Identifying a field-friendly RNA preservation protocol

Anders

13 September 2014

The goal

Our objective is to examine our ability to differentiate among RNA samples preserved under distinct conditions in order to evaluate our ability to recover gene expression data from them.

We want to test three field-friendly approaches, and compare it to the gold-standard approach of triazol preservation (which is Phenol-based solvent, and thus a hazard to handle and difficult to transport).

Model assumptions

RNA-seq data usually consists of number of reads that map to a particular gene. The counts can be modelled by a Poisson Distribution or a Negative Binomial Distribution. Often times the data violates the assumption of equal mean and variance of the Poisson, and that is why the more flexible Negative Binomial is used.

Here, we will use a Negative Binomial, parametrized by mean (mu) and size (dispersion parameter) to model the mean number of reads mapping to a particular gene. Under this parametrization, prob = size/(size+mu) and the variance is mu + mu^2/size.

We will model four scenarios for the purposes of this study. In the first, we will examine the hypothesis that the different preservation methods do not produce significant differences in downstream analysis. In the second, the different preservation methods will vary in their performance relative to the control. But, we will assume that the effect of the preservation treatment is equal across all genes (i.e., there is a uniform effect of lowering or increasing the read count, either by poorly preserving the RNA relative to the control, or performing better than the control).

The third and fourth scenarios assume that one of the treatments leads to uneven preservation effects across different genes. In the third scenario, we assume that some genes are affected differently by the preservation method because of their GC content. This means that the same genes are affected in a consistent manner across replicates (if the gene is poorly preserved in 1 sample it will be equally poorly affected in another sample). In the fourth scenario, we extend the third scenario by assuming that the effect is random across replicates. Thus, read count at a gene might be poorly preserved in one replicate but well preserved in another.

We will analyze the data using the `edgeR` package in `R`, which is designed to estimate *differential gene expression* across two or more samples. Our hypothesis is that the different treatments should affect the total amount of RNA we can recover from each sample, and thus should mimick results from a differential gene expression study.

Ideally, we would have a preservation method that is cheap, safe, and easy to transport. But, that it also performs as well or better than the control method (which involves dangerous goods, thus creating a potentially unnecessary hazard as well as complications for transport), and is consistent in how it preserves RNA across genes and samples.

Data simulation

```
#common variables
nGenes = 5000
mean = 100
size = 0.5

std1 = sqrt(20)
std2 = 1

treats = 4
bioReps = 8
techReps = 3

deviations = c(0,-80,-20,50)

mainFac = factor(
  rep(c("control","treatA","treatB","treatC"),
      each=techReps*bioReps))
mainFac = relevel(mainFac,"control")
bioRepFac = factor(rep(
  rep(paste("Bio",1:bioReps,sep=""),
                          each=techReps),treats))

design <- model.matrix(~mainFac)
```

In each case, we will model 5000 genes (approximately the number of genes in *Plasmodium falciparum*), and 8 biological replicates, with 3 technical replicates of each biological replicate. This will produce a total of 24 measurements per treatment, and 96 for the whole experiment.

Scenario 1

For the first scenario, 5000 values were drawn from a NB(mu=100,size=0.5) distribution. These were used as the expected read count for each of the sampled genes. The mean read count for each gene for each biological replicate under each treatment were then be drawn from a Normal distribution with mean equal to the value drawn above and standard deviation 10 truncated at 0. Finally, the observed read count for each technical replicate were then drawn from a Normal distribution with mean as drawn for the biological replicate and standard deviation 1, again truncated at 0.

```
set.seed(seed = 1234)

meanGeneCounts = rnbinom(n = nGenes, size = size, mu = mean)

meanBiolTreats = matrix(0,nrow = nGenes, ncol = treats*bioReps)

for(i in 1:nGenes){
  meanBiolTreats[i,] = rnorm(n = treats*bioReps,
                             mean = meanGeneCounts[i],
                             sd = std1)
}

readCounts = matrix(0,
                    ncol = treats*bioReps*techReps,
                    nrow = nGenes)
```

```
for(g in 1:nGenes){
  means = meanBiolTreats[g,]
  for(t in 1:(treats*bioReps)){
    nReads = floor(rnorm(n = techReps, mean = means[t], sd = std2))
    nReads[nReads<0] = 0
    p = 3*(t-1) + 1
    readCounts[g,p:(p+techReps-1)] <- nReads
  }
}


keep=rowSums(cpm(readCounts)>2)==96

readCountsKeep = readCounts[keep,]

nrow(readCountsKeep)
```

```
## [1] 3634
```

```
sc1_y = DGEList(counts = readCountsKeep,group = mainFac)

#plotMDS(y,labels = mainFac,col=rep(c("black","navyblue","gold","darkgreen"),each=24))


sc1_logFC <- predFC(sc1_y,design,prior.count=1,dispersion=0.05)

cor(sc1_logFC)
```

```
##               (Intercept) mainFactreatA mainFactreatB mainFactreatC
## (Intercept)    1.00000000   -0.02393681    -0.0214690   -0.03553131
## mainFactreatA -0.02393681    1.00000000     0.5306645    0.50245388
## mainFactreatB -0.02146900    0.53066449     1.0000000    0.52501557
## mainFactreatC -0.03553131    0.50245388     0.5250156    1.00000000
```

```
sc1_y  <- estimateGLMCommonDisp(sc1_y , design, verbose=TRUE)
```

```
## Disp = 0 , BCV = 1e-04
```

```
sc1_y  <- estimateGLMTrendedDisp(sc1_y , design)
```

```
## Loading required package: splines
```

```
sc1_y  <- estimateGLMTagwiseDisp(sc1_y , design)
#plotBCV(sc1_y)

sc1_fit <- glmFit(sc1_y, design)

sc1_lrt <- glmLRT(sc1_fit)

#topTags(sc1_lrt)
```

```r
sc1_FDR <- p.adjust(sc1_lrt$table$PValue, method="BH")
sum(sc1_FDR < 0.05)
```

```
## [1] 25
```

```r
top <- rownames(topTags(sc1_lrt))
#cpm(sc1_y)[top,]
summary(sc1_dt <- decideTestsDGE(sc1_lrt))
```

```
##    [,1]
## -1   14
## 0  3609
## 1    11
```

```r
sc1_isDE <- as.logical(sc1_dt)
sc1_DEnames <- rownames(sc1_y)[sc1_isDE]
#plotSmear(sc1_lrt, de.tags=DEnames,ylim=c(-2,2))
#abline(h=c(-1,1), col="blue")
```

Scenario 2

For the second scenario, the same approach as above was applied. However, the values drawn from the NB distribution were subttracted by 80 and 20 for the treatments A and B (representing preservation methods that perform poorly relative to the control), and added by 50 for treatment C (representing a preservation method that performs better than the control).

```r
set.seed(seed = 7878)


meanGeneCounts = rnbinom(n = nGenes, size = size, mu = mean)

meanBiolTreats = matrix(0,nrow = nGenes, ncol = treats*bioReps)

for(i in 1:nGenes){
  meanBiolTreats[i,] = rnorm(n = treats*bioReps,
                             mean = meanGeneCounts[i],
                             sd = std1)
}

readCounts = matrix(0,ncol = treats*bioReps*techReps,
                    nrow = nGenes)

for(g in 1:nGenes){
  means = meanBiolTreats[g,]
  for(t in 1:treats){
    diff = deviations[t]
    for(b in 1:bioReps){
      pos = bioReps*(t-1)+b
      nReads = floor(rnorm(n = techReps, mean = means[pos], sd = std2))
      nReads = nReads + diff
```

```
      nReads[nReads<0] = 0
      p = 3*(pos-1) + 1
      readCounts[g,p:(p+techReps-1)] <- nReads
    }
  }
}


keep=rowSums(cpm(readCounts)>2)==96

readCountsKeep = readCounts[keep,]
nrow(readCountsKeep)
```

```
## [1] 1759
```

```
sc2_y = DGEList(counts = readCountsKeep,group = mainFac)

#plotMDS(y,labels = mainFac,col=rep(c("black","navyblue","gold","darkgreen"),each=24))


sc2_logFC <- predFC(sc2_y,design,prior.count=1,dispersion=0.05)

cor(sc2_logFC)
```

```
##               (Intercept) mainFactreatA mainFactreatB mainFactreatC
## (Intercept)     1.0000000     0.8647053     0.9203601    -0.9665118
## mainFactreatA   0.8647053     1.0000000     0.9448840    -0.9390671
## mainFactreatB   0.9203601     0.9448840     1.0000000    -0.9483058
## mainFactreatC  -0.9665118    -0.9390671    -0.9483058     1.0000000
```

```
sc2_y  <- estimateGLMCommonDisp(sc2_y , design, verbose=TRUE)
```

```
## Disp = 0 , BCV = 0
```

```
sc2_y  <- estimateGLMTrendedDisp(sc2_y , design)
sc2_y  <- estimateGLMTagwiseDisp(sc2_y , design)
#plotBCV(sc2_y)

sc2_fit <- glmFit(sc2_y, design)

sc2_lrt <- glmLRT(sc2_fit)

#topTags(sc2_lrt)

sc2_FDR <- p.adjust(sc2_lrt$table$PValue, method="BH")
sum(sc2_FDR < 0.05)
```

```
## [1] 1341
```

```
top <- rownames(topTags(sc2_lrt))
#cpm(sc2_y)[top,]
summary(sc2_dt <- decideTestsDGE(sc2_lrt))
```

```
##    [,1]
## -1  446
## 0   418
## 1   895
```

```
sc2_isDE <- as.logical(sc2_dt)
sc2_DEnames <- rownames(sc2_y)[sc2_isDE]
#plotSmear(sc2_lrt, de.tags=DEnames,ylim=c(-2,2))
#abline(h=c(-1,1), col="blue")
```

Scenario 3

For the third scenario, the same approach as above for scenario 2 was applied. However, treatment C was chosen to have an effect that varied across genes. For this treatment, the amount by which the mean expect reads for a gene was modified was randomly selected with replacement from the vector [0,-20,-80,50,200] with probability [0.1,0.8,0.05,0.04,0.01]. These were values were selected at the beginning of the simulations, and remained constant across replicates.

```
set.seed(seed = 2222)

randomDev = sample(x = c(deviations,200),
                   replace = T,
                   prob = c(0.1,0.8,0.05,0.04,0.01),
                   size = nGenes)

meanGeneCounts = rnbinom(n = nGenes, size = size, mu = mean)

meanBiolTreats = matrix(0,nrow = nGenes, ncol = treats*bioReps)

for(i in 1:nGenes){
  meanBiolTreats[i,] = rnorm(n = treats*bioReps,
                             mean = meanGeneCounts[i],
                             sd = std1)
}

readCounts = matrix(0,
                    ncol = treats*bioReps*techReps,
                    nrow = nGenes)

for(g in 1:nGenes){
  means = meanBiolTreats[g,]
  for(t in 1:treats){
    diff = ifelse(t!=4,deviations[t],randomDev[g])
    for(b in 1:bioReps){
      pos = bioReps*(t-1)+b
      nReads = floor(rnorm(n = techReps, mean = means[pos], sd = std2))
      nReads = nReads + diff
      nReads[nReads<0] = 0
```

```
      p = 3*(pos-1) + 1
      readCounts[g,p:(p+techReps-1)] <- nReads
    }
  }
}

keep=rowSums(cpm(readCounts)>2)==96

readCountsKeep = readCounts[keep,]
nrow(readCountsKeep)
```

```
## [1] 1724
```

```
sc3_y = DGEList(counts = readCountsKeep,group = mainFac)

#plotMDS(y,labels = mainFac,col=rep(c("black","navyblue","gold","darkgreen"),each=24))


sc3_logFC <- predFC(sc3_y,design,prior.count=1,dispersion=0.05)

cor(sc3_logFC)
```

```
##               (Intercept) mainFactreatA mainFactreatB mainFactreatC
## (Intercept)     1.0000000     0.8673904     0.9199124     0.6149711
## mainFactreatA   0.8673904     1.0000000     0.9489920     0.6872254
## mainFactreatB   0.9199124     0.9489920     1.0000000     0.6633538
## mainFactreatC   0.6149711     0.6872254     0.6633538     1.0000000
```

```
sc3_y  <- estimateGLMCommonDisp(sc3_y , design, verbose=TRUE)
```

```
## Disp = 0 , BCV = 1e-04
```

```
sc3_y  <- estimateGLMTrendedDisp(sc3_y , design)
sc3_y  <- estimateGLMTagwiseDisp(sc3_y , design)
#plotBCV(sc3_y)

sc3_fit <- glmFit(sc3_y, design)

sc3_lrt <- glmLRT(sc3_fit)

#topTags(sc3_lrt)

sc3_FDR <- p.adjust(sc3_lrt$table$PValue, method="BH")
sum(sc3_FDR < 0.05)
```

```
## [1] 1601
```

```
top <- rownames(topTags(sc3_lrt))
#cpm(sc3_y)[top,]
summary(sc3_dt <- decideTestsDGE(sc3_lrt))
```

```
##    [,1]
## -1 1022
## 0   123
## 1   579
```

```
sc3_isDE <- as.logical(sc3_dt)
sc3_DEnames <- rownames(sc3_y)[sc3_isDE]
#plotSmear(sc3_lrt, de.tags=DEnames,ylim=c(-2,2))
#abline(h=c(-1,1), col="blue")
```

Scenario 4

For the fourth scenario, the same approach as above for scenario 2 was applied. However, treatment C was chosen to have an effect that varied across genes and replicates. For this treatment, the amount by which the mean expect reads for a gene was modified was randomly selected with replacement from the vector [0,-20,-80,50,200] with probability [0.1,0.8,0.05,0.04,0.01]. The value by which the read count at a gene was modified was selected to be replicate specific, and thus changed across replicates.

```
set.seed(seed = 9999)

meanGeneCounts = rnbinom(n = nGenes,
                         size = size,
                         mu = mean)

meanBiolTreats = matrix(0,
                        nrow = nGenes,
                        ncol = treats*bioReps)

for(i in 1:nGenes){
  meanBiolTreats[i,] = rnorm(n = treats*bioReps,
                             mean = meanGeneCounts[i],
                             sd = std1)
}

readCounts = matrix(0,
                    ncol = treats*bioReps*techReps,
                    nrow = nGenes)

for(g in 1:nGenes){
  means = meanBiolTreats[g,]
  for(t in 1:treats){
    diff = deviations[t]
    for(b in 1:bioReps){
      pos = bioReps*(t-1)+b
      nReads = floor(rnorm(n = techReps,
                           mean = means[pos],
                           sd = std2))
      nReads = nReads + ifelse(t!=4,
                               diff,
                               sample(x = c(deviations,200),
                                      replace = T,
                                      prob = c(0.1,0.8,0.05,0.04,0.01),
                                      size = 1))
```

```
      nReads[nReads<0] = 0
      p = 3*(pos-1) + 1
      readCounts[g,p:(p+techReps-1)] <- nReads
    }
  }
}


keep=rowSums(cpm(readCounts)>2)==96

readCountsKeep = readCounts[keep,]
nrow(readCountsKeep)
```

```
## [1] 1728
```

```
sc4_y = DGEList(counts = readCountsKeep,group = mainFac)

#plotMDS(y,labels = mainFac,col=rep(c("black","navyblue","gold","darkgreen"),each=24))


sc4_logFC <- predFC(sc4_y,design,prior.count=1,dispersion=0.05)

cor(sc4_logFC)
```

```
##               (Intercept) mainFactreatA mainFactreatB mainFactreatC
## (Intercept)     1.0000000     0.8714187     0.9237018     0.7575432
## mainFactreatA   0.8714187     1.0000000     0.9447789     0.8368684
## mainFactreatB   0.9237018     0.9447789     1.0000000     0.8027103
## mainFactreatC   0.7575432     0.8368684     0.8027103     1.0000000
```

```
sc4_y  <- estimateGLMCommonDisp(sc4_y , design, verbose=TRUE)
```

```
## Disp = 0.01821 , BCV = 0.1349
```

```
sc4_y  <- estimateGLMTrendedDisp(sc4_y , design)
sc4_y  <- estimateGLMTagwiseDisp(sc4_y , design)
#plotBCV(sc4_y)

sc4_fit <- glmFit(sc4_y, design)

sc4_lrt <- glmLRT(sc4_fit)

#topTags(sc4_lrt)

sc4_FDR <- p.adjust(sc4_lrt$table$PValue, method="BH")
sum(sc4_FDR < 0.05)
```

```
## [1] 1349
```

```
top <- rownames(topTags(sc4_lrt))
#cpm(sc4_y)[top,]
summary(sc4_dt <- decideTestsDGE(sc4_lrt))
```

```
##    [,1]
## -1  855
## 0   379
## 1   494
```

```
sc4_isDE <- as.logical(sc4_dt)
sc4_DEnames <- rownames(sc4_y)[sc4_isDE]
#plotSmear(sc4_lrt, de.tags=DEnames,ylim=c(-2,2))
#abline(h=c(-1,1), col="blue")
```

Results

As we can see in figures 1 and 2, these four scenarios produce very different patterns that are easily recognisable. In scenario 1 (Figure 1 top row), the MDS plot shows no structure in the read count data across the different replicates and treatments, and we do not see any significant signatures of 'differential gene expression' in the scatter plot. In scenario 2 (Figure 1 bottom row), the first component of MDS plot (x-axis) shows a clear separation between the treatment A (the worser performer) and the two other treatments that preserved RNA similarly to the control. In the scatter plot, we can see a number of genes showing signaturs of differential expression (red dots), but log2 fold change is constrained to values between -1 and 1, showing no drastic changes after accounting for differences in the total number of reads obtained for each treatment.

Scenarios 3 and 4 (Figure 2), show a different picture. In scenario 3 (Figure 2 top row), the first component of the MDS (x-axis) shows a clear separation between Treatment B (performing similarly to the control), and Treatments A and C. While the second component (y-axis), shows a clear separation between Treatments A (uniform loss of reads across genes) and C (non-uniform loss of reads across genes). The scatter plot suggests a trend for *down-regulation* of genes (loss of reads), but with a few genes appearing to be *up-regulated* (gain of reads). This fits well with our simulation approach, in which the effect of the preservation method is gene-specific. In scenario 4 (Figure 2 bottom row), we can see clear differences in the expected patterns relative to scenario 3, if the effects of the preservation method are random across genes/replicates. The MDS plot shows that samples from Treatment C do not cluster together, and the scatter plot shows a trend for *down-regulation* towards genes with lower counts (left portion of the figure), but no signatures of *up-regulation*.

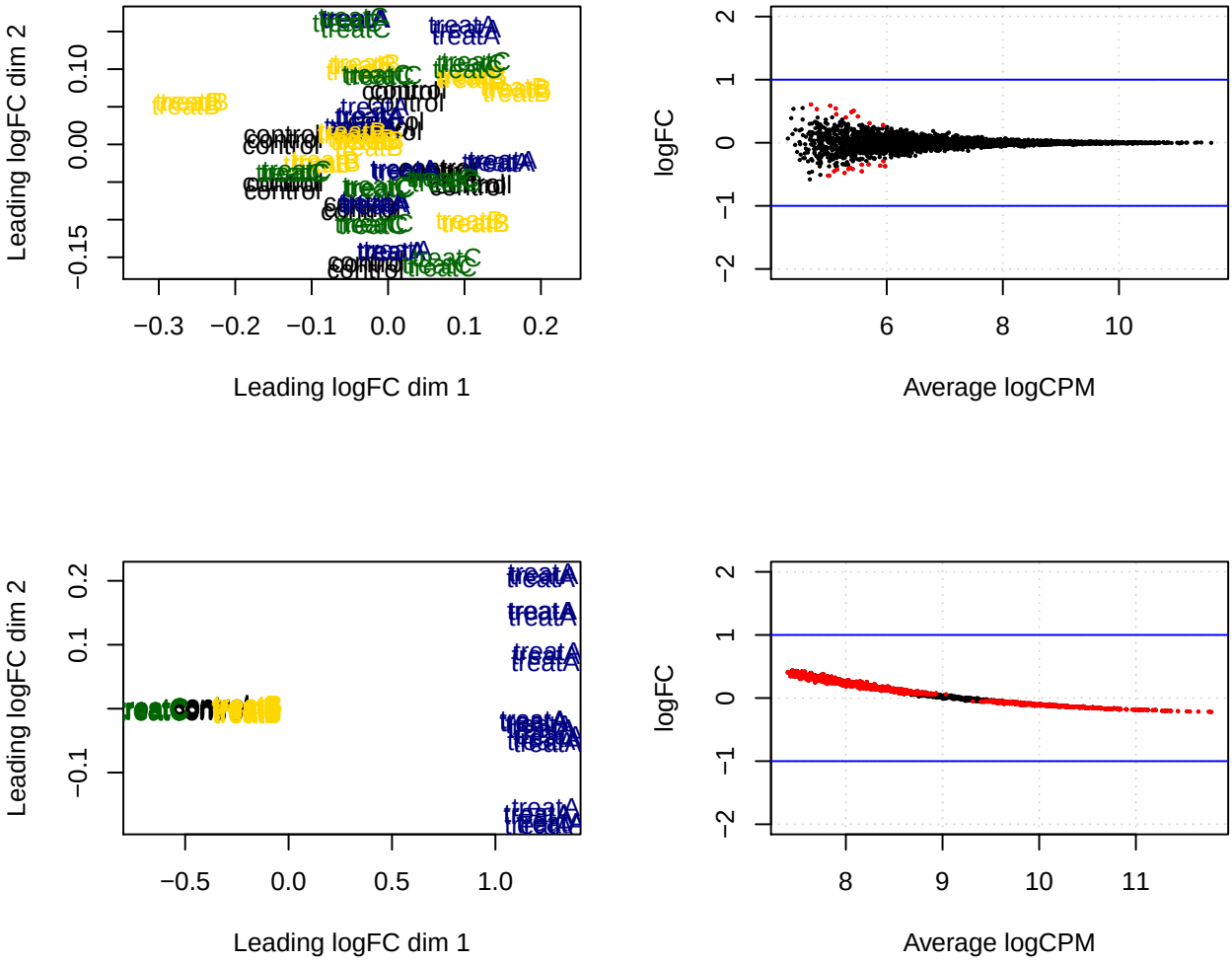

Figure 1: Simulation results for scenarios 1 and 2. Left: MDS plots; Right: Scatter plot of log Counts per Million reads vs log2 Fold-Change in read count; blue horizontal lines represent log2 fold change.

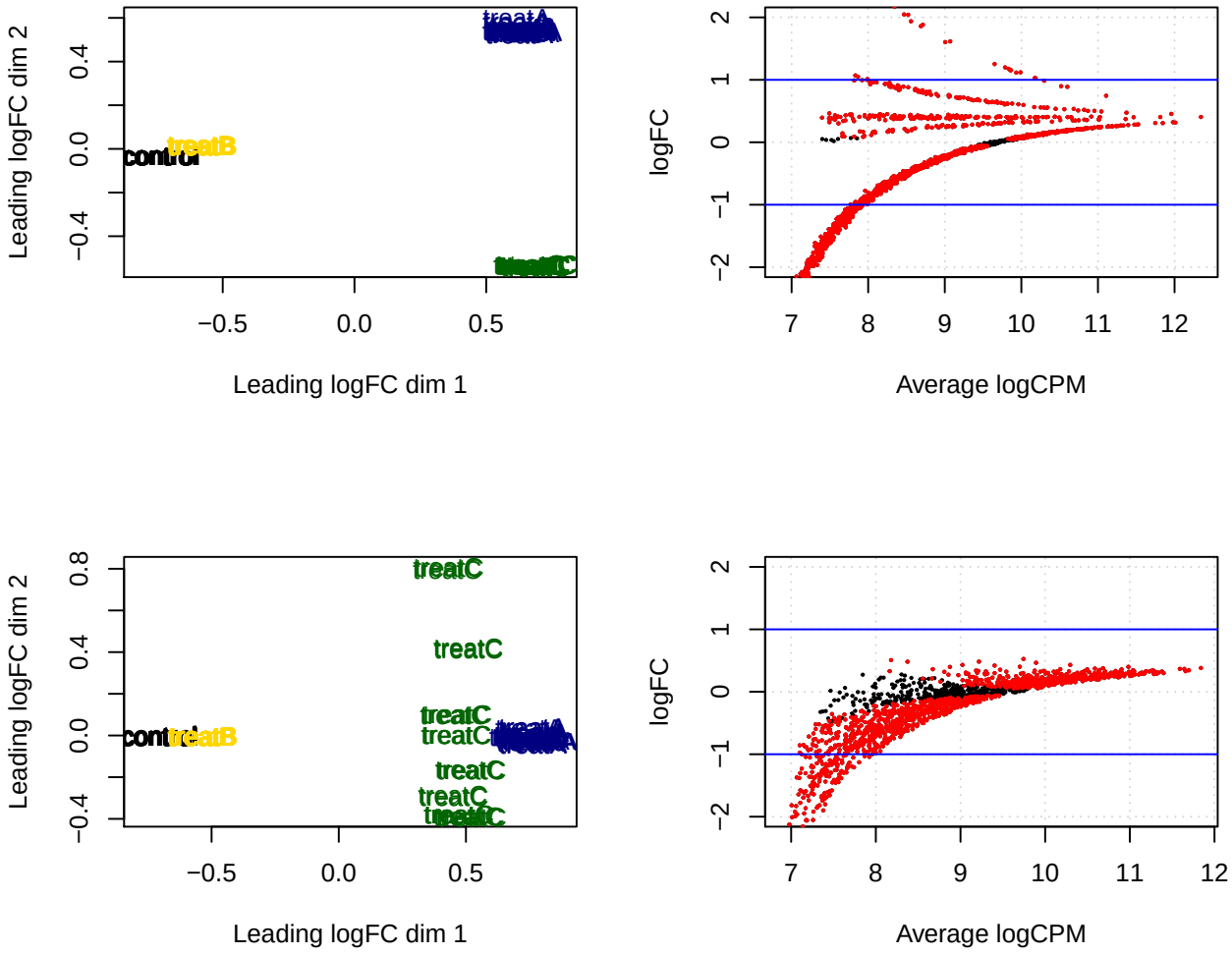

Figure 2: Simulation results for scenarios 3 and 4. Left: MDS plots; Right: Scatter plot of log Counts per Million reads vs log2 Fold-Change in read count; blue horizontal lines represent log2 fold change.