Trisomy analysis

Rob Scharpf

May, 2009

1 Estimating copy number

At present, software for copy number estimation is provided only for the Affymetrix 6.0 platform. This vignette estimates copy number for the HapMap samples.

1.1 Preprocess and genotype the samples

We preprocess and genotype the samples as described in the CRLMM vignette.

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

Specify the complete path for the CEL files and a directory in which to store intermediate files:

```
R> celFiles <- list.celfiles("/thumper/ctsa/snpmicroarray/hapmap/raw/affy/1m",
     full.names = TRUE, pattern = ".CEL")
R> outdir <- "/thumper/ctsa/snpmicroarray/rs/data/hapmap/1m/affy"
```

Preprocess and genotype (for more info see the crlmm vignette):

```
R> crlmmFilenames <- file.path(outdir, paste("crlmmResults_",
     1:24, ".rda", sep = ""))
R> if (!all(file.exists(crlmmFilenames))) {
     message("Preprocessing and crlmm genotyping.  Results are written to",
         outdir, "...")
     intensityFile <- file.path(outdir, "intensities.rda")
     crlmmWrapper(celFiles, outdir, save.it = TRUE,
         intensityFile = intensityFile)
 }
```

Load results for chromosome 22.

```
R> CHR <- 22
R> if (!exists("crlmmResults")) load(file.path(outdir,
     paste("crlmmResults_", CHR, ".rda", sep = "")))
R> class(crlmmResults)
```

```
[1] "CrlmmSetList"
attr(,"package")
[1] "crlmm"
```

```
R> show(crlmmResults)
```

```
ABset (storageMode: lockedEnvironment)
assayData: 24760 features, 96 samples
  element names: A, B
phenoData
  sampleNames: SL272_1000K.CEL, 135.3_1000K.CEL, ..., RH73_1
  000K.CEL  (96 total)
  varLabels and varMetadata description:
    SNR: NA
    gender: NA
featureData
  featureNames: CN_220571, CN_893722, ..., CN_893721  (24760 total)
  fvarLabels and fvarMetadata description: none
experimentData: use 'experimentData(object)'
Annotation: genomewidesnp6
SnpSet (storageMode: lockedEnvironment)
assayData: 24760 features, 96 samples
  element names: call, callProbability
phenoData
  sampleNames: SL272_1000K.CEL, 135.3_1000K.CEL, ..., RH73_1
  000K.CEL  (96 total)
  varLabels and varMetadata description:
    SNR: Signal-to-noise Ratio
    gender: Gender: Male (1) and Female (2)
    batchQC: Quality score for batch
featureData
  featureNames: CN_220571, CN_893722, ..., CN_893721  (24760 total)
  fvarLabels and fvarMetadata description:
    SNPQC: SNP Quality Score
    spAA: Shift in parameters AA
    spAB: Shift in parameters AB
    spBB: Shift in parameters BB
experimentData: use 'experimentData(object)'
Annotation: genomewidesnp6
```

1.2 Copy number

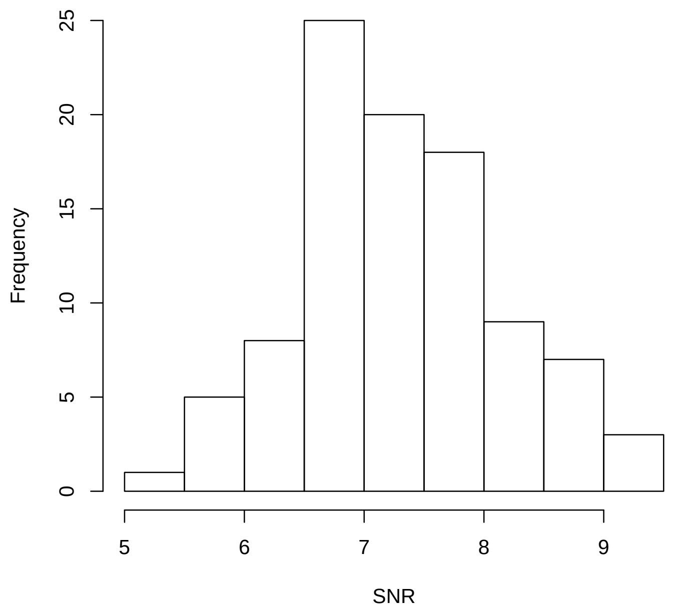
We require 6 items for copy number estimation:

- 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)

The above items are contained in the `crlmmResults` object. A histogram of the signal to noise ratio for the HapMap samples:

```
R> hist(crlmmResults[[2]]$SNR, xlab = "SNR", main = "")
```



We suggest excluding samples with a signal to noise ratio less than 5. As batch effects can be very large in the quantile-normalized intensities, we suggest adjusting for date or chemistry plate. Ideally, one would have 70+ files in a given batch. Here we make a table of date versus ancestry:

```
R> sns <- sampleNames(crlmmResults)
R> sns[1]

[1] "SL272_1000K.CEL"

R> batch <- substr(basename(sns), 13, 13)
R> table(batch)
```

```
batch
 0  2  C  E
 6  1 77 12
```

```
R> table(format(as.POSIXlt(scanDates(crlmmResults)),
     "%d %b %Y"), batch)
```

```
             batch
              0  2  C  E
  11 Dec 2007  0  0 19  0
  12 Dec 2007  0  1 43  8
  13 Dec 2007  0  0  9  1
  19 Nov 2008  1  0  6  3
  30 Oct 2008  5  0  0  0
```

As all of these samples were run on the first week of March, we would expect that any systematic artifacts to the intensities that develop over time to be minimal (a best case scenario). As this is typically not the case, we illustrate how one may adjust for batch using the chemistry plate as an argument for `batch` in the `computeCopynumber` function.

```
R> if (!exists("cnset")) {
     if (file.exists(file.path(outdir, paste("cnset_",
         CHR, ".rda", sep = ""))))
         load(file.path(outdir, paste("cnset_",
             CHR, ".rda", sep = "")))
     else {
         cnset <- computeCopynumber(crlmmResults,
             SNRmin = 5, batch = batch, CHR = CHR,
             cdfName = "genomewidesnp6")
         save(cnset, file = file.path(outdir, paste("cnset_",
             CHR, ".rda", sep = "")))
     }
 }
R> scanDates(cnset) <- scanDates(crlmmResults)
R> cnset <- cnset[order(chromosome(cnset), position(cnset)),
     ]
R> crlmmResults <- crlmm:::harmonizeDimnamesTo(crlmmResults,
     cnset)
```

The above algorithm for estimating copy number is predicated on the assumption that most samples within a batch have copy number 2 at any given locus. For common copy number variants, this assumption may not hold. An additional iteration using a bias correction provides additional robustness to this assumption. Set the `bias.adj` argument to `TRUE`:

```
R> cnset <- computeCopynumber(crlmmResults, SNRmin = 5,
     batch = batch, CHR = CHR, cdfName = "genomewidesnp6",
     bias.adj = TRUE)
```

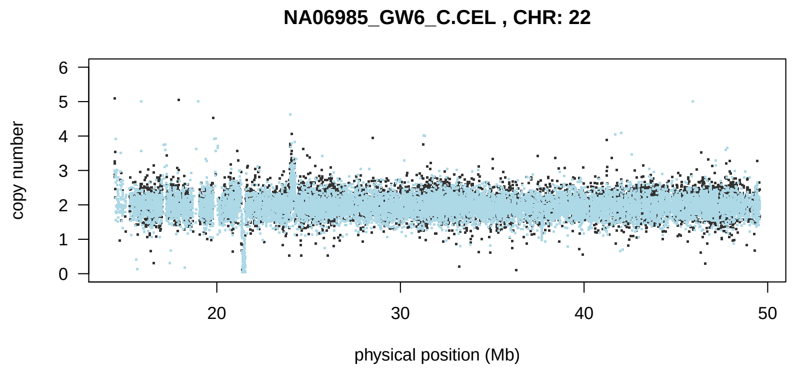


Figure 1: Total copy number (y-axis) for chromosome 22 plotted against physical position (x-axis) for one sample.

2 Suggested plots

One sample at a time Plot physical position versus copy number for the first sample. Recall that the copy number estimates were multiplied by 100 and stored as an integer.

```
R> par(las = 1)
R> plot(position(cnset), copyNumber(cnset)[, 1],
     pch = ".", cex = 2, xaxt = "n", col = "grey20",
     ylim = c(0, 6), ylab = "copy number", xlab = "physical position (Mb)",
     main = paste(sampleNames(cnset)[1], ", CHR:",
         unique(chromosome(cnset))))
R> points(position(cnset)[cnIndex(cnset)], copyNumber(cnset)[cnIndex(cnset),
     1], pch = ".", cex = 2, col = "lightblue")
R> axis(1, at = pretty(range(position(cnset))), labels = pretty(range(position(cnset)))/1e+(
```

One SNP at a time Plot the prediction regions for total copy number 2 and 3 for the first plate. Plotting symbols are the genotype calls (1=AA, 2=AB, 3=BB); light grey points are from other plates. One could also add the prediction regions for 0-4 copies, but it gets crowded.

```
R> xlim <- ylim <- c(6.5, 13)
R> pch <- 21
R> colors <- c("red", "blue", "green3")
R> cex <- 0.6
R> par(mfrow = c(3, 3), las = 1, pty = "s", ask = FALSE,
     mar = c(2, 2, 2, 2), oma = c(2, 2, 1, 1))
R> indices <- split(snpIndex(crlmmResults), rep(1:length(snpIndex(crlmmResults)),
     each = 9, length.out = length(snpIndex(crlmmResults))))
R> par(ask = TRUE)
R> j <- 1
R> for (i in indices[[j]]) {
     gt <- calls(crlmmResults)[i, ]
     plot(crlmmResults[i, ], pch = pch, col = colors[gt],
         bg = colors[gt], cex = cex, xlim = xlim,
         ylim = ylim)
     mtext("A", 1, outer = TRUE, line = 1)
     mtext("B", 2, outer = TRUE, line = 1)
 }

R> require(RColorBrewer)
R> greens <- brewer.pal(9, "Greens")
R> J <- split(1:ncol(cnset), cnset$batch)
R> colors <- c("red", "blue", "green3")
R> cex <- 0.6
R> colors <- c("blue", greens[8], "red")
R> pch.col <- c("grey40", "black", "grey40")
R> xlim <- ylim <- c(6.5, 13)
R> plotpoints <- FALSE
R> lwd <- 2
R> pdf("figures/snp22plots%02d.pdf", width = 600,
     height = 600)
R> ask <- FALSE
R> par(mfrow = c(3, 3), las = 1, pty = "s", ask = ask,
     mar = c(2, 2, 2, 2), oma = c(2, 2, 1, 1))
R> indices <- split(snpIndex(crlmmResults), rep(1:length(snpIndex(crlmmResults)),
     each = 9, length.out = length(snpIndex(crlmmResults))))
R> for (j in seq(along = indices)[1:10]) {
     cat(j, "\n")
     k <- 1
     for (i in indices[[j]]) {
         gt <- calls(crlmmResults)[i, ]
         pch <- as.character(gt)
         cex <- 0.9
         plot(crlmmResults[i, ], pch = pch, col = pch.col[gt],
             cex = cex, xlim = xlim, ylim = ylim,
             type = "n")
         if (plotpoints) {
             for (b in seq(along = unique(cnset$batch))) {
                 points(crlmmResults[i, J[[b]]],
                   pch = pch, col = colors[b],
                   bg = colors[b], cex = cex, xlim = xlim,
                   ylim = ylim)
             }
```

```
        }
        for (b in seq(along = unique(cnset$batch))) {
            ellipse(cnset[i, J[[b]]], copynumber = 2,
                col = colors[b], lwd = lwd)
        }
        if (k == 1) {
            legend("bottomleft", bty = "n", fill = colors,
                legend = c("CEPH", "Yoruba", "Asian"))
            mtext("A", 1, outer = TRUE, line = 1)
            mtext("B", 2, outer = TRUE, line = 0)
        }
        k <- k + 1
    }
 }
R> dev.off()
```

3 Smoothing via a hidden Markov model

Here we smooth via a hidden Markov model. To facilitate comparisons with the Birdseye HMM, the same transition probabilities are specified.

```
R> library(VanillaICE)
R> copyNumberStates <- 0:5
R> require(Biobase)
R> cnset <- crlmm:::thresholdModelParams(cnset)
R> if (!exists("fit")) {
     if (file.exists(file.path(outdir, paste("fit_",
         CHR, ".rda", sep = ""))))
         load(file.path(outdir, paste("fit_", CHR,
             ".rda", sep = "")))
     else {
         emission.cn <- computeEmission(crlmmResults,
             cnset, copyNumberStates)
         tau <- transitionProbability(chromosome = chromosome(cnset),
             position = position(cnset), TAUP = 1e+08)
         initialP <- rep(1/length(copyNumberStates),
             length(copyNumberStates))
         fit <- viterbi(initialStateProbs = log(initialP),
             emission = emission.cn, tau = tau[,
                 "transitionPr"], arm = tau[, "arm"],
             normalIndex = 2, normal2altered = 0.005,
             altered2normal = 0.5, altered2altered = 0.0025)
         brks <- breaks(x = fit, states = copyNumberStates,
             position = tau[, "position"], chromosome = tau[,
                 "chromosome"], sampleNames = sampleNames(cnset))
```

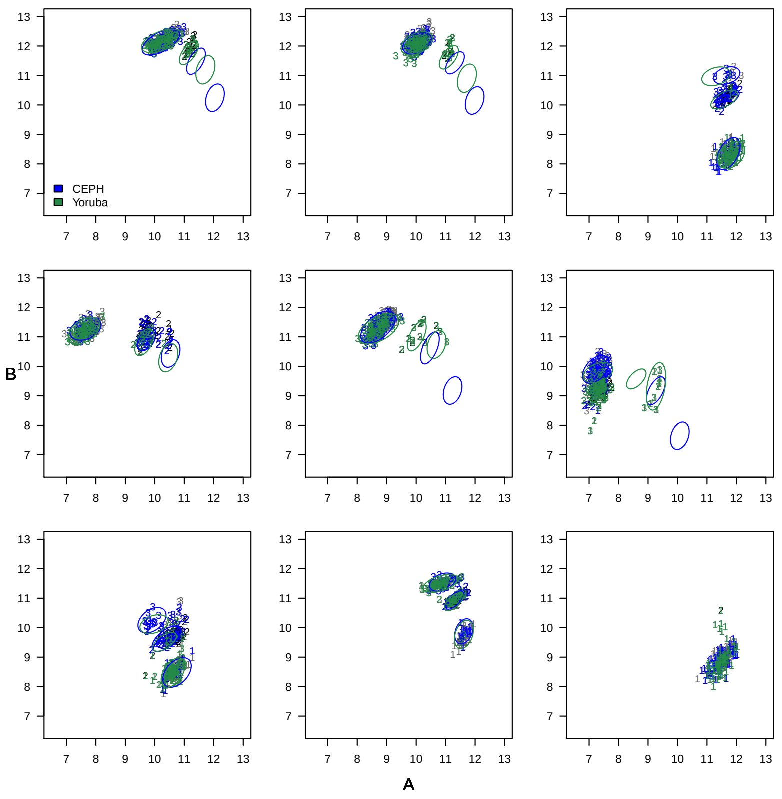


Figure 2: Prediction regions for copy number 2 for one SNP. The plate effects are negligible for this SNP and we only plot the prediction region for the first plate.

```
     }
 }

R> dir.create("figures")
R> par(las = 1, mfrow = c(1, 1), ask = FALSE)
R> bmp("figures/chr22plots%02d.png", width = 800,
     height = 500)
R> for (j in 1:ncol(cnset)) {
     cat(j, " ")
     index <- 1:nrow(fit)
     plot(0:1, type = "n", xlim = range(position(cnset)[index]),
         xaxt = "n", ylim = c(-0.5, 6), ylab = "copy number",
         xlab = "physical position (Mb)", main = paste(sampleNames(cnset)[j],
             ", CHR:", unique(chromosome(cnset))),
         lwd = 2, col = "blue")
     points(position(cnset)[index], copyNumber(cnset)[index,
         j], pch = ".", cex = 2, col = "grey60")
     lines(position(cnset)[index], fit[index, j] -
         1, type = "s", lwd = 2, col = "blue")
     abline(h = 0:4, col = "grey60")
     axis(1, at = pretty(range(position(cnset)[index])),
         labels = pretty(range(position(cnset)[index]))/1e+06)
 }
R> dev.off()
```

4 Session information

```
R> toLatex(sessionInfo())
```

- R version 2.10.0 Under development (unstable) (2009-05-28 r48680), `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_MONETARY=C`, `LC_MESSAGES=en_US.iso885915`, `LC_PAPER=en_US.iso885915`, `LC_NAME=C`, `LC_ADDRESS=C`, `LC_TELEPHONE=C`, `LC_MEASUREMENT=en_US.iso885915`, `LC_IDENTIFICATION=C`
- Base packages: base, datasets, graphics, grDevices, methods, stats, utils
- Other packages: Biobase 2.5.3, crlmm 1.3.6, ellipse 0.3-5, genomewidesnp6Crlmm 1.0.4, RColorBrewer 1.0-2, VanillaICE 1.7.3, xtable 1.5-5
- Loaded via a namespace (and not attached): affyio 1.13.3, annotate 1.23.0, AnnotationDbi 1.7.0, Biostrings 2.13.10, DBI 0.2-4,

genefilter 1.25.2, IRanges 1.3.26, mvtnorm 0.9-7, oligoClasses 1.7.4, preprocessCore 1.7.4, RSQLite 0.7-1, SNPchip 1.9.0, splines 2.10.0, survival 2.35-4