nuB[i, p] + CB *
```

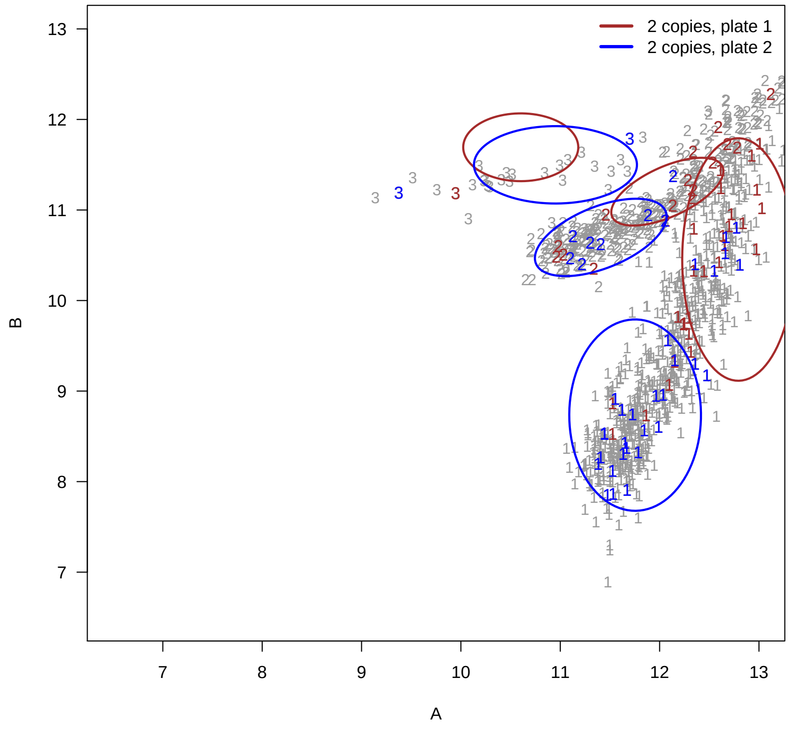


Figure 3: The prediction regions for copy number 2 shift when there is a large plate effect.

```
          phiB[i, p])), scale = scale), col = ellipse.col,
          lwd = 2)
 }
R> legend("topright", lwd = 3, col = c("brown", "blue"),
     legend = c("2 copies, plate 1", "2 copies, plate 2 "),
     bty = "n")
```

Alternatively, loop through a large number of SNPs to see how these regions move

```
R> par(las = 1, pty = "s", ask = TRUE)
R> for (i in I[1:50]) {
     plot(log2(A[i, ]), log2(B[i, ]), pch = as.character(calls[i,
         ]), col = col[DS], cex = 0.9, ylim = ylim,
         xlim = ylim)
     for (CT in 2:3) {
         if (CT == 2)
             ellipse.col <- "brown"
         else ellipse.col <- "purple"
         for (CA in 0:CT) {
             CB <- CT - CA
             A.scale <- sqrt(tau2A[i, p] * (CA ==
                 0) + sig2A[i, p] * (CA > 0))
             B.scale <- sqrt(tau2B[i, p] * (CB ==
                 0) + sig2B[i, p] * (CB > 0))
             scale <- c(A.scale, B.scale)
             lines(ellipse(x = corr[i, p] * (CA >
                 0) * (CB > 0), centre = c(log2(nuA[i,
                 p] + CA * phiA[i, p]), log2(nuB[i,
                 p] + CB * phiB[i, p])), scale = scale),
                 col = ellipse.col, lwd = 2)
         }
     }
     legend("topright", lwd = 3, col = col, legend = c("3 copies",
         "2 copies"), bty = "n")
 }
```

3 Session information

```
R> sessionInfo()

R version 2.9.0 Under development (unstable) (2009-02-08 r47879)
x86_64-unknown-linux-gnu

locale:
LC_CTYPE=en_US.iso885915;LC_NUMERIC=C;LC_TIME=en_US.iso885915;LC_COLLATE=en_US.iso885915;LC_

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

other attached packages:
[1] SNPchip_1.7.1           oligoClasses_1.5.5
[3] genefilter_1.23.2       Biobase_2.3.10
[5] ellipse_0.3-5           genomewidesnp6Crlmm_1.0
[7] crlmm_1.0.35

loaded via a namespace (and not attached):
[1] affyio_1.11.3        annotate_1.21.3
[3] AnnotationDbi_1.5.15 DBI_0.2-4
[5] preprocessCore_1.5.3 RSQLite_0.7-1
```

```
[7] survival_2.34-1      tools_2.9.0
[9] xtable_1.5-4
```