crlmm to downstream data analysis

VJ Carey

April 7, 2009

1 Running CRLMM on a nontrivial set of CEL files

We work with the 90 CEU samples hybridized to Affy 6.0 chips, which are assumed to be in the current directory. First, we identify the files and run `crlmm`. The results will be saved to the variable `crlmmResult`.

```
> library(crlmm)
> celFiles <- list.celfiles()
> celFiles[1:4]

[1] "NA06985_GW6_C.CEL" "NA06991_GW6_C.CEL" "NA06993_GW6_C.CEL"
[4] "NA06994_GW6_C.CEL"

> if (!exists("crlmmResult")) {
+     if (file.exists("crlmmResult.rda"))
+         load("crlmmResult.rda")
+     else {
+         crlmmResult <- crlmm(celFiles)
+         save(crlmmResult, file = "crlmmResult.rda")
+     }
+ }
```

This is currently a *SnpSet* object.

```
> class(crlmmResult)

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

2 Adding information to a *SnpSet*

We will use the *GGdata* package to obtain extra information on the samples. This will be later used when building an *eSet* extension to store the genotyping results.

```
> library(GGdata)
> if (!exists("hmceuB36")) data(hmceuB36)
> pd <- phenoData(hmceuB36)
> ggn <- sampleNames(pd)
> preSN <- sampleNames(crlmmResult)
> simpSN <- gsub("_.*", "", preSN)
> if (!all.equal(simpSN, ggn)) stop("align GGdata phenoData with crlmmResult read")
```

The additional information obtained from *GGdata* can be easily combined to what is already available on `crlmmResult`.

```
> sampleNames(crlmmResult) <- simpSN
> phenoData(crlmmResult) <- combine(pd, phenoData(crlmmResult))
> dim(calls(crlmmResult))

[1] 906600     90

> dim(confs(crlmmResult))

[1] 906600     90

> calls(crlmmResult)[1:10, 1:2]

              NA06985 NA06991
SNP_A-2131660       2       2
SNP_A-1967418       3       3
SNP_A-1969580       3       3
SNP_A-4263484       2       1
SNP_A-1978185       1       1
SNP_A-4264431       1       1
SNP_A-1980898       3       3
SNP_A-1983139       1       1
SNP_A-4265735       2       2
SNP_A-1995832       2       3

> confs(crlmmResult)[1:10, 1:2]

              NA06985 NA06991
SNP_A-2131660   10561   11574
SNP_A-1967418   12517   14866
```

```
SNP_A-1969580    7632    7606
SNP_A-4263484   15621   20059
SNP_A-1978185   14030   18021
SNP_A-4264431   17792   17235
SNP_A-1980898    7640    7642
SNP_A-1983139   14127    8974
SNP_A-4265735    8976    9153
SNP_A-1995832   10336   17920
```

3 Coercing to snp.matrix as a prelude to a GWAS

```
> library(snpMatrix)
> crlmmSM <- as(t(calls(crlmmResult)) - 1, "snp.matrix")
> crlmmSM

A snp.matrix with  90 rows and  906600 columns
Row names:  NA06985 ... NA12892
Col names:  SNP_A-2131660 ... SNP_A-8574011
```

4 Conducting a GWAS

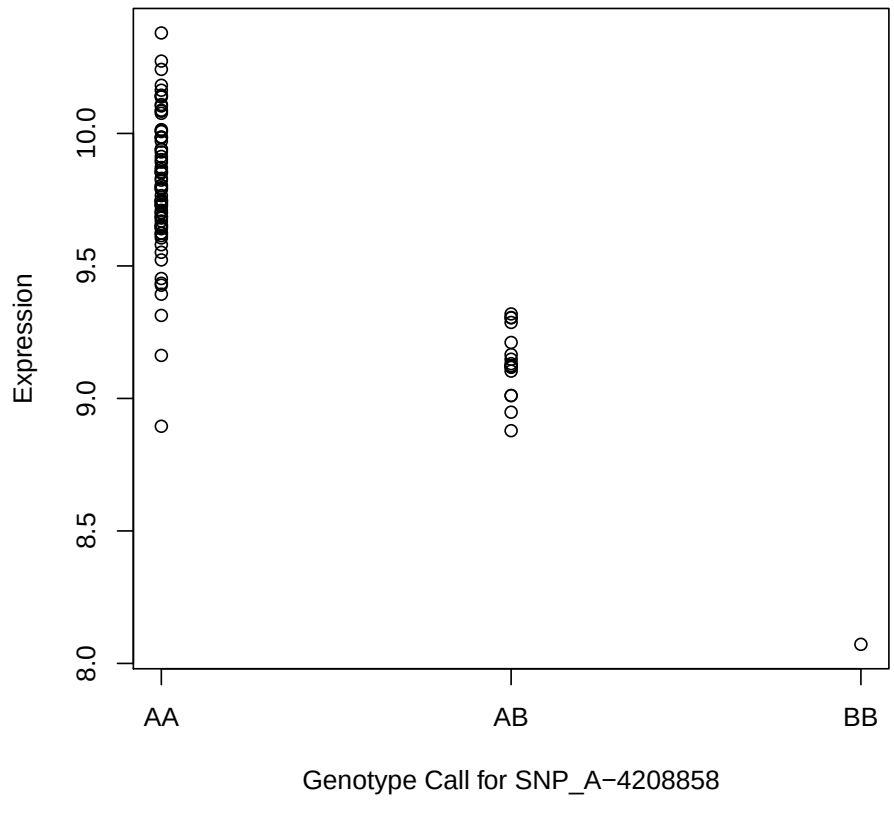
We want to find SNP for which rare allele count is predictive of expression of CPNE1. We will use expression data available from GGdata. This is a very naive analysis.

```
> library(illuminaHumanv1.db)
> rmm <- revmap(illuminaHumanv1SYMBOL)
> mypr <- get("CPNE1", rmm)
> ex <- as.numeric(exprs(hmceuB36)[mypr[1], ])
> subjdata <- pData(hmceuB36)
> subjdata[["ex"]] <- ex
> gwas <- snp.rhs.tests(ex ~ male, data = subjdata, snp.data = crlmmSM,
+     family = "gaussian")
> ok <- which(p.value(gwas) < 1e-10)
> gwas[ok, ]

             Chi.squared Df      p.value
SNP_A-4208858    53.62528  1 2.426168e-13
SNP_A-2022241    42.67385  1 6.467116e-11
SNP_A-2039695    46.23796  1 1.047283e-11
SNP_A-2047882    48.35134  1 3.563006e-12
SNP_A-2108011    46.23796  1 1.047283e-11
SNP_A-2125946    41.91646  1 9.525721e-11
```

```
SNP_A-2171015    46.23796  1 1.047283e-11
SNP_A-2184991    44.04283  1 3.212681e-11
SNP_A-2216659    48.35134  1 3.563006e-12
SNP_A-2220183    53.62528  1 2.426168e-13
SNP_A-2231089    46.23796  1 1.047283e-11
SNP_A-2231469    53.62528  1 2.426168e-13
SNP_A-2275065    53.62528  1 2.426168e-13
SNP_A-1806237    48.35134  1 3.563006e-12
SNP_A-1912540    46.23796  1 1.047283e-11
SNP_A-1921183    48.35134  1 3.563006e-12
SNP_A-8611599    51.91596  1 5.792757e-13
SNP_A-8699268    51.91596  1 5.792757e-13
```

```
> plot(ex ~ calls(crlmmResult)["SNP_A-4208858", ], xlab = "Genotype Call for SNP_A-42
+     ylab = "Expression", xaxt = "n")
> axis(1, at = 1:3, labels = c("AA", "AB", "BB"))
```



5 Session Info

This vignette was created using the following packages:

```
> sessionInfo()

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

locale:
LC_CTYPE=en_US.UTF-8;LC_NUMERIC=C;LC_TIME=en_US.UTF-8;LC_COLLATE=en_US.UTF-8;LC_MONETAR

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

other attached packages:
 [1] GGdata_0.99.3           illuminaHumanv1.db_1.1.3 GGBase_3.2.11
 [4] RSQLite_0.7-1           DBI_0.2-4                snpMatrix_1.7.5
 [7] survival_2.34-1         GSEABase_1.5.2           graph_1.21.4
[10] annotate_1.21.3         AnnotationDbi_1.5.15     Biobase_2.3.11
[13] crlmm_1.0.78

loaded via a namespace (and not attached):
[1] affyio_1.11.3        cluster_1.11.12      genefilter_1.23.2
[4] mvtnorm_0.9-4        preprocessCore_1.5.3 tools_2.9.0
[7] XML_1.99-0           xtable_1.5-4
```