Exploring the MAQC data with Bioconductor

VJ Carey

April 8, 2011

1 Introduction

See the Sept 2006 issue of Nature Biotechnology for several articles about the MAQC initiative. The *MAQCsubset* package includes excerpts from the data publised at GEO GSE5350.

```
> library(MAQCsubset)
> data(afxsubRMAES)
> afxsubRMAES

ExpressionSet (storageMode: lockedEnvironment)
assayData: 54675 features, 24 samples
  element names: exprs
protocolData: none
phenoData
  sampleNames: AFX_1_A1.CEL AFX_1_A2.CEL ... AFX_3_D2.CEL (24 total)
  varLabels: site samp repl pctBrain
  varMetadata: labelDescription
featureData: none
experimentData: use 'experimentData(object)'
  pubMedIds: 16964226
Annotation: hgu133plus2

> pd = pData(afxsubRMAES)
> table(pd$site, pd$samp)

    A B C D
  1 2 2 2 2
  2 2 2 2 2
  3 2 2 2 2
```

Samples labeled "A" have 100% stratagene universal human RNA, while samples labeled "B" have 100% Ambion human brain RNA. Samples labeled C have .75A+.25B, and samples labeled D have .75B+.25A.

2 The proboscis plot

For Figure 2 of Shippy et al., *Using RNA sample titrations* (Nat Biotech, 24(9):1123-1131, Sep 2006), genes differentially expressed between samples A and B using t tests at $p = 0.001$ are identified. If, for such genes, the A samples are up-regulated relative to the B samples, then a self-consistent monotone titration (SCMT) is declared if the C samples for such genes are up-regulated relative to the D samples. For genes upregulated on B samples relative to A samples, then SCMT occurs if the D samples are up-regulated relative to the C samples.

Figure 2 of Shippy et al. plots the proportion of genes exhibiting SCMT against the intensity ratios (A/B or B/A as appropriate). These plots, formed for each manufacturer/normalization combination and for each site, have the appearance of long pointy noses and are thus called proboscis plots. The following code computes the necessary quantities:

```
> proboscis = function(es, site = 1, ABp = 0.001, CDp = 0.01, mmrad = 100) {
+     require(genefilter)
+     mcall = match.call()
+     mm = function(x, rad) {
+         start = ceiling(rad/2)
+         stop = floor(length(x) - (rad/2))
+         sapply(start:stop, function(i) mean(x[(i - floor(rad/2)):(i +
+             floor(rad/2))]))
+     }
+     ess = es[, es$site == site]
+     essab = ess[, ess$samp %in% c("A", "B")]
+     essab$samp = factor(essab$samp)
+     esscd = ess[, ess$samp %in% c("C", "D")]
+     esscd$samp = factor(esscd$samp)
+     tt = rowttests(exprs(essab), essab$samp)
+     L = which(tt$p < ABp & tt$dm < 0)
+     R = which(tt$p < ABp & tt$dm > 0)
+     ttcd = rowttests(exprs(esscd), esscd$samp)
+     ABL = tt$dm[L]
+     CDL = ttcd$dm[L]
+     ABR = tt$dm[R]
+     CDR = ttcd$dm[R]
+     NN = list(ttab = tt, ttcd = ttcd, ABL = sort(ABL), cdokL = 1 *
+         (CDL < 0)[order(ABL)], ABR = sort(ABR), dcokR = 1 * (CDR >
+         0)[order(ABR)])
+     `A-B` = c(ONR <- mm(NN$ABL, mmrad), mm(NN$ABR, mmrad))
+     `P(SCMT|A-B)` = c(mm(NN$cdokL, mmrad), mm(NN$dcokR, mmrad))
+     new("proboStruct", call = mcall, list(`A-B` = `A-B`, `P(SCMT|A-B)` = `P(SCMT|A-
```

```
+                  leftinds = 1:length(ONR)))
+ }
> NN1 = proboscis(afxsubRMAES)
> NN2 = proboscis(afxsubRMAES, site = 2)
> NN3 = proboscis(afxsubRMAES, site = 3)
```
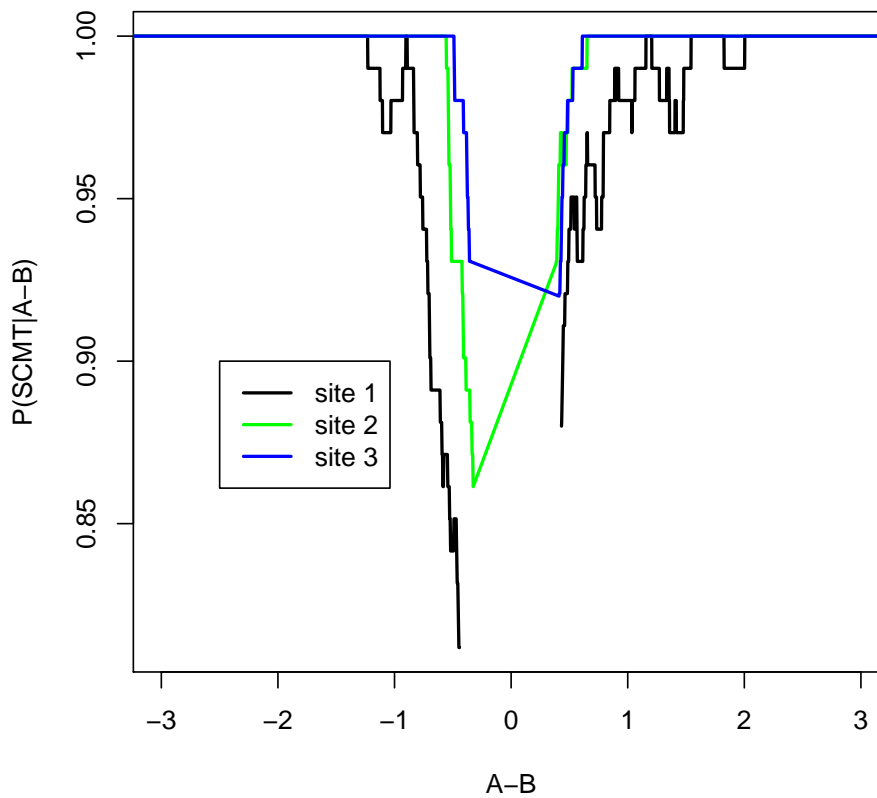
There are simple graphical methods:

```
> plot(NN1, lwd = 2)
> lines(NN2[[1]], NN2[[2]], col = "green", lwd = 2)
> lines(NN3[[1]], NN3[[2]], col = "blue", lwd = 2)
> legend(-2.5, 0.9, lty = 1, lwd = 2, legend = c("site 1", "site 2",
+     "site 3"), col = c("black", "green", "blue"))
```


These do not look exactly like the plots in the Shippy paper, presumably because only two replicates per site are in use in this display.