

The use of within-array replicate spots for assessing differential expression in microarray experiments: supplementary material

Gordon Smyth

6 April 2006

1 Introduction

This document documents the R analysis used in the article:

Smyth, G. K., Michaud, J., and Scott, H. (2005). The use of within-array replicate spots for assessing differential expression in microarray experiments. *Bioinformatics* **21**(9), 2067–2075.

The reader should obtain the file <http://bioinf.wehi.edu.au/folders/gordon/dupcor/dupcor-data.zip> and should unpack it to a convenient directory. In R, set the working directory to the directory you unpacked the files to. On my PC, this is

```
> setwd("C:/Gordon/MicrArra/dupcor-data")
```

although your directory will be different. You should see the following files:

```
> dir()
[1] "a_h8ka010702e.gal" "analysis.pdf"      "analysis.tex"
[4] "plots"              "spot"              "spottypes-sc.txt"
[7] "targets-sc.txt"
```

The data files are contained in the directory `spot`.

2 Read and normalize

```
> library(limma)
> library(statmod)
```

```
Attaching package: 'statmod'
```

```
The following object(s) are masked from package:limma :
```

```
matvec vecmat
```

```

> sessionInfo()
R version 2.2.1, 2005-12-20, i386-pc-mingw32

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

other attached packages:
  statmod      limma
  "1.2.4"      "2.4.13"
> targets <- readTargets("targets-sc.txt")
> RG <- read.maimages(targets$SlideNumber,
+   source="spot",path="spot",ext="spot",wt.fun=wtarea(165))
Read spot/2150.spot
Read spot/2151.spot
Read spot/2153.spot
Read spot/2154.spot
Read spot/2155.spot
Read spot/2156.spot
Read spot/2157.spot
Read spot/2158.spot
Read spot/2159.spot
Read spot/2160.spot
Read spot/2161.spot
Read spot/2162.spot
Read spot/2163.spot
Read spot/2164.spot
Read spot/2165.spot
Read spot/2166.spot
Read spot/2167.spot
Read spot/2173.spot
Read spot/2174.spot
Read spot/2175.spot
Read spot/2176.spot
Read spot/2177.spot
Read spot/2178.spot
Read spot/2179.spot
Read spot/2180.spot
Read spot/2181.spot
> RG$genes <- readGAL("a_h8ka010702e.gal")
> spottypes <- readSpotTypes("spottypes-sc.txt")
> RG$genes$Status <- controlStatus(spottypes, RG)
Matching patterns for: Name
Found 19200 Other
Found 24 Calib1
Found 24 Calib2
Found 24 Calib3
Found 24 Calib4
Found 24 Calib5

```

```

Found 24 Calib6
Found 24 Calib7
Found 24 Calib8
Found 24 Calib9
Found 24 Calib10
Found 24 3DL
Found 24 3UL
Found 24 3DH
Found 24 3UH
Found 24 10DL
Found 24 10UL
Found 24 10DH
Found 24 10UH
Setting attributes: values col cex
> RG$printer <- list(ndups=2,spacing=1)

```

We now subset to the Scorecard probes and apply global loess normalization.

```

> i <- RG$genes$Status != "Other"
> RGsc <- RG[i,]
> MAsc <- normalizeWithinArrays(RGsc, method="loess")

```

3 All 26 arrays

Fit a model using duplicate-correlations to all 26 arrays:

```

> MAsc$design <- modelMatrix(targets,ref="Reference")
Found unique target names:
Reference Test
> dupfit <- duplicateCorrelation(MAsc)
> dupfit$consensus
[1] 0.9217067
> fit <- lmFit(MAsc,cor=dupfit$consensus)
> fit <- eBayes(fit)

```

Fit a model after averaging the duplicate spots:

```

> MAscM <- avedups(MAsc,ndups=2,spacing=1,weights=NULL)
> fitM <- lmFit(MAscM)
> fitM <- eBayes(fitM)

```

Box-plots in color for ordinary t -statistics:

```

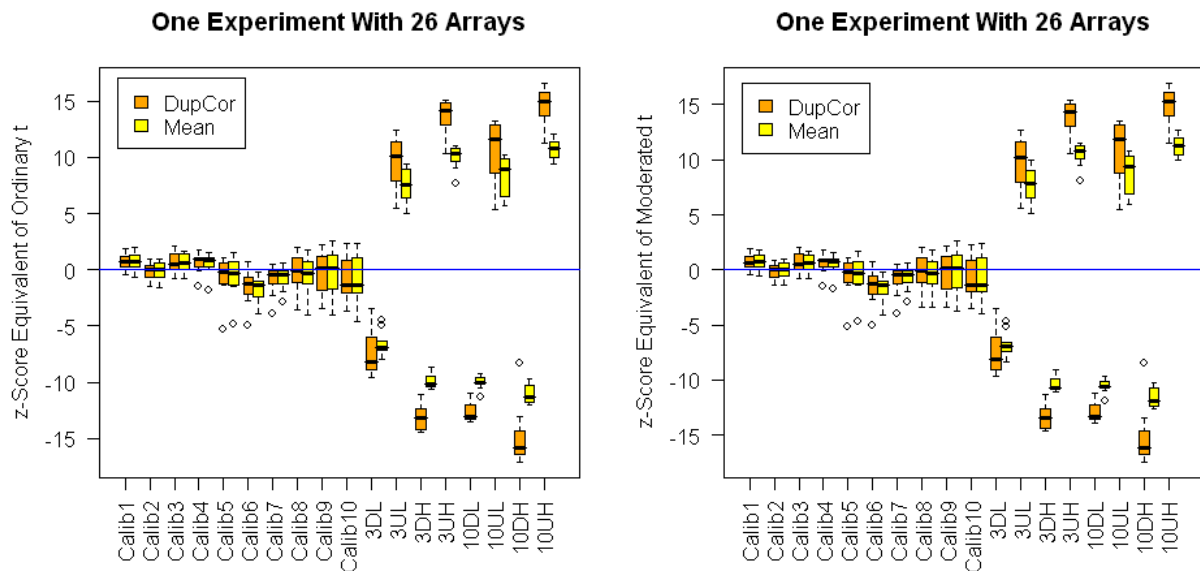
> f <- factor(fit$genes$Status,levels=spottypes$SpotType[-1])
> zordt <- zscoreT(fit$coef/fit$stdev/fit$sigma,df=fit$df.residual)
> zordtm <- zscoreT(fitM$coef/fitM$stdev/fitM$sigma,df=fitM$df.residual)
> png(file="plots/all.png",height=480,width=960)
> par(mfrow=c(1,2))
> boxplot(zordt~f,boxwex=0.4, at=1:18-0.2,col="orange",las=2,
+ ylab="z-Score Equivalent of Ordinary t",main="One Experiment With 26 Arrays")

```

```
> boxplot(zordtm~f,boxwex=0.4, at=1:18+0.2,col="yellow",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,17,c("DupCor","Mean"),fill=c("orange","yellow"))
> abline(h=0,col="blue")
```

and for moderated t -statistics:

```
> zmodt <- zscoreT(fit$t,df=fit$df.prior+fit$df.residual)
> zmodtm <- zscoreT(fitm$t,df=fitm$df.prior+fitm$df.residual)
> boxplot(zmodt~f,boxwex=0.4, at=1:18-0.2,col="orange",las=2,
+ ylab="z-Score Equivalent of Moderated t",main="One Experiment With 26 Arrays")
> boxplot(zmodtm~f,boxwex=0.4, at=1:18+0.2,col="yellow",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,17,c("DupCor","Mean"),fill=c("orange","yellow"))
> abline(h=0,col="blue")
> dev.off()
null device
1
```



The same in black and white:

```
> par(mfrow=c(1,2))
> boxplot(zordt~f,boxwex=0.4, at=1:18-0.2,col="gray",las=2,
+ ylab="z-Score Equivalent of Ordinary t",main="One Experiment With 26 Arrays")
> boxplot(zordtm~f,boxwex=0.4, at=1:18+0.2,col="white",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,17,c("DupCor","Mean"),fill=c("gray","white"))
> abline(h=0,lty=2)
>
> boxplot(zmodt~f,boxwex=0.4, at=1:18-0.2,col="gray",las=2,
+ ylab="z-Score Equivalent of Moderated t",main="One Experiment With 26 Arrays")
> boxplot(zmodtm~f,boxwex=0.4, at=1:18+0.2,col="white",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,17,c("DupCor","Mean"),fill=c("gray","white"))
> abline(h=0,lty=2)
```

4 Arrays in pairs of 2

Fit duplicate-correlation models to each pair of two arrays:

```
> dupby2 <- list()
> for (j in 1:13) {
+   dupfit <- duplicateCorrelation(MAsc[(2*j-1):(2*j)])
+   dupby2[[j]] <- eBayes(lmFit(MAsc[(2*j-1):(2*j)],cor=dupfit$consensus))
+ }
```

Warning messages:

```
1: Too much damping - convergence tolerance not achievable in: glmGamFit(dx, dy, start = st
2: Too much damping - convergence tolerance not achievable in: glmGamFit(dx, dy, start = st
3: Too much damping - convergence tolerance not achievable in: glmGamFit(dx, dy, start = st
4: Too much damping - convergence tolerance not achievable in: glmGamFit(dx, dy, start = st
5: Too much damping - convergence tolerance not achievable in: glmGamFit(dx, dy, start = st
6: Too much damping - convergence tolerance not achievable in: glmGamFit(dx, dy, start = st
```

The warnings can be ignored.

Fit models with duplicates averaged:

```
> meanby2 <- list()
> for (j in 1:13) {
+   meanby2[[j]] <- eBayes(lmFit(MAscM[(2*j-1):(2*j)]))
+ }
```

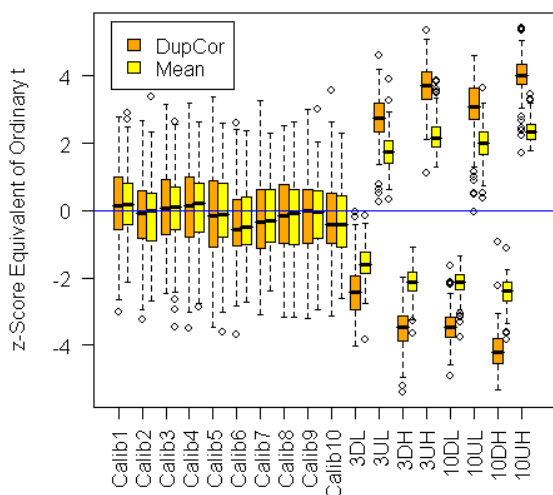
Boxplots for ordinary t

```
> zordtby2 <- matrix(0,216,13)
> for (j in 1:13) zordtby2[,j] <-
+   zscoreT(dupby2[[j]]$coef/dupby2[[j]]$stdev/dupby2[[j]]$sigma,df=3)
> zordtmby2 <- matrix(0,216,13)
> for (j in 1:13) zordtmby2[,j] <-
+   zscoreT(meanby2[[j]]$coef/meanby2[[j]]$stdev/meanby2[[j]]$sigma,df=1)
> png(file="plots/by2.png",height=480,width=960)
> par(mfrow=c(1,2))
> boxplot(zordtby2~f,boxwex=0.4, at=1:18-0.2,col="orange",las=2,
+   ylab="z-Score Equivalent of Ordinary t",main="13 Experiments With 2 Arrays")
> boxplot(zordtmby2~f,boxwex=0.4, at=1:18+0.2,col="yellow",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,5.6,c("DupCor","Mean"),fill=c("orange","yellow"))
> abline(h=0,col="blue")
```

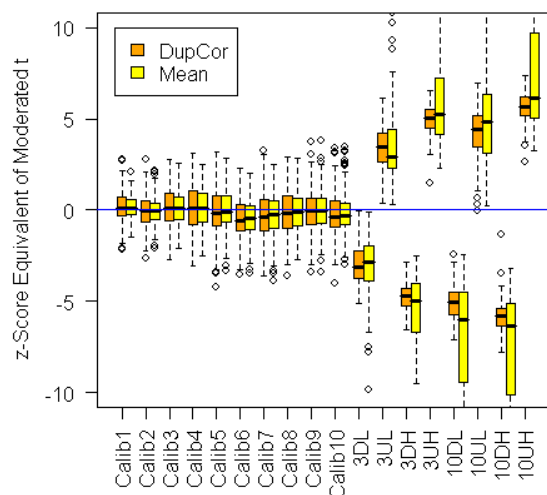
and for moderated t

```
> zmodtby2 <- matrix(0,216,13)
> for (j in 1:13) zmodtby2[,j] <- zscoreT(dupby2[[j]]$t,df=dupby2[[j]]$df.prior+3)
> zmodtmby2 <- matrix(0,216,13)
> for (j in 1:13) zmodtmby2[,j] <- zscoreT(meanby2[[j]]$t,df=meanby2[[j]]$df.prior+1)
> boxplot(zmodtby2~f,boxwex=0.4, at=1:18-0.2,col="orange",las=2,ylim=c(-10,10),
+   ylab="z-Score Equivalent of Moderated t",main="13 Experiments With 2 Arrays")
> boxplot(zmodtmby2~f,boxwex=0.4, at=1:18+0.2,col="yellow",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,10,c("DupCor","Mean"),fill=c("orange","yellow"))
> abline(h=0,col="blue")
> dev.off()
```

13 Experiments With 2 Arrays



13 Experiments With 2 Arrays



Now the same in black and white:

```
> boxplot(zordtby2~f,boxwex=0.4, at=1:18-0.2,col="gray",las=2,
+ ylab="z-Score Equivalent of Ordinary t",main="13 Experiments With 2 Arrays")
> boxplot(zordtmb2~f,boxwex=0.4, at=1:18+0.2,col="white",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,5.6,c("DupCor","Mean"),fill=c("gray","white"))
> abline(h=0,lty=2)
>
> boxplot(zmodtby2~f,boxwex=0.4, at=1:18-0.2,col="gray",las=2,ylim=c(-10,10),
+ ylab="z-Score Equivalent of Moderated t",main="13 Experiments With 2 Arrays")
> boxplot(zmodtmb2~f,boxwex=0.4, at=1:18+0.2,col="white",add=TRUE,yaxt="n",xaxt="n")
> legend(0.5,10,c("DupCor","Mean"),fill=c("gray","white"))
> abline(h=0,lty=2)
```