Further Analysis of RNA-seq data

Mark Dunning

Last modified: March 11, 2016

Contents

1 Clustering

1.1 Example from the DESeq2 Vignette

The general application of clustering to RNA-seq data is outlined by the DESeq2 vignette

```
library("pasilla")
library("Biobase")
data("pasillaGenes")
countData <- counts(pasillaGenes)
colData <- pData(pasillaGenes)[,c("condition","type")]
```

```
library(DESeq2)
dds <- DESeqDataSetFromMatrix(countData = countData,
colData = colData,
design = ~ condition)
dds

## class: DESeqDataSet
## dim: 14470 7
## metadata(0):
## assays(1): counts
## rownames(14470): FBgn0000003 FBgn0000008 ... FBgn0261574 FBgn0261575
## rowRanges metadata column names(0):
## colnames(7): treated1fb treated2fb ... untreated3fb untreated4fb
## colData names(2): condition type
```

```r
featureData <- data.frame(gene=rownames(pasillaGenes))
(mcols(dds) <- DataFrame(mcols(dds), featureData))
```

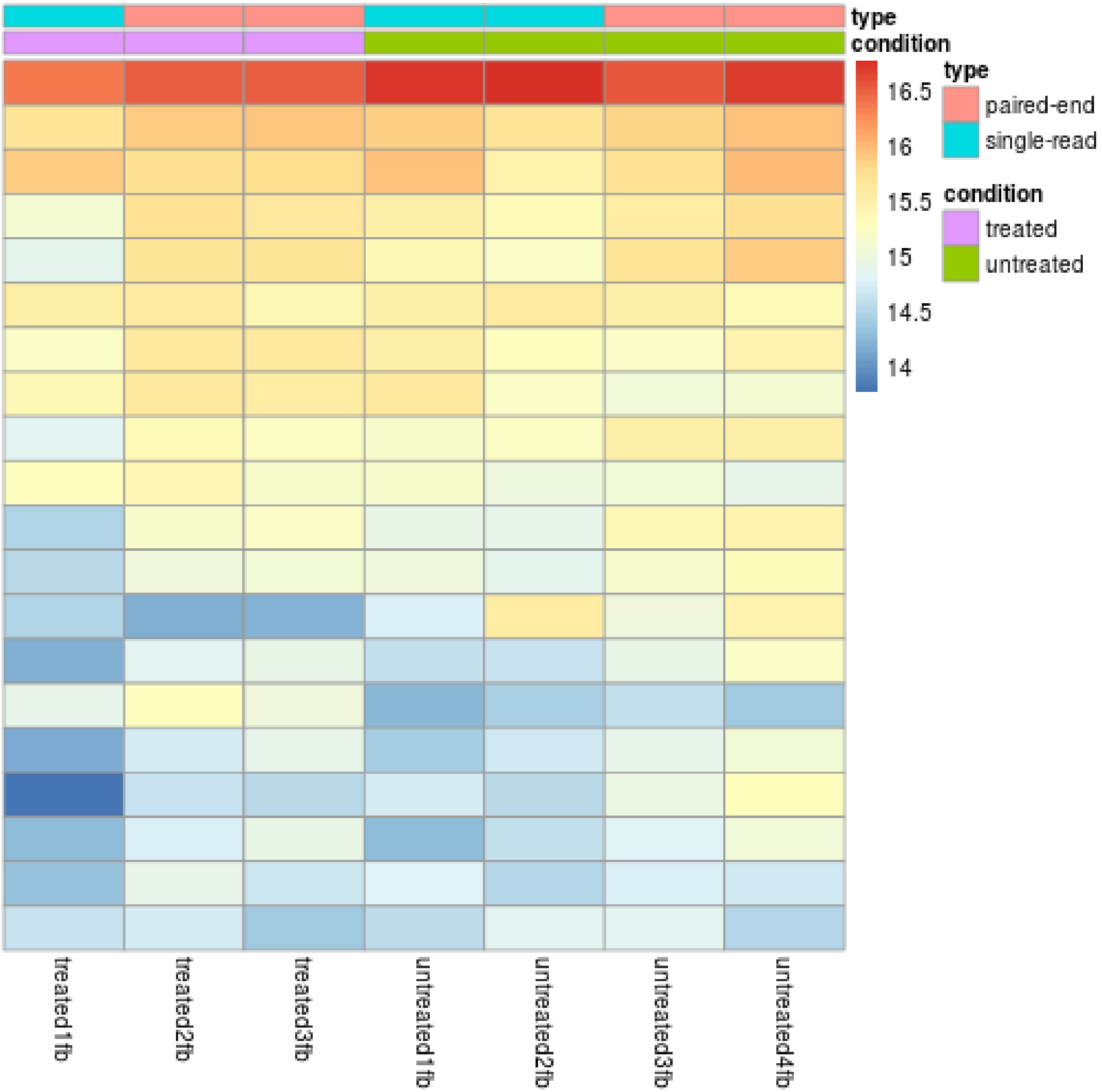
```
## DataFrame with 14470 rows and 1 column
##              gene
##          <factor>
## 1     FBgn0000003
## 2     FBgn0000008
## 3     FBgn0000014
## 4     FBgn0000015
## 5     FBgn0000017
## ...           ...
## 14466 FBgn0261571
## 14467 FBgn0261572
## 14468 FBgn0261573
## 14469 FBgn0261574
## 14470 FBgn0261575
```

```r
dds <- estimateSizeFactors(dds)
```

```r
library("pheatmap")
select <- order(rowMeans(counts(dds,normalized=TRUE)),decreasing=TRUE)[1:20]
```

```r
nt <- normTransform(dds) # defaults to log2(x+1)
log2.norm.counts <- assay(nt)[select,]
df <- as.data.frame(colData(dds)[,c("condition","type")])
```

```r
pheatmap(log2.norm.counts, cluster_rows=FALSE, show_rownames=FALSE,
cluster_cols=FALSE, annotation_col=df)
```



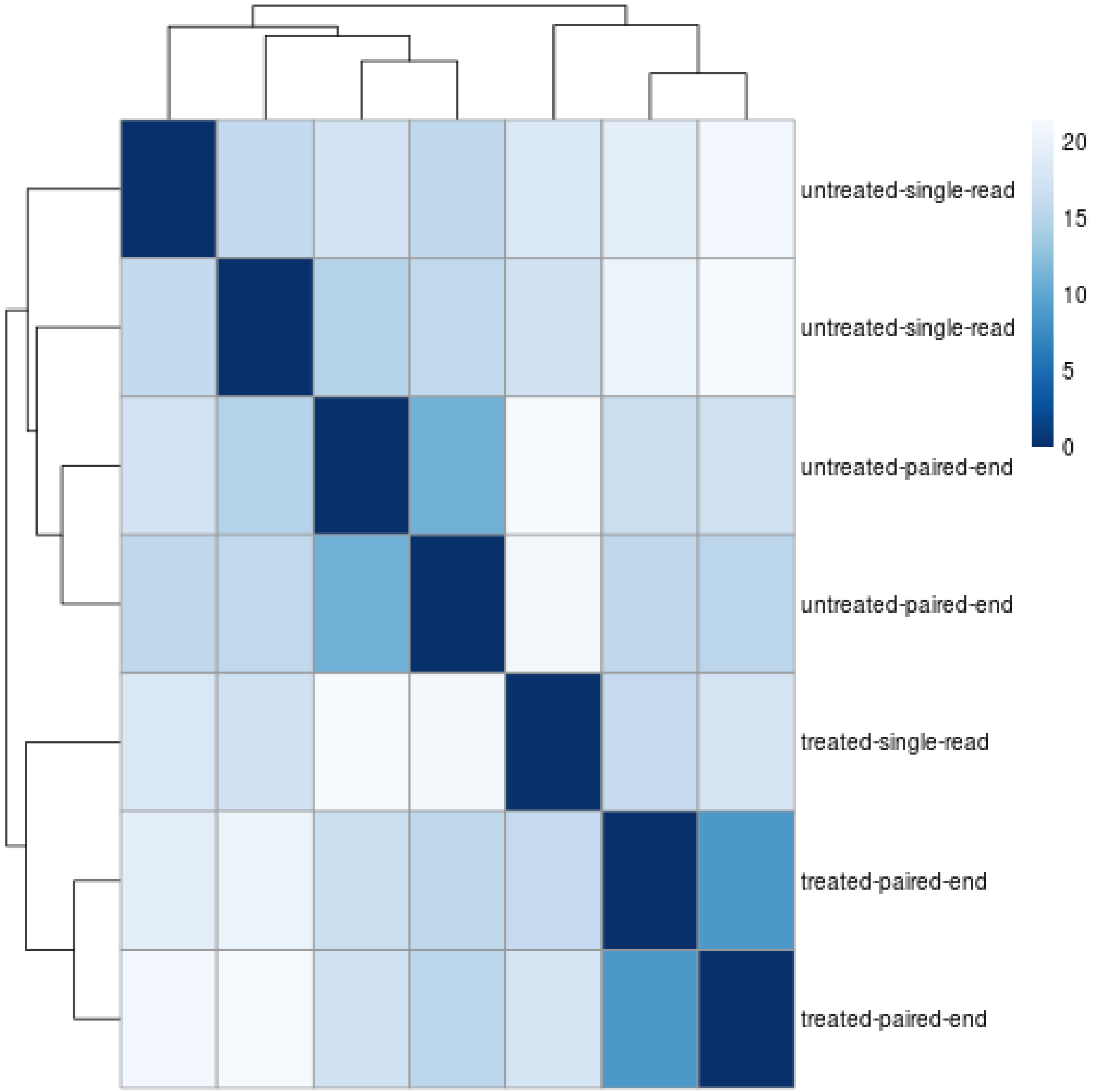
```
rld <- rlog(dds,blind = TRUE)
vsd <- varianceStabilizingTransformation(dds)
head(assay(rld), 3)
```

```
##             treated1fb treated2fb treated3fb untreated1fb untreated2fb untreated3fb
## FBgn0000003  -2.706406  -2.705902  -2.688123    -2.706143    -2.706466    -2.705817
## FBgn0000008   5.690343   5.746280   5.659962     5.630195     5.708844     5.859262
## FBgn0000014  -1.348685  -1.371296  -1.371567    -1.371876    -1.372650    -1.350820
##             untreated4fb
## FBgn0000003    -2.705902
## FBgn0000008     5.541170
## FBgn0000014    -1.371294
```

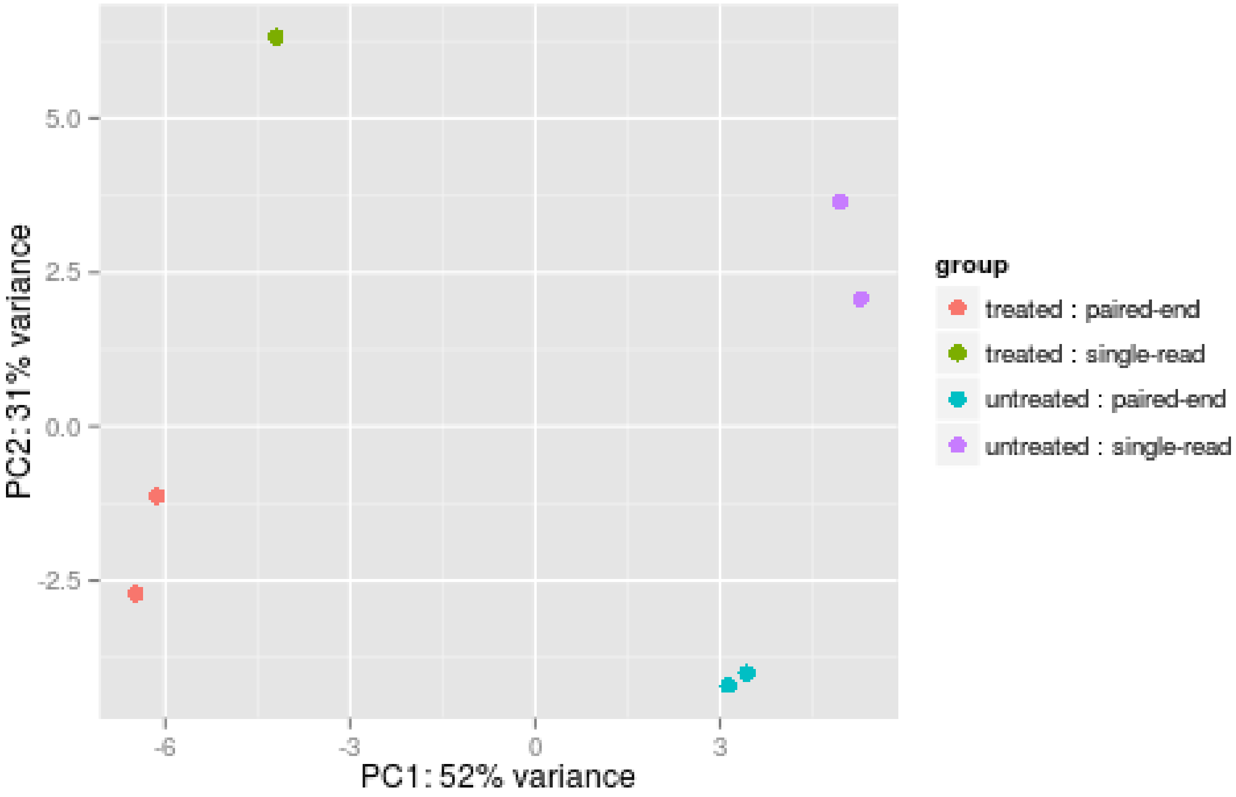
```
sampleDists <- dist(t(assay(rld)))
sampleDists
```

```
##              treated1fb treated2fb treated3fb untreated1fb untreated2fb untreated3fb
## treated2fb    16.065502
## treated3fb    17.783173   8.735605
## untreated1fb  18.243759  19.328951  20.816289
## untreated2fb  17.304728  20.186898  21.330465    15.886346
## untreated3fb  21.432540  16.775214  17.161326    17.502227    15.040183
## untreated4fb  20.947289  15.603203  15.407050    15.585671    15.793143    11.028331
```

```
library("RColorBrewer")
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(rld$condition, rld$type, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists,
col=colors)
```



```
plotPCA(rld, intgroup=c("condition", "type"))
```



1.2 Re-visiting our ESCC dataset

```
library(DESeq2)
load("Day2/Counts.RData")
#Load data
Counts <- tmp$counts
colnames(Counts) <- c("16N", "16T", "18N", "18T", "19N", "19T") #Rename the columns
Coldata <- data.frame(sampleReplicate=c("16", "16", "18", "18", "19", "19"),
sampleType=c("N", "T", "N", "T", "N", "T"))
rownames(Coldata) <- c("16N", "16T", "18N", "18T", "19N", "19T")

deSeqData <- DESeqDataSetFromMatrix(countData=Counts, colData=Coldata,
               design= ~sampleReplicate + sampleType)
deSeqData
```

```
## class: DESeqDataSet
## dim: 25702 6
## metadata(0):
## assays(1): counts
## rownames(25702): 653635 100422834 ... 114760 100506511
## rowRanges metadata column names(0):
## colnames(6): 16N 16T ... 19N 19T
## colData names(2): sampleReplicate sampleType
```

```
nrow(counts(deSeqData))
```

```
## [1] 25702
```

```
summary(rowSums(counts(deSeqData)))
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##       0       2      88    1641     769 2523000
```

```
deSeqData <- deSeqData[rowSums(counts(deSeqData))>1,]
```

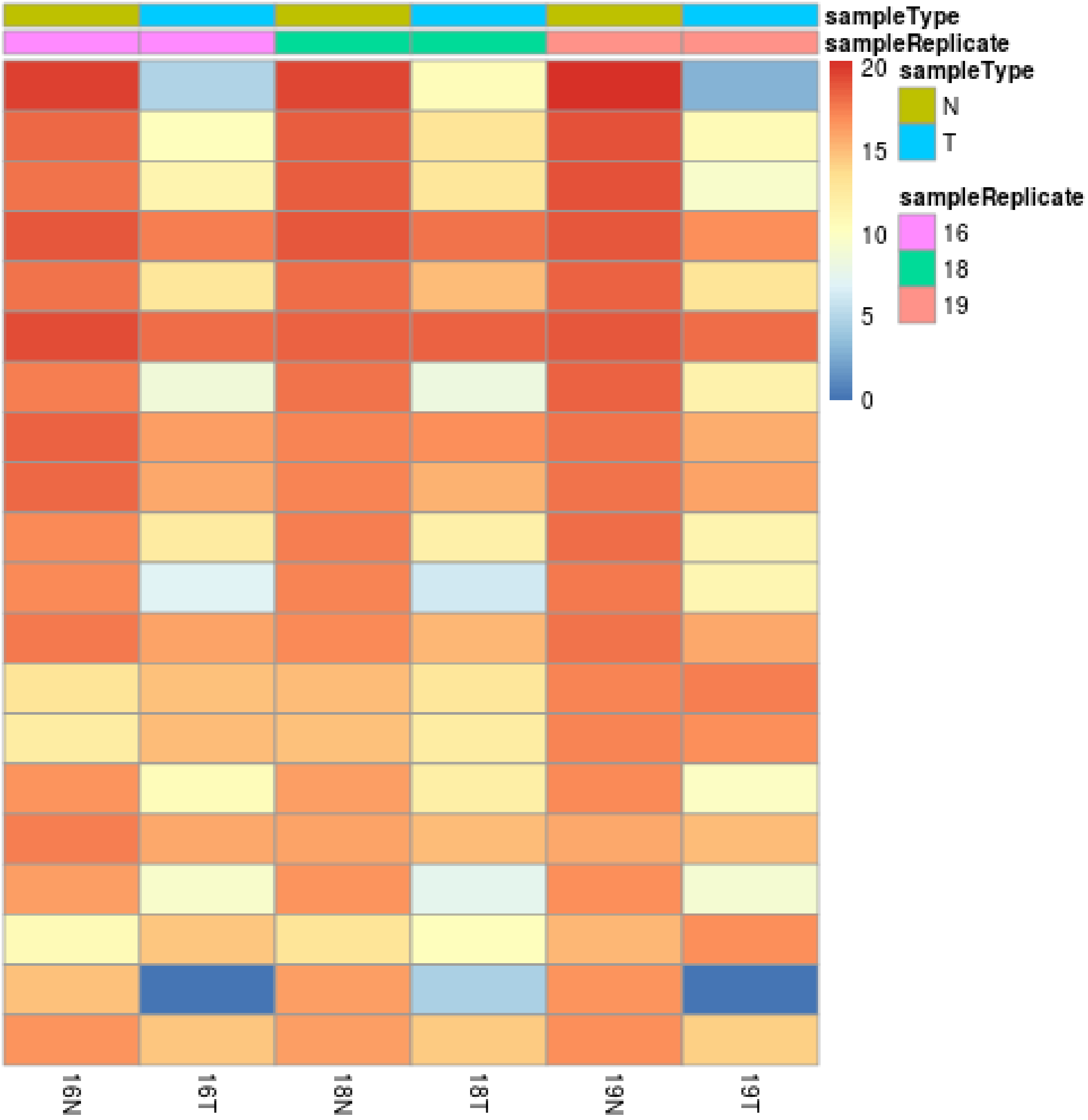
```
deSeqData <- estimateSizeFactors(deSeqData)
colData(deSeqData)
```

```
## DataFrame with 6 rows and 3 columns
##     sampleReplicate sampleType sizeFactor
##            <factor>   <factor>  <numeric>
## 16N              16          N  0.7951120
## 16T              16          T  1.5016595
## 18N              18          N  0.6487028
## 18T              18          T  2.0675928
## 19N              19          N  0.4225496
## 19T              19          T  1.5782240
```

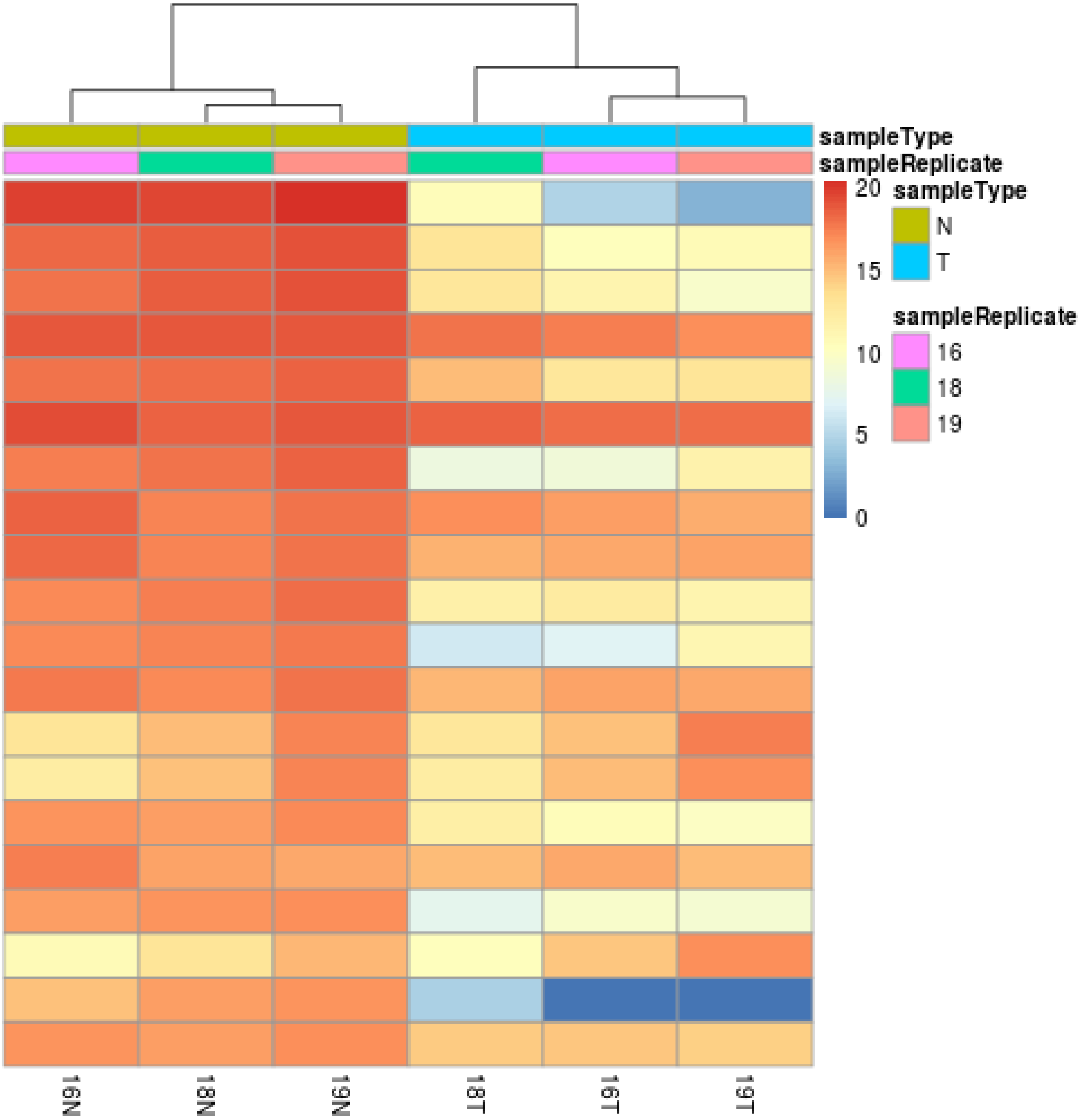
```
head(counts(deSeqData))
```

```
##           16N 16T 18N 18T 19N 19T
## 653635      0   1   0   1   0   0
## 729737      1   0   2   2   2   1
## 100131754   1   6   3   4   2   6
## 100133331   1   0   1   1   1   1
## 100288069   2   4   0   2   0   2
## 400728      0   1   2   2   0   1
```

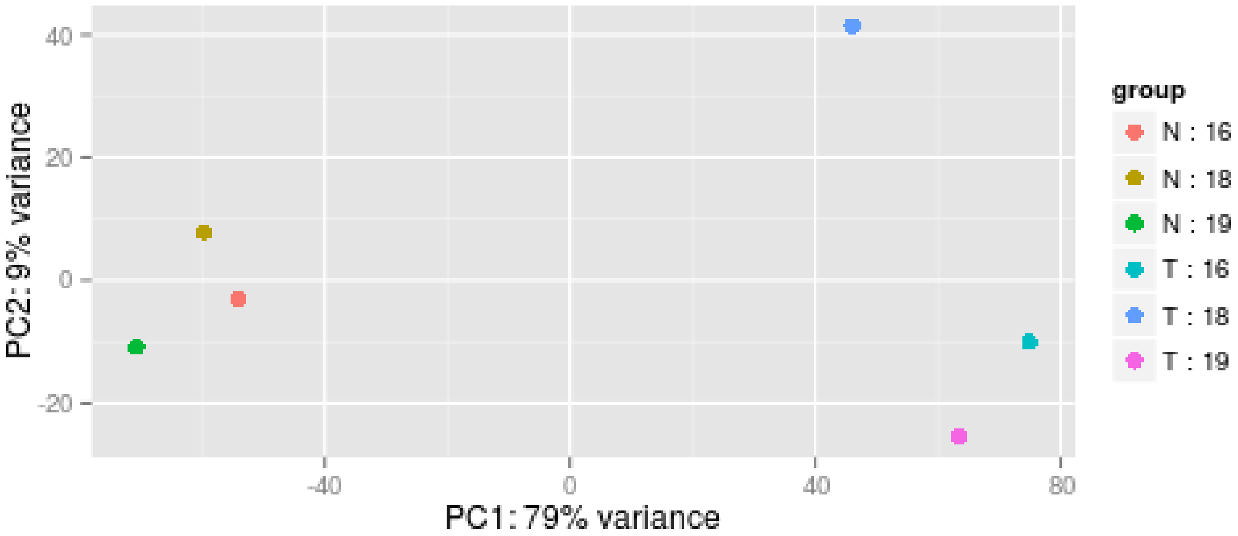
```
library(genefilter)
nt <- normTransform(deSeqData) # defaults to log2(x+1)
select <- order(rowVars(counts(deSeqData,normalized=TRUE)),decreasing=TRUE)[1:20]

log2.norm.counts <- assay(nt)[select,]
df <- as.data.frame(colData(deSeqData)[,c("sampleReplicate","sampleType")])
pheatmap(log2.norm.counts, cluster_rows=FALSE, show_rownames=FALSE,
cluster_cols=FALSE, annotation_col=df)
```



```
pheatmap(log2.norm.counts, cluster_rows=FALSE, show_rownames=FALSE,
cluster_cols=TRUE, annotation_col=df)
```

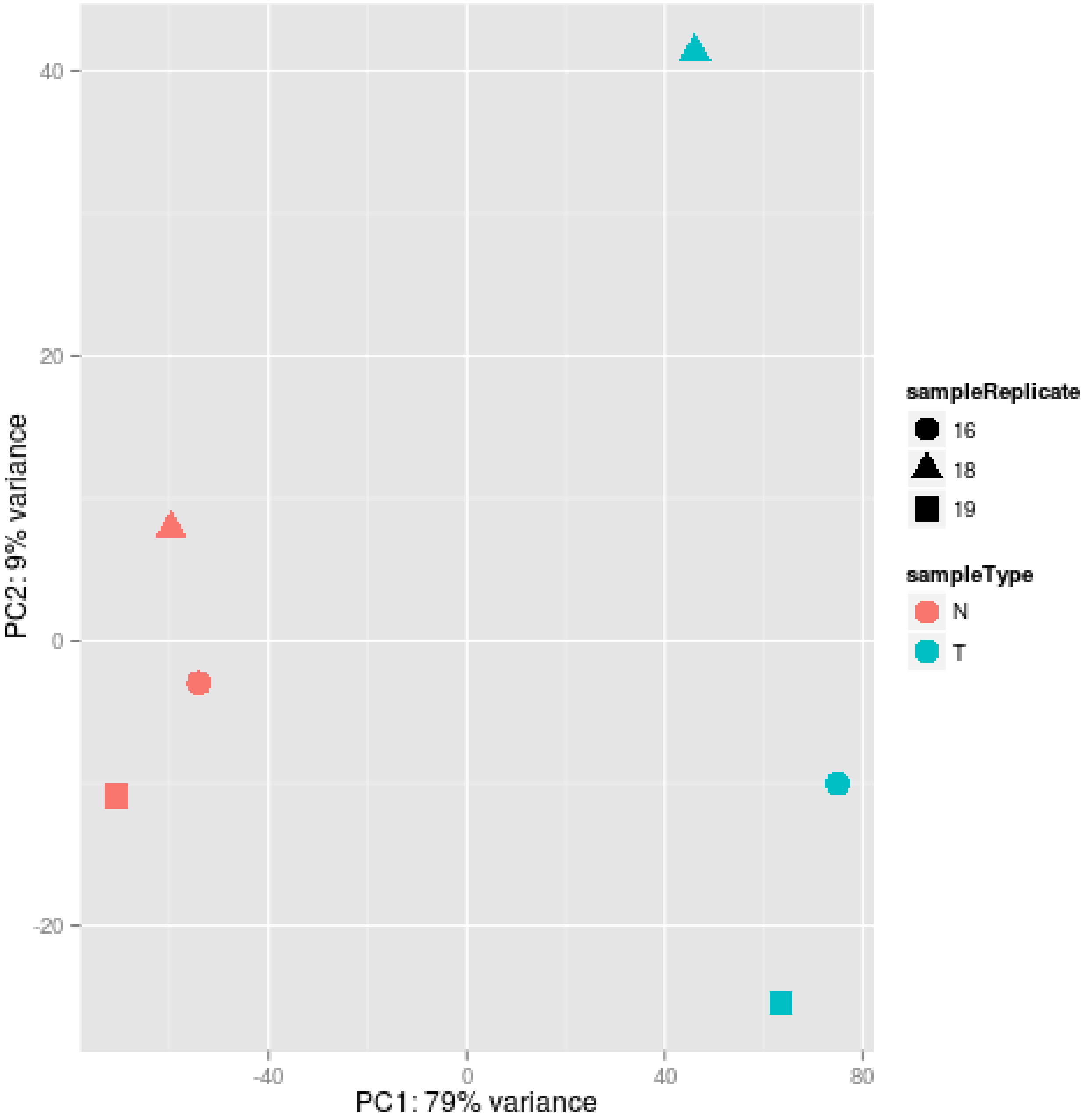


```
plotPCA(nt, intgroup=c("sampleType","sampleReplicate"))
```



```
pcData <- plotPCA(nt, intgroup=c("sampleType","sampleReplicate"),returnData=TRUE)
pcData

##            PC1        PC2  group sampleType sampleReplicate name
## 16N -54.03182  -2.966875 N : 16          N              16  16N
## 16T  74.89342 -10.012828 T : 16          T              16  16T
## 18N -59.67904   7.900549 N : 18          N              18  18N
## 18T  46.04823  41.388418 T : 18          T              18  18T
## 19N -70.69108 -10.878899 N : 19          N              19  19N
## 19T  63.46029 -25.430365 T : 19          T              19  19T
```

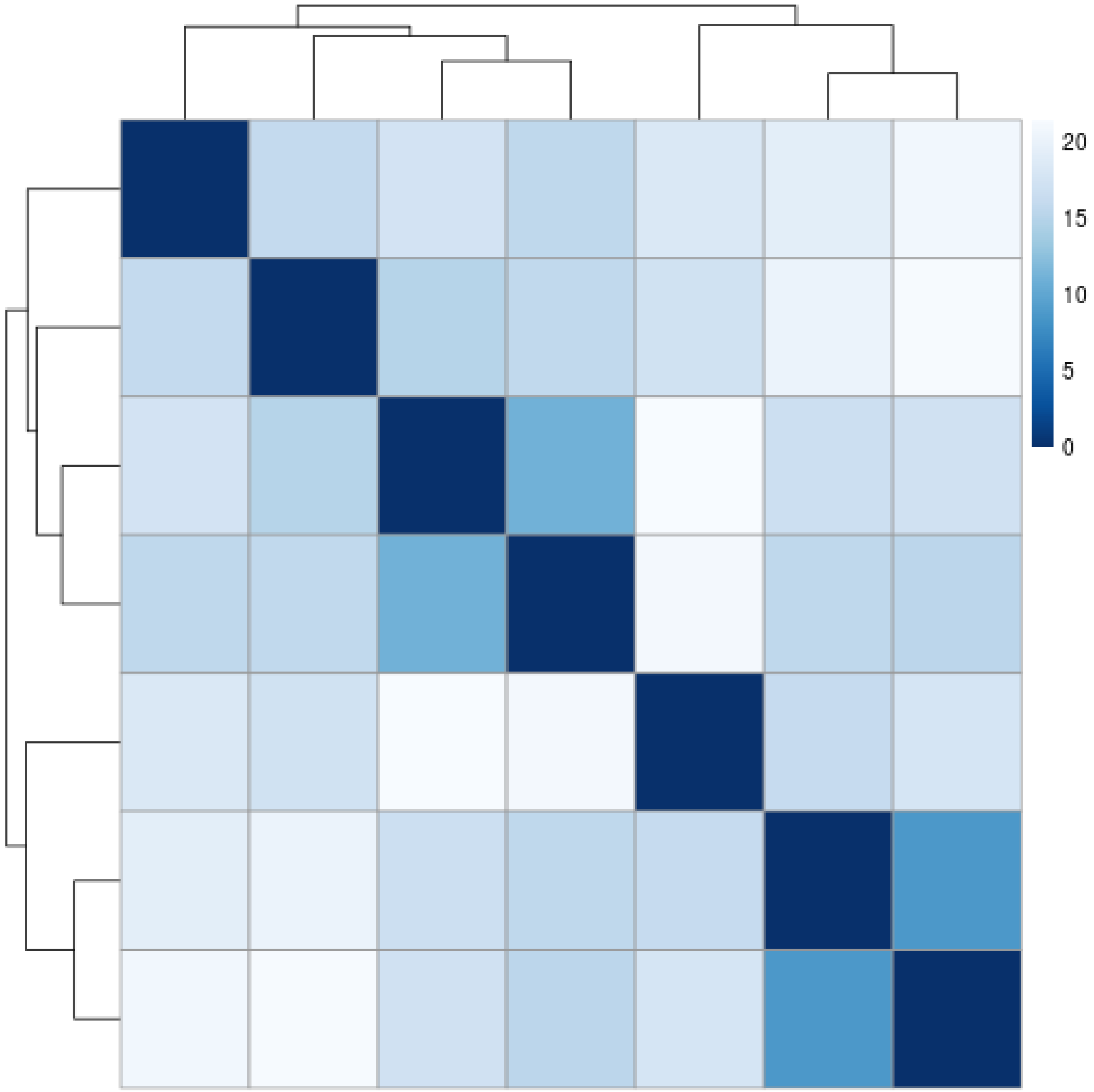
```
library(ggplot2)
percentVar <- round(100*attr(pcData, "percentVar"))
ggplot(pcData, aes(x=PC1,y=PC2,color=sampleType,shape=sampleReplicate))+
  geom_point(size=5)+
  xlab(paste0("PC1: ", percentVar[1], "% variance")) +
  ylab(paste0("PC2: ", percentVar[2], "% variance"))
```



```
sampleDists <- dist(t(assay(rld)))
sampleDists
```

```
##              treated1fb treated2fb treated3fb untreated1fb untreated2fb untreated3fb
## treated2fb    16.065502
## treated3fb    17.783173   8.735605
## untreated1fb  18.243759  19.328951  20.816289
## untreated2fb  17.304728  20.186898  21.330465    15.886346
## untreated3fb  21.432540  16.775214  17.161326    17.502227    15.040183
## untreated4fb  20.947289  15.603203  15.407050    15.585671    15.793143    11.028331
```

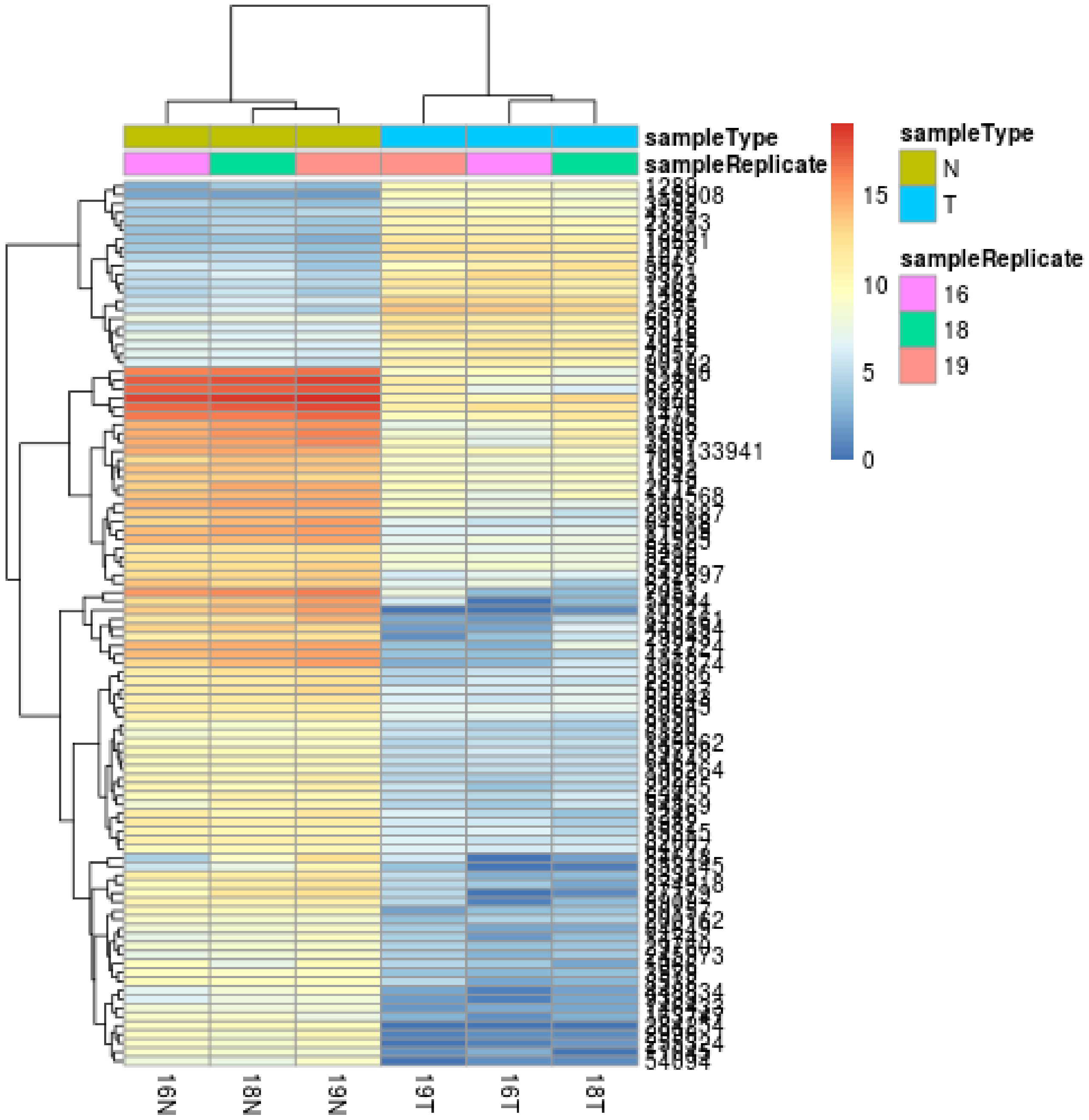
```
sampleDistMatrix <- as.matrix(sampleDists)
rownames(sampleDistMatrix) <- paste(rld$sampleReplicate, rld$sampleType, sep="-")
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
clustering_distance_rows=sampleDists,
clustering_distance_cols=sampleDists,
col=colors)
```



```
deSeqData <- estimateDispersions(deSeqData)
mcols(deSeqData)

## DataFrame with 19759 rows and 9 columns
##          baseMean      baseVar   allZero dispGeneEst   dispFit dispersion  dispIter
##         <numeric>    <numeric> <logical>   <numeric> <numeric>  <numeric> <integer>
## 1       0.1915974   0.09142536     FALSE       1e-08 30.177678  10.000000         2
## 2       1.7791441   3.16701736     FALSE       1e-08  3.371946   2.048356        10
## 3       3.3912347   2.10590810     FALSE       1e-08  1.834066   1.164001        10
## 4       1.0471810   0.72294203     FALSE       1e-08  5.633245   3.119035         9
## 5      1.2356074   1.36074619     FALSE       1e-08  4.795058   3.223883         8
## ...          ...          ...       ...         ...       ...        ...       ...
## 19755  9.2406499   65.3232419     FALSE  0.20676722 0.7596962  0.6331789         8
## 19756 23.2888549  348.4889762     FALSE  0.00000001 0.3839695  0.2984309         7
## 19757  0.8152529    0.9601181     FALSE  0.00000001 7.1969006  5.5870641         8
## 19758 35.1201859 1632.9900258     FALSE  0.00000001 0.3007107  0.2377299         7
## 19759  0.8152529    0.9601181     FALSE  0.00000001 7.1969006  5.5870641         8
##       dispOutlier   dispMAP
##         <logical> <numeric>
## 1           FALSE 10.000000
## 2           FALSE  2.048356
## 3           FALSE  1.164001
## 4           FALSE  3.119035
## 5           FALSE  3.223883
## ...           ...       ...
## 19755       FALSE 0.6331789
## 19756       FALSE 0.2984309
## 19757       FALSE 5.5870641
## 19758       FALSE 0.2377299
## 19759       FALSE 5.5870641

deSeqData <- nbinomWaldTest(deSeqData)
res <- results(deSeqData)
```

```
res.sig <- res[which(res$padj < 0.05),]
N <- 100
res.sig.ord <- res.sig[order(res.sig$padj,decreasing = FALSE),]
topNGenes <- rownames(res.sig.ord)[1:N]
```

```
pheatmap(assay(nt)[match(topNGenes, rownames(assay(nt))),],annotation_col=df)
```


sampleType
sampleReplicate
15
10
5
0
sampleType
N
T
sampleReplicate
16
18
19
54094
16N
18N
19N
19T
16T
18T