

Using *crlmm* to genotype data from Illumina's Infinium BeadChips

Matt Ritchie

October 7, 2010

1 Getting started

In this user guide we read in and genotype data from 40 HapMap samples which have been analyzed using Illumina's 370k Duo BeadChips. This data is available in the *hapmap370k* package. Additional chip-specific model parameters and basic SNP annotation information used by CRLMM is stored in the *human370v1cCrlmm* package. The required packages can be installed in the usual way using the `biocLite` function.

```
> source("http://www.bioconductor.org/biocLite.R")
> biocLite(c("crlmm", "hapmap370k", "human370v1cCrlmm"))
```

2 Reading in data

The function `readIdatFiles` extracts the Red and Green intensities from the binary `idat` files output by Illumina's scanning device. The file `samples370k.csv` contains information about each sample.

```
> options(width = 50)

> library(Biobase)
> library(crlmm)
> library(hapmap370k)
> data.dir = system.file("idatFiles", package = "hapmap370k")
> samples = read.csv(file.path(data.dir,
+   "samples370k.csv"), as.is = TRUE)
> samples[1:5, ]

> RG = readIdatFiles(samples, path = data.dir,
+   arrayInfoColNames = list(barcode = NULL,
+   position = "SentryPosition"),
+   saveDate = TRUE)
```

Reading in this data takes approximately 100 seconds and peak memory usage was 0.8 GB of RAM on our linux system. If memory is limiting, load the *ff* package and run the same command. When this package is available, the objects are stored using disk rather than RAM. The *RG* object is an *NChannelSet* which stores the Red and Green intensities for each bead-type. The scanning date of each array is stored in *protocolData*.

```
> class(RG)

[1] "NChannelSet"
attr(,"package")
[1] "Biobase"

> dim(RG)

Features  Samples
 381079    40

> slotNames(RG)

[1] "assayData"          "phenoData"
[3] "featureData"       "experimentData"
[5] "annotation"        "protocolData"
[7] ".__classVersion__"

> channelNames(RG)

[1] "G"    "R"    "zero"

> exprs(channel(RG, "R"))[1:5, 1:5]

      4030186347_A 4030186263_B 4019585415_B
10008           321           170           2961
10010           1738          3702           3105
10025             80            101            145
10026           5043          1856           6519
10039           4905          2464           9080
      4031058127_B 4031058211_B
10008           3468            262
10010           3425             70
10025             29             21
10026           8304          9872
10039           9788          10867

> exprs(channel(RG, "G"))[1:5, 1:5]
```

| | | | |
|-------|--------------|--------------|--------------|
| | 4030186347_A | 4030186263_B | 4019585415_B |
| 10008 | 4183 | 4484 | 3765 |
| 10010 | 2593 | 51 | 3824 |
| 10025 | 2768 | 2322 | 3435 |
| 10026 | 216 | 2840 | 211 |
| 10039 | 297 | 3016 | 345 |
| | 4031058127_B | 4031058211_B | |
| 10008 | 3558 | 6502 | |
| 10010 | 3528 | 6154 | |
| 10025 | 3471 | 3608 | |
| 10026 | 164 | 188 | |
| 10039 | 361 | 380 | |

```
> pd = pData(RG)
> pd[1:5, ]
```

| | HapMap.Name | Gender | Plate |
|--------------|-------------|-----------------|---------------|
| 4030186347_A | NA06991 | Female | WG1000442-DNA |
| 4030186263_B | NA07000 | Female | WG1000442-DNA |
| 4019585415_B | NA10859 | Female | WG1000453-DNA |
| 4031058127_B | NA11882 | Female | WG1000453-DNA |
| 4031058211_B | NA06993 | Male | WG1000447-DNA |
| | Well | SentrixPosition | |
| 4030186347_A | E11 | 4030186347_A | |
| 4030186263_B | D08 | 4030186263_B | |
| 4019585415_B | B02 | 4019585415_B | |
| 4031058127_B | D08 | 4031058127_B | |
| 4031058211_B | D11 | 4031058211_B | |

```
> scandatetime = strptime(protocolData(RG)[["ScanDate"]],
+   "%m/%d/%Y %H:%M:%S %p")
> datescanned = substr(scandatetime, 1,
+   10)
> scanbatch = factor(datescanned)
> levels(scanbatch) = 1:16
> scanbatch = as.numeric(scanbatch)
```

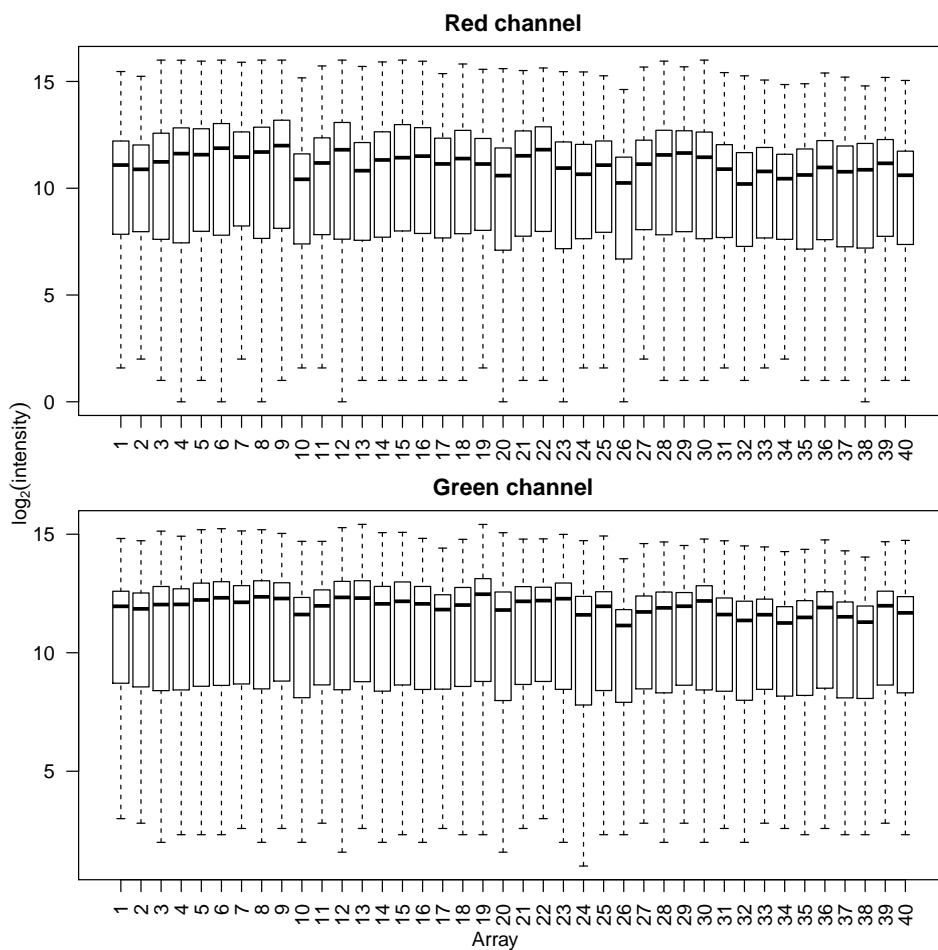
Plots of the summarised data can be easily generated to check for arrays with poor signal.

```
> par(mfrow = c(2, 1), mai = c(0.4, 0.4,
+   0.4, 0.1), oma = c(1, 1, 0, 0))
> boxplot(log2(exprs(channel(RG, "R"))),
```

```

+   xlab = "Array", ylab = "", names = 1:40,
+   main = "Red channel", outline = FALSE,
+   las = 2)
> boxplot(log2(exprs(channel(RG, "G"))),
+   xlab = "Array", ylab = "", names = 1:40,
+   main = "Green channel", outline = FALSE,
+   las = 2)
> mtext(expression(log[2](intensity)), side = 2,
+   outer = TRUE)
> mtext("Array", side = 1, outer = TRUE)

```



3 Genotyping

Next we use the function `crlmmIllumina` which performs preprocessing followed by genotyping using the CRLMM algorithm.

```
> crlmmResult = crlmmIllumina(RG = RG, cdfName = "human370v1c",
+   sns = pData(RG)$ID, returnParams = TRUE)
```

This analysis took 18 minutes to complete and peak memory usage was 2.5 GB on our system. The output stored in `crlmmResult` is a *SnpSet* object.

```
> class(crlmmResult)
```

```
[1] "SnpSet"
attr("package")
[1] "Biobase"
```

```
> dim(crlmmResult)
```

```
Features  Samples
 346451      40
```

```
> slotNames(crlmmResult)
```

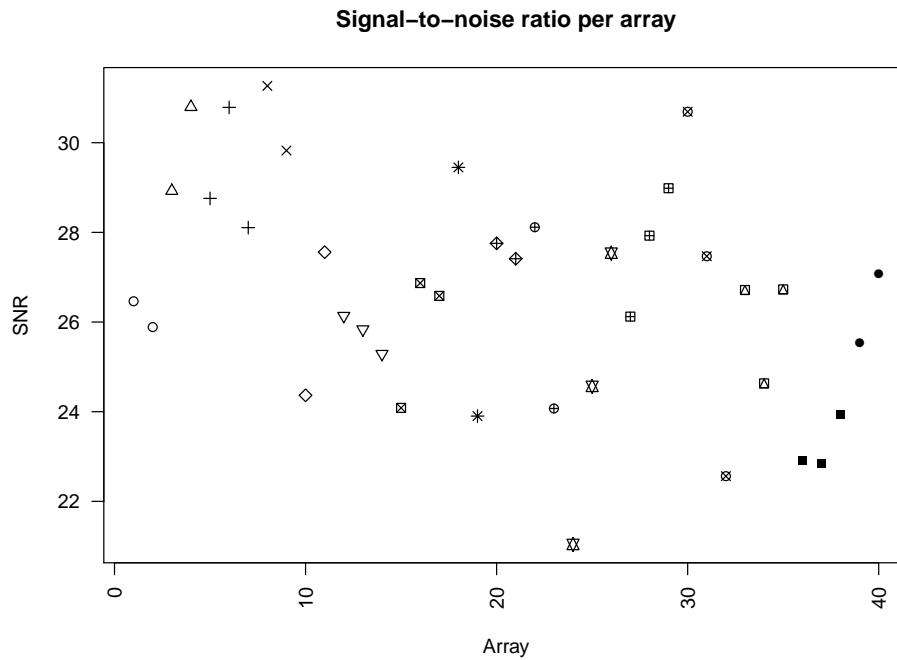
```
[1] "assayData"      "phenoData"
[3] "featureData"    "experimentData"
[5] "annotation"     "protocolData"
[7] ".__classVersion__"
```

```
> calls(crlmmResult)[1:10, 1:5]
```

```
      1 2 3 4 5
rs12354060 1 1 3 3 3
rs6650104  1 1 1 1 1
rs12184279 1 1 1 1 1
rs12564807 1 1 1 1 1
rs3115860  2 1 1 2 2
rs3115850  1 2 2 1 1
rs7515489  3 3 1 1 1
rs12124819 1 2 2 1 1
rs17160939 1 1 1 1 1
rs12086311 3 3 3 3 3
```

Plotting the *SNR* reveals no obvious batch effects in this data set (different symbols are used for arrays scanned on different days).

```
> plot(crlmmResult[["SNR"]], pch = scanbatch,
+   xlab = "Array", ylab = "SNR", main = "Signal-to-noise ratio per array",
+   las = 2)
```



4 System information

This analysis was carried out on a linux machine with 32GB of RAM using the following packages:

```
> sessionInfo()
```

```
R version 2.12.0 alpha (2010-09-21 r52960)
Platform: x86_64-unknown-linux-gnu (64-bit)
```

```
locale:
```

```
[1] LC_CTYPE=en_US.iso885915
[2] LC_NUMERIC=C
[3] LC_TIME=en_US.iso885915
[4] LC_COLLATE=en_US.iso885915
[5] LC_MONETARY=C
[6] LC_MESSAGES=en_US.iso885915
[7] LC_PAPER=en_US.iso885915
[8] LC_NAME=C
[9] LC_ADDRESS=C
[10] LC_TELEPHONE=C
[11] LC_MEASUREMENT=en_US.iso885915
```

[12] LC_IDENTIFICATION=C

attached base packages:

[1] tools stats graphics grDevices
[5] utils datasets methods base

other attached packages:

[1] human370v1cCrlmm_1.0.1 hapmap370k_1.0.0
[3] crlmm_1.7.15 oligoClasses_1.11.8
[5] Biobase_2.9.1 weaver_1.15.0
[7] codetools_0.2-2 digest_0.4.2

loaded via a namespace (and not attached):

[1] affyio_1.17.4 annotate_1.27.1
[3] AnnotationDbi_1.11.5 Biostrings_2.17.47
[5] bit_1.1-4 DBI_0.2-5
[7] ellipse_0.3-5 ff_2.1-2
[9] genefilter_1.31.2 IRanges_1.7.34
[11] mvtnorm_0.9-92 preprocessCore_1.11.0
[13] RSQLite_0.9-2 splines_2.12.0
[15] survival_2.35-8 xtable_1.5-6