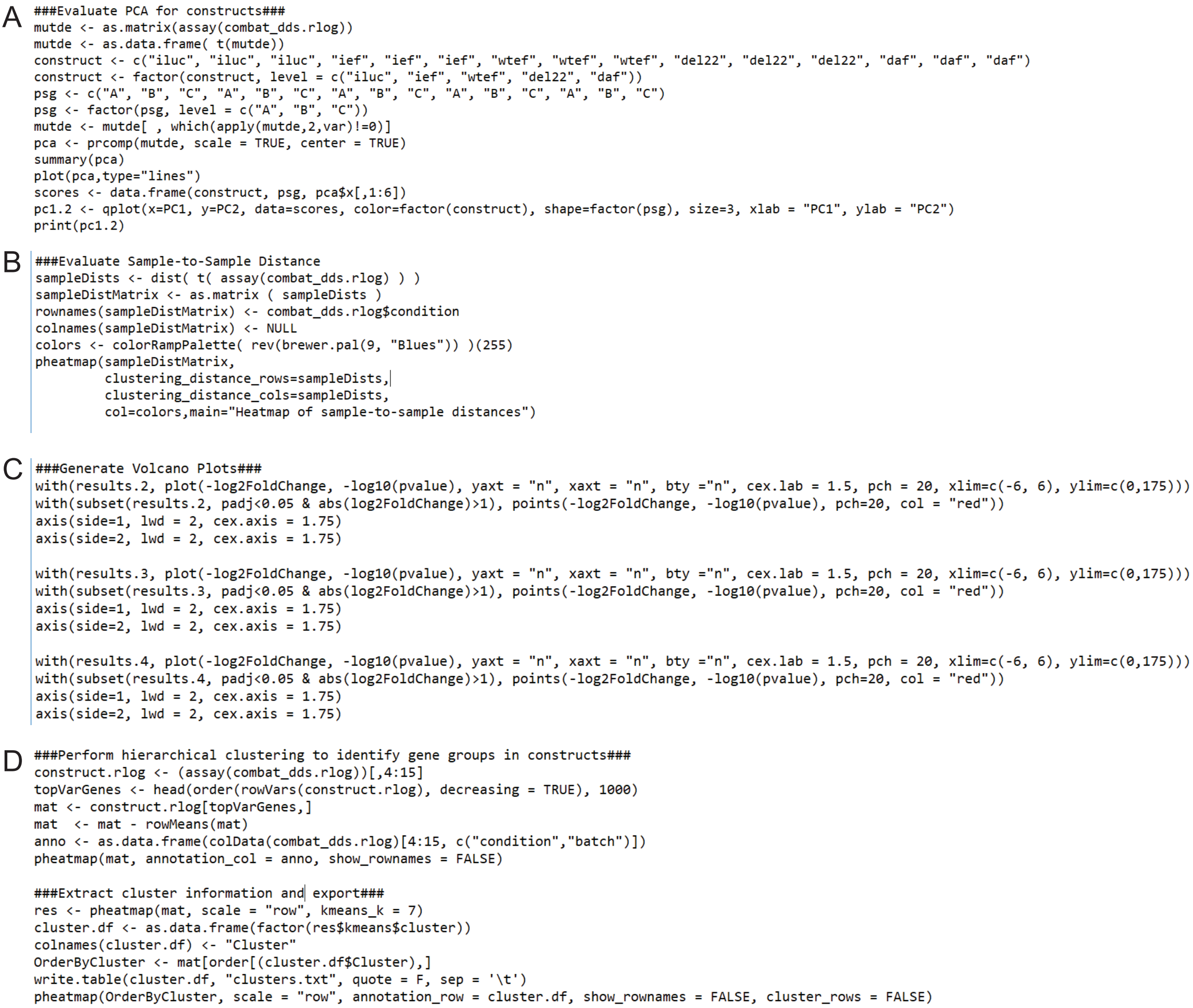

```
###Evaluate PCA for constructs###
mutde <- as.matrix(assay(combat_dds.rlog))
mutde <- as.data.frame( t(mutde))
construct <- c("iluc", "iluc", "iluc", "ief", "ief", "ief", "wtef", "wtef", "wtef", "del22", "del22", "del22", "daf", "daf", "daf")
construct <- factor(construct, level = c("iluc", "ief", "wtef", "del22", "daf"))
psg <- c("A", "B", "C", "A", "B", "C", "A", "B", "C", "A", "B", "C", "A", "B", "C")
psg <- factor(psg, level = c("A", "B", "C"))
mutde <- mutde[ , which(apply(mutde,2,var)!=0)]
pca <- prcomp(mutde, scale = TRUE, center = TRUE)
summary(pca)
plot(pca,type="lines")
scores <- data.frame(construct, psg, pca$x[,1:6])
pc1.2 <- qplot(x=PC1, y=PC2, data=scores, color=factor(construct), shape=factor(psg), size=3, xlab = "PC1", ylab = "PC2")
print(pc1.2)
```

```
###Evaluate Sample-to-Sample Distance
sampleDists <- dist( t( assay(combat_dds.rlog) ) )
sampleDistMatrix <- as.matrix ( sampleDists )
rownames(sampleDistMatrix) <- combat_dds.rlog$condition
colnames(sampleDistMatrix) <- NULL
colors <- colorRampPalette( rev(brewer.pal(9, "Blues")) )(255)
pheatmap(sampleDistMatrix,
         clustering_distance_rows=sampleDists,
         clustering_distance_cols=sampleDists,
         col=colors,main="Heatmap of sample-to-sample distances")
```

```
###Generate Volcano Plots###
with(results.2, plot(-log2FoldChange, -log10(pvalue), yaxt = "n", xaxt = "n", bty ="n", cex.lab = 1.5, pch = 20, xlim=c(-6, 6), ylim=c(0,175)))
with(subset(results.2, padj<0.05 & abs(log2FoldChange)>1), points(-log2FoldChange, -log10(pvalue), pch=20, col = "red"))
axis(side=1, lwd = 2, cex.axis = 1.75)
axis(side=2, lwd = 2, cex.axis = 1.75)

with(results.3, plot(-log2FoldChange, -log10(pvalue), yaxt = "n", xaxt = "n", bty ="n", cex.lab = 1.5, pch = 20, xlim=c(-6, 6), ylim=c(0,175)))
with(subset(results.3, padj<0.05 & abs(log2FoldChange)>1), points(-log2FoldChange, -log10(pvalue), pch=20, col = "red"))
axis(side=1, lwd = 2, cex.axis = 1.75)
axis(side=2, lwd = 2, cex.axis = 1.75)

with(results.4, plot(-log2FoldChange, -log10(pvalue), yaxt = "n", xaxt = "n", bty ="n", cex.lab = 1.5, pch = 20, xlim=c(-6, 6), ylim=c(0,175)))
with(subset(results.4, padj<0.05 & abs(log2FoldChange)>1), points(-log2FoldChange, -log10(pvalue), pch=20, col = "red"))
axis(side=1, lwd = 2, cex.axis = 1.75)
axis(side=2, lwd = 2, cex.axis = 1.75)
```

```
###Perform hierarchical clustering to identify gene groups in constructs###
construct.rlog <- (assay(combat_dds.rlog))[,4:15]
topVarGenes <- head(order(rowVars(construct.rlog), decreasing = TRUE), 1000)
mat <- construct.rlog[topVarGenes,]
mat  <- mat - rowMeans(mat)
anno <- as.data.frame(colData(combat_dds.rlog)[4:15, c("condition","batch")])
pheatmap(mat, annotation_col = anno, show_rownames = FALSE)

###Extract cluster information and export###
res <- pheatmap(mat, scale = "row", kmeans_k = 7)
cluster.df <- as.data.frame(factor(res$kmeans$cluster))
colnames(cluster.df) <- "Cluster"
OrderByCluster <- mat[order[(cluster.df$Cluster),]
write.table(cluster.df, "clusters.txt", quote = F, sep = '\t')
pheatmap(OrderByCluster, scale = "row", annotation_row = cluster.df, show_rownames = FALSE, cluster_rows = FALSE)
```