Copy number estimation and genotype calling with crlmm

Rob Scharpf

May, 2009

Abstract

This vignette estimates copy number for HapMap samples on the Affymetrix 6.0 platform. See [1] for the working paper.

1 Simple usage

CRLMM supports the following platforms:

```
R> library(crlmm)
R> crlmm:::validCdfNames()
```

```
[1] "genomewidesnp6"  "genomewidesnp5"  "human370v1c"
[4] "human370quadv3c" "human550v3b"     "human650v3a"
[7] "human610quadv1b" "human660quadv1a" "human1mduov3b"
```

Preprocess and genotype. Provide the complete path for the filenames:

```
R> celFiles <- list.celfiles("/thumper/ctsa/snpmicroarray/hapmap/raw/affy/1m",
     full.names = TRUE, pattern = ".CEL")
R> batch <- substr(basename(celFiles), 13, 13)
R> celFiles <- celFiles[batch %in% c("C", "Y")]
R> batch <- batch[batch %in% c("C", "Y")]
```

While genotyping with crlmm can be performed using a small number of samples, copy number estimation requires at least 10 samples – preferably all of the samples that were processed together on the same plate. Crlmm does not use a reference dataset when estimating model parameters because of large batch effects [1]. The quantile normalization performed as part of the preprocessing of the raw data is insufficient for removing batch effects. Processing a reference dataset, such as HapMap samples, along with the experimental data will not improve copy number estimation for the experimental dataset, and should not be used as a means to increase the sample size. Furthermore, processing a reference dataset without acknowledging that these samples were derived from a different batch can result in incorrect copy number estimates in both the experimental and reference datasets. The appropriate way to acknowledge batch is to supply the batch name for each sample to be processed in the argument to `cnOptions`:

```
R> cnOpts <- cnOptions(cdfName = "genomewidesnp6",
     outdir = "/thumper/ctsa/beaty/scharpf/crlmmOut/hapmap",
     batch = batch)
R> str(cnOpts)
```

```
List of 27
 $ outdir              : chr "/thumper/ctsa/beaty/scharpf/crlmmOut/hapmap"
 $ cdfName             : chr "genomewidesnp6"
 $ crlmmFile           : chr "/thumper/ctsa/beaty/scharpf/crlmmOut/hapmap/snpsetObject.rda"
 $ intensityFile       : chr "/thumper/ctsa/beaty/scharpf/crlmmOut/hapmap/normalizedIntensities.rda"
 $ rgFile             : chr "/thumper/ctsa/beaty/scharpf/crlmmOut/hapmap/rgFile.rda"
 $ save.it            : logi TRUE
 $ save.cnset         : logi TRUE
 $ load.it            : logi TRUE
 $ splitByChr         : logi TRUE
 $ MIN.OBS            : num 3
 $ MIN.SAMPLES        : num 10
 $ batch              : chr [1:180] "C" "C" "C" "C" ...
 $ DF.PRIOR           : num 50
 $ GT.CONF.THR        : num 0.99
 $ prior.prob         : num [1:4] 0.25 0.25 0.25 0.25
 $ bias.adj           : logi FALSE
 $ SNRmin             : num 4
 $ chromosome         : int [1:24] 1 2 3 4 5 6 7 8 9 10 ...
 $ seed               : num 123
 $ verbose            : logi TRUE
 $ PHI.THR            : num 64
 $ nHOM.THR           : num 5
 $ MIN.NU             : num 8
 $ MIN.PHI            : num 8
 $ THR.NU.PHI         : logi TRUE
 $ thresholdCopynumber: logi TRUE
 $ unlink             : logi TRUE
```

```
R> stopifnot(length(cnOpts$batch) == length(celFiles))
R> names(cnOpts$batch) <- basename(celFiles)
```

The next code chunk quantile normalizes the samples to a target reference distribution, uses the crlmm algorithm to genotype, and then estimates the copy number for each batch. Currently processing the 180 HapMap cel files will require less than 20G of RAM. We are working on methods to reduce the memory footprint.

```
R> if (FALSE) crlmmCopynumber(celFiles, cnOpts)
R> load(file.path(cnOpts[["outdir"]], "cnSet_21.rda"))
```

The following R objects are created from crlmmCopynumber:

```
R> fns <- list.files(cnOpts[["outdir"]], pattern = "cnSet",
     full.name = TRUE)
R> basename(fns)[1:5]

[1] "cnSet_10.rda" "cnSet_11.rda" "cnSet_12.rda"
[4] "cnSet_13.rda" "cnSet_14.rda"
```

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> cnOpts[["bias.adj"]] <- TRUE
R> if (FALSE) crlmmCopynumber(celFiles, cnOpts)
```

2 Accessors

2.1 Assay data accessors

`ABset`**: quantile normalized intensities** An object of class `ABset` is stored in the first element of the `crlmmSetList` object. The following accessors may be of use:

Accessors for the quantile normalized intensities for the A allele at polymorphic loci:

```
R> a <- A(cnSet)[isSnp(cnSet), ]
R> dim(a)
```

```
[1] 12579   180
```

The quantile-normalized intensities for nonpolymorphic loci are obtained by:

```
R> npIntensities <- A(cnSet)[!isSnp(cnSet), ]
```

Quantile normalized intensities for the B allele at polymorphic loci:

```
R> b.snps <- B(cnSet[isSnp(cnSet), ])
```

Note that NAs are recorded in the 'B' assay data element for nonpolymorphic loci:

```
R> all(is.na(B(cnSet[!isSnp(cnSet), ])))
```

```
[1] TRUE
```

`SnpSet`**: Genotype calls and confidence scores** Genotype calls:

```
R> genotypes <- snpCall(cnSet)
```

Confidence scores of the genotype calls:

```
R> genotypeConf <- confs(cnSet[isSnp(cnSet), ])
```

`CopyNumberSet`**: allele-specific copy number** Allele-specific copy number at polymorphic loci:

```
R> ca <- CA(cnSet[isSnp(cnSet), ])
```

Total copy number at nonpolymorphic loci:

```
R> cn.nonpolymorphic <- CA(cnSet[!isSnp(cnSet), ])
```

Total copy number at both polymorphic and nonpolymorphic loci:

```
R> cn <- copyNumber(cnSet)
```

2.2 Other accessors

Information on physical position and chromosome can be accessed by the following accessors:

```
R> xx <- position(cnSet)
R> yy <- chromosome(cnSet)
```

There are many parameters computed during copy number estimation that are at present stored in the `featureData` slot. In particular, the estimation procedure fits a linear model to the normalized intensities for each allele. These parameters are not generally meant to be extracted by the user; for now we just mention where they are stored.

```
R> fvarLabels(cnSet)
```

```
 [1] "chromosome" "position"   "isSnp"      "SNPQC"     
 [5] "spAA"       "spAB"       "spBB"       "tau2A_C"   
 [9] "tau2A_Y"    "tau2B_C"    "tau2B_Y"    "sig2A_C"   
[13] "sig2A_Y"    "sig2B_C"    "sig2B_Y"    "nuA_C"     
[17] "nuA_Y"      "nuA.se_C"   "nuA.se_Y"   "nuB_C"     
[21] "nuB_Y"      "nuB.se_C"   "nuB.se_Y"   "phiA_C"    
[25] "phiA_Y"     "phiA.se_C"  "phiA.se_Y"  "phiB_C"    
[29] "phiB_Y"     "phiB.se_C"  "phiB.se_Y"  "phiAX_C"   
[33] "phiAX_Y"    "phiBX_C"    "phiBX_Y"    "corr_C"    
[37] "corr_Y"     "corrA.BB_C" "corrA.BB_Y" "corrB.AA_C"
[41] "corrB.AA_Y"
```

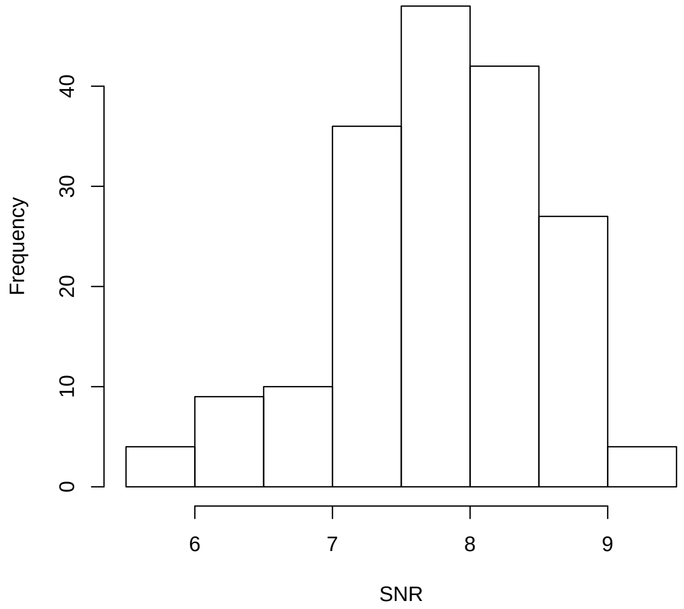

Figure 1: Signal to noise ratios for the HapMap samples.

SNR. A histogram of the signal to noise ratio for the HapMap samples:

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

Data/Batch. For Affymetrix 6.0, we currently suggest excluding or flagging samples with a signal to noise ratio less than 5. Adjusting by date or chemistry plate can be helpful for limiting the influence of batch effects. Ideally, one would have 70+ files in a given batch. Here we make a table of date versus ancestry (batch):

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).

3 Suggested visualizations

One sample at a time: locus-level estimates Figure 2 plots physical position (horizontal axis) versus copy number (vertical axis) for the first sample. There is less information to estimate copy number at nonpolymorphic loci; improvements to the univariate prediction regions at nonpolymorphic loci are a future area of research.

```
R> par(las = 1, mar = c(4, 5, 4, 2))
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)[!isSnp(cnSet)], copyNumber(cnSet)[!isSnp(cnSet),
     1], pch = ".", cex = 2, col = "lightblue")
R> axis(1, at = pretty(range(position(cnSet))), labels = pretty(range(position(cnSet)))/1e+06)
```

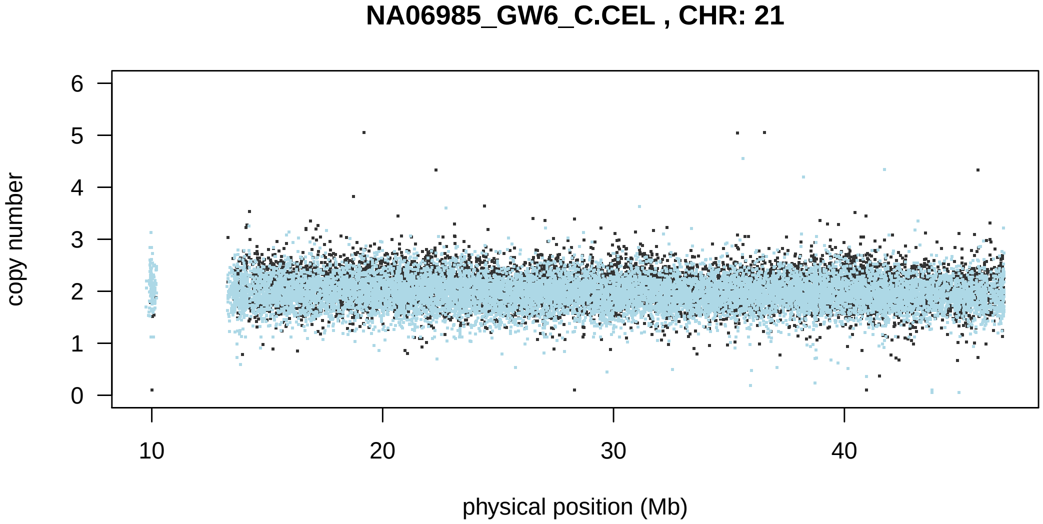

Figure 2: Total copy number (y-axis) for chromosome 22 plotted against physical position (x-axis) for one sample. Estimates at nonpolymorphic loci are plotted in light blue.

One SNP at a time Scatterplots of the A and B allele intensities (log-scale) can be useful for assessing the biallelic genotype calls. The following code chunk is displayed in Figure 3.

```
R> myScatter <- function(object, add = FALSE, ...) {
     A <- log2(A(object))
     B <- log2(B(object))
     if (!add) {
         plot(A, B, ...)
     }
     else {
         points(A, B, ...)
     }
 }
R> index <- which(isSnp(cnSet))[1:9]
R> xlim <- ylim <- c(6.5, 13)
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> for (i in index) {
     gt <- calls(cnSet)[i, ]
     if (i != 89) {
         myScatter(cnSet[i, ], pch = pch, col = colors[snpCall(cnSet)[i,
             ]], bg = colors[snpCall(cnSet)[i,
             ]], cex = cex, xlim = xlim, ylim = ylim)
         mtext("A", 1, outer = TRUE, line = 1)
         mtext("B", 2, outer = TRUE, line = 1)
         crlmm:::ellipse.CNSet(cnSet[i, ], copynumber = 2,
             batch = "C", lwd = 2, col = "black")
         crlmm:::ellipse.CNSet(cnSet[i, ], copynumber = 2,
             batch = "Y", lwd = 2, col = "grey50")
     }
     else {
         plot(0:1, xlim = c(0, 1), ylim = c(0,
             1), type = "n", xaxt = "n", yaxt = "n")
         legend("center", legend = c("CN = 2, CEPH",
             "CN = 2, Yoruban"), col = c("black",
             "grey50"), lwd = 2, bty = "n")
     }
 }
```

4 Session information

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

- R version 2.11.0 Under development (unstable) (2009-11-22 r50541), 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, tools, utils
- Other packages: Biobase 2.7.2, crlmm 1.5.20, lattice 0.17-26, oligoClasses 1.9.22, RColorBrewer 1.0-2

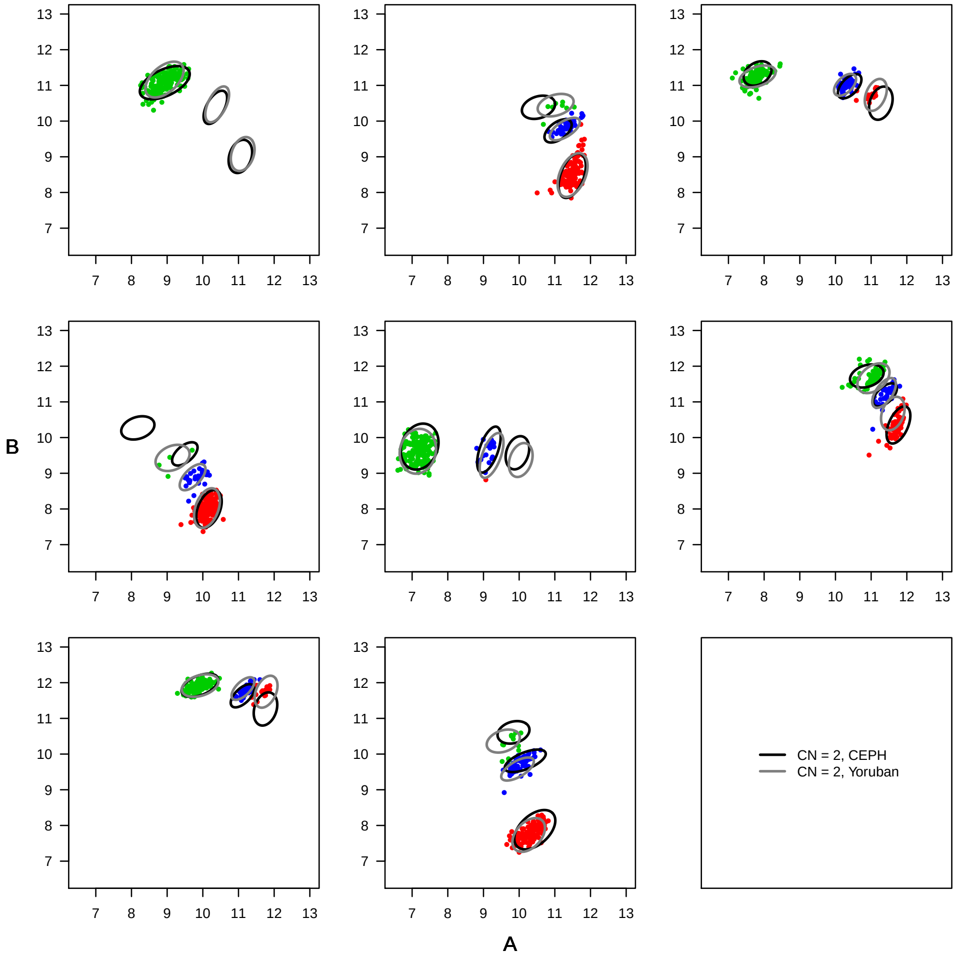

Figure 3: Scatterplots of A versus B intensities. Each panel displays a single SNP. The ellipses indicate the 95% probability region for copy number 2 for the CEPH (black) and Yoruban subjects (grey).

- Loaded via a namespace (and not attached): affyio 1.15.1, annotate 1.25.0, AnnotationDbi 1.9.2, Biostrings 2.15.11, DBI 0.2-4, ellipse 0.3-5, genefilter 1.29.3, grid 2.11.0, IRanges 1.5.21, mvtnorm 0.9-8, preprocessCore 1.9.0, RSQLite 0.7-3, splines 2.11.0, survival 2.35-7, xtable 1.5-6

References

References

[1] Robert B Scharpf, Ingo Ruczinski, Benilton Carvalho, Betty Doan, Aravinda Chakravarti, and Rafael Irizarry. A multilevel model to address batch effects in copy number estimation using snp arrays. May 2009.