Exploratory analysis of microbiome dataset

gg

2016-05-04

To run this file: Rscript -e "rmarkdown::render('tiger_ladybug.Rmd')"

Introduction

There is a frequent assertion that data generated by high throughput sequencing instruments are counts. On the surface, this makes sense because we map reads to intervals and initially observe the number of counts per interval. However, immediately problems arise. One common issue is that the results are strongly influenced by the total read count per sample. Samples with similar read counts a

```
Tiger <- round(runif(100, 1800,2200))
Ladybug <- round(runif(100, 8000,12000))
Alien <- round(runif(100, 450,550))

d <- data.frame(cbind(Tiger, Ladybug, Alien))

d.rare <- codaSeq.rarefy(d, n=1000, samples.by.row=FALSE)

if(plot==TRUE) pdf("tiger_count.pdf", height=4, width=14)
par(mfrow=c(1,4),mar=c(5,5,4,1))
plot(d$Tiger, d$Ladybug, main=round(cor(d$Tiger, d$Ladybug), 2), cex.lab=1.8,
    cex.main=2, pch=19,
    col=rgb(1,0,0,0.5), xlab="Tiger", ylab="Ladybug")
plot(d$Tiger, d$Alien, main=round(cor(d$Tiger, d$Alien), 2), cex.lab=1.8,
    cex.main=2, pch=19,
    col=rgb(1,0,0,0.5), xlab="Tiger", ylab="Alien")
plot(d$Ladybug, d$Alien, main=round(cor(d$Ladybug, d$Alien), 2), cex.lab=1.8,
    cex.main=2, pch=19, col=rgb(1,0,0,0.5), xlab="Ladybug", ylab="Alien")
scatterplot3d(d, type="h", lty.hplot=3, angle=40, pch=19, cex.lab=1.5)
```

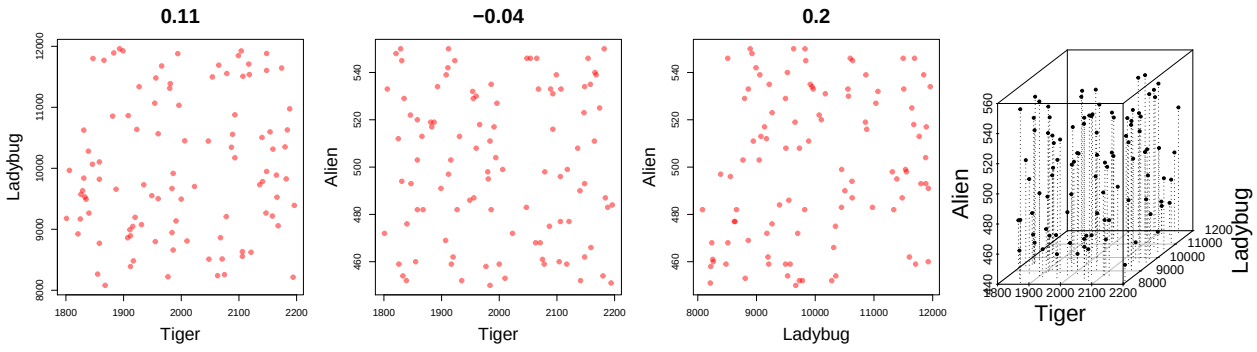

Figure 1: Scatter plots and volume plot of 100 randomly generated tiger, ladybug and space alien counts. It is obvious that there is no correlation between the values, as expected for randomly generated data.

```
if(plot==TRUE) dev.off()
```

s

```
if(plot==TRUE) pdf("tiger_prop.pdf", height=7, width=9)
par(mfrow=c(2,3),mar=c(5,5,4,1))
plot(d.rare$Tiger, d.rare$Ladybug, main=round(cor(d.rare$Tiger, d.rare$Ladybug), 2),
    cex.main=2, pch=19, col=rgb(1,0,0,0.5), cex.lab=1.8, xlab="Tiger", ylab="Ladybug")
plot(d.rare$Tiger, d.rare$Alien, main=round(cor(d.rare$Tiger, d.rare$Alien), 2),
    cex.main=2, pch=19, col=rgb(1,0,0,0.5), cex.lab=1.8, xlab="Tiger", ylab="Alien")
plot(d.rare$Ladybug, d.rare$Alien, main=round(cor(d.rare$Ladybug, d.rare$Alien), 2),
    cex.main=2, pch=19, col=rgb(1,0,0,0.5), cex.lab=1.8, xlab="Ladybug", ylab="Alien")
scatterplot3d(d.rare, type="h", lty.hplot=3, cex.lab=1.5, angle=40, pch=19)
plot(acomp(d.rare), scale=T, center=T, pch=19, cex=1, axes=FALSE)
if(plot==TRUE) dev.off()
```

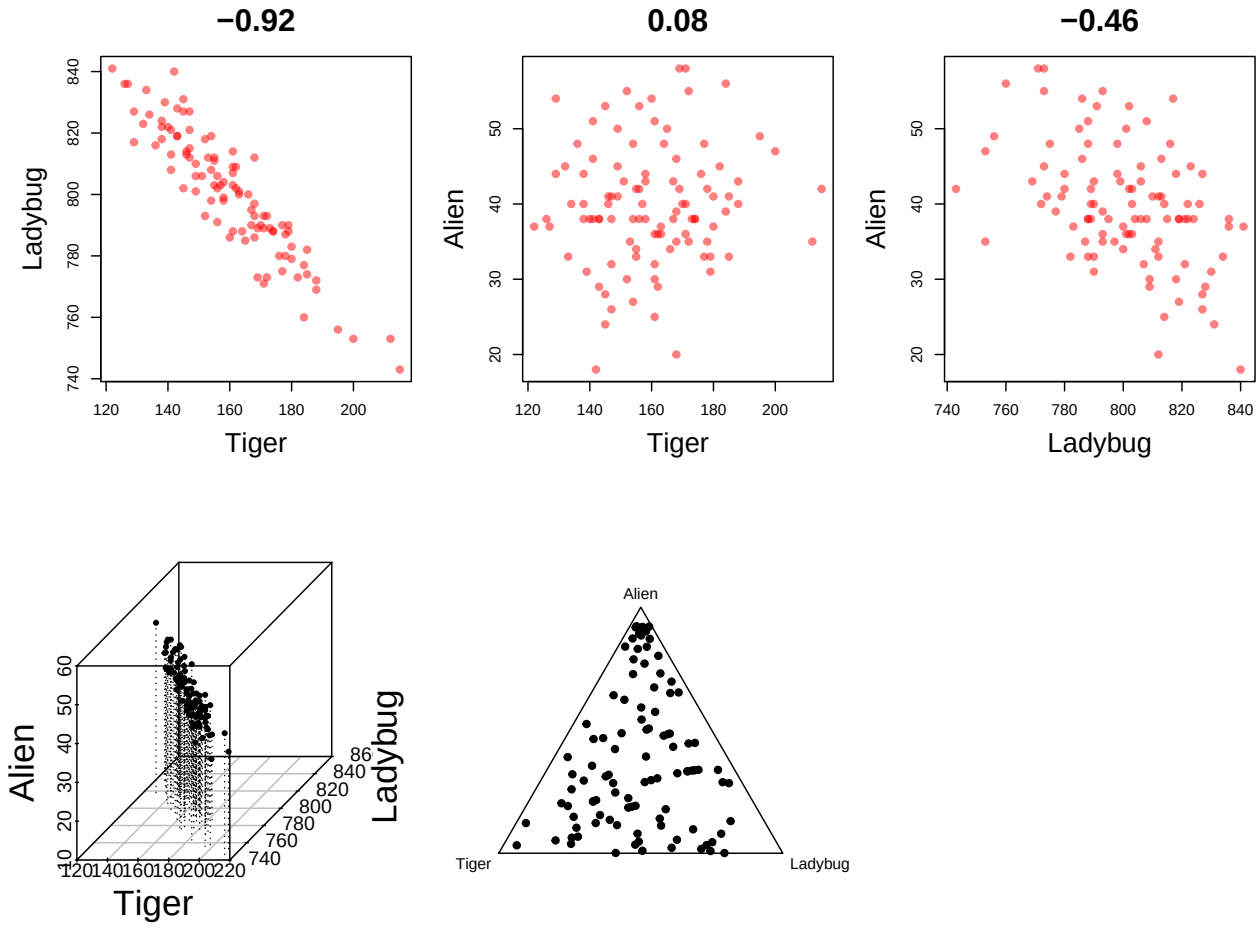

Figure 2: test

```
d <- read.table("../../working_papers/Foxman_rev/code/tongue_vs_cheek.txt",
    header=T, row.names=1)

d.tax <- data.frame(d$tax)
```

```
rownames(d.tax) <- rownames(d)

d.tax$genus <- gsub(".+;g__", "", d.tax[,1])
d$tax <- NULL

d.n0 <- cmultRepl(t(d), label=0, method="CZM") # all OTUs
```

```
## No. corrected values:  792304
```

```
d.clr <- codaSeq.clr(d.n0)
mean.clr <- apply(d.clr, 2, mean)
var.clr <- apply(d.clr,2, var)
plot(mean.clr, var.clr)
abline(h=median(var.clr), lty=3, col="grey", lwd=2)
```

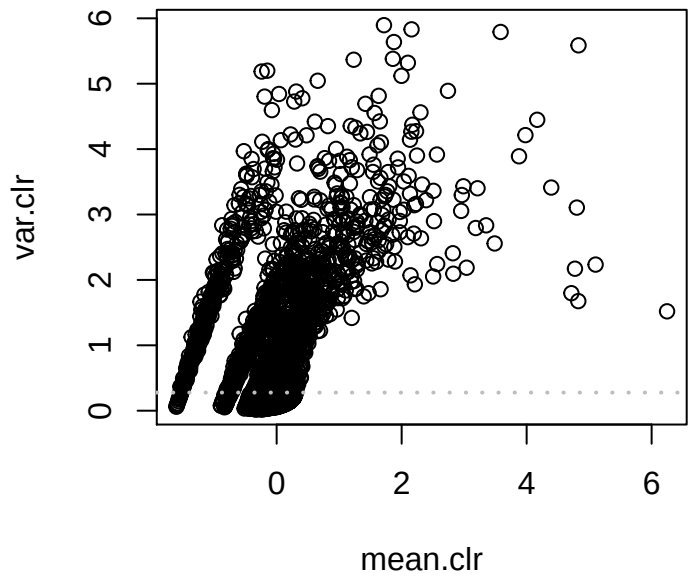

Figure 3: test

```
# determine outliers
if(plot==TRUE) pdf("outliers.pdf", height=10, width=6)
par(mfrow=c(2,1))
bm.outlier <- codaSeq.outlier(d.clr[1:187,], col=rgb(0,0,1,0.3))
td.outlier <- codaSeq.outlier(d.clr[188:366,], col=rgb(1,0,0,0.3))
```

```
if(plot==TRUE) dev.off()

#### this is the filtered dataset used as the base for everything that follows
# make sure all taxa have at least one count in a sample
d.good <- cbind(d[,bm.outlier$good], d[,td.outlier$good])
d.good.filt <- codaSeq.filter(d.good, min.reads=2000, min.prop=0,
    min.occurrence=0, samples.by.row=FALSE)

#### filter OTUs by variance greater than the median to make the high variance dataset
d.n0.good.filt <- cmultRepl(t(d.good.filt), label=0, method="CZM") # all OTUs
```

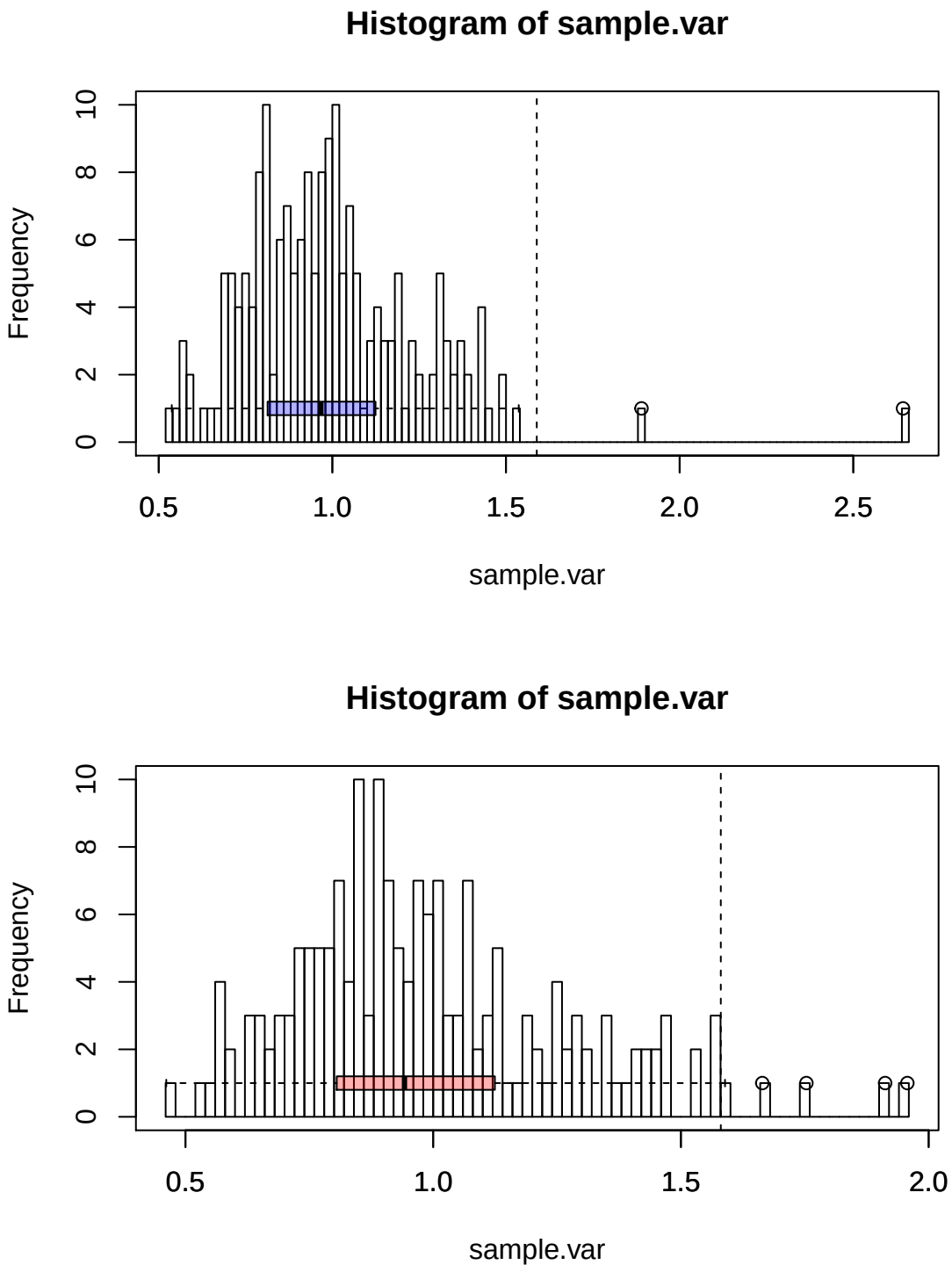

Figure 4: test

```
## No. corrected values:  594693
```

```
d.clr.good.filt <- codaSeq.clr(d.n0.good.filt)

mean.clr.good.filt <- apply(d.clr.good.filt, 2, mean)
var.clr.good.filt <- apply(d.clr.good.filt,2, var)

names.hvar <- names(var.clr.good.filt)[which(var.clr.good.filt >
    median(var.clr.good.filt))]

# this is the high variance count table
temp <- d.good.filt[names.hvar,]
d.hvar.good.filt <- codaSeq.filter(temp, min.reads=2000, min.prop=0,
    min.occurrence=0, samples.by.row=FALSE)

#### NOTE both of the below have a 2000 read cutoff applied
#### minimum 1% abundance dataset
d.filt.abund <- codaSeq.filter(d.good.filt, min.reads=2000, min.prop=0.01,
    min.occurrence=0, samples.by.row=FALSE)

#### minimum 50% sparsity dataset
d.filt.sparse <- codaSeq.filter(d.good.filt, min.reads=2000, min.prop=0,
    min.occurrence=.5, samples.by.row=FALSE)
```

```
if(plot==TRUE) pdf("sparsity_biplot.pdf", height=10, width=10)
par(mfrow=c(2,2))

#### entire good  dataset
pcx.g <- prcomp(d.clr.good.filt)
mv.g <- sum(pcx.g$sdev^2)
label.col <- c(rep("red", length(grep("^td", rownames(pcx.g$x)))),
    rep("blue", length(grep("^bm", rownames(pcx.g$x))) )
rownames(pcx.g$x) <- c(rep("td", length(grep("^td", rownames(pcx.g$x)))),
    rep("bm", length(grep("^bm", rownames(pcx.g$x))) )

# relationships between samples
coloredBiplot(pcx.g, cex=c(0.5,0.1), var.axes=FALSE,
    xlab=paste("PC1: ", round(sum(pcx.g$sdev[1]^2)/mv.g, 3)),
    ylab=paste("PC2: ", round(sum(pcx.g$sdev[2]^2)/mv.g, 3)),
    col="black", xlabs.col=label.col, scale=0,
    main="All", cex.lab=1.8, cex.main=1.5)
abline(v=0, lty=3, col="grey", lwd=1.5)
abline(h=0, lty=3, col="grey", lwd=1.5)

#### high variance plot
d.hv.n0 <- cmultRepl(t(d.hvar.good.filt), label=0, method="CZM") # all OTUs
```

```
## No. corrected values:  370430
```

```
d.clr.hv <- codaSeq.clr(d.hv.n0)

pcx.hv <- prcomp(d.clr.hv)
mv.hv <- sum(pcx.hv$sdev^2)
```

```
label.col <- c(rep("red", length(grep("^td", rownames(pcx.hv$x)))),
    rep("blue", length(grep("^bm", rownames(pcx.hv$x)))) )
rownames(pcx.hv$x) <- c(rep("td", length(grep("^td", rownames(pcx.hv$x)))),
    rep("bm", length(grep("^bm", rownames(pcx.hv$x)))) )

# relationships between samples
coloredBiplot(pcx.hv, cex=c(0.5,0.1), var.axes=FALSE,
    xlab=paste("PC1: ", round(sum(pcx.hv$sdev[1]^2)/mv.hv, 3)),
    ylab=paste("PC2: ", round(sum(pcx.hv$sdev[2]^2)/mv.hv, 3)),
    col="black", xlabs.col=label.col, scale=0,
    main="High Variance", cex.lab=1.8, cex.main=1.5)
abline(v=0, lty=3, col="grey", lwd=1.5)
abline(h=0, lty=3, col="grey", lwd=1.5)


#### minimum 1% abundance
d.n0.a <- cmultRepl(t(d.filt.abund), label=0, method="CZM") # all OTUs
```

```
## No. corrected values:  66589
```

```
d.clr.abund <- codaSeq.clr(d.n0.a)
# prcomp of filtered dataset
pcx.a <- prcomp(d.clr.abund)
mv.a <- sum(pcx.a$sdev^2)

label.col <- c(rep("red", length(grep("^td", rownames(pcx.a$x)))),
    rep("blue", length(grep("^bm", rownames(pcx.a$x)))) )
rownames(pcx.a$x) <- c(rep("td", length(grep("^td", rownames(pcx.a$x)))),
    rep("bm", length(grep("^bm", rownames(pcx.a$x)))) )
# relationships between samples
coloredBiplot(pcx.a, cex=c(0.5,0.1), var.axes=FALSE,
    xlab=paste("PC1: ", round(sum(pcx.a$sdev[1]^2)/mv.a, 3)),
    ylab=paste("PC2: ", round(sum(pcx.a$sdev[2]^2)/mv.a, 3)),
    scale=0, col="black", xlabs.col=label.col, main=">1% Abundance",
    cex.lab=1.8, cex.main=1.5)
abline(v=0, lty=3, col="grey", lwd=1.5)
abline(h=0, lty=3, col="grey", lwd=1.5)

#### minimum 50% sparsity
d.n0.s <- cmultRepl(t(d.filt.sparse), label=0, method="CZM") # all OTUs
```

```
## No. corrected values:  16394
```

```
d.clr.s <- codaSeq.clr(d.n0.s)
# prcomp of filtered dataset
pcx.s <- prcomp(d.clr.s)
mv.s <- sum(pcx.s$sdev^2)

label.col <- c(rep("red", length(grep("^td", rownames(pcx.s$x)))),
    rep("blue", length(grep("^bm", rownames(pcx.s$x)))) )
rownames(pcx.s$x) <- c(rep("td", length(grep("^td", rownames(pcx.s$x)))),
    rep("bm", length(grep("^bm", rownames(pcx.s$x)))) )
```

```
# relationships between samples
coloredBiplot(pcx.s, cex=c(0.5,0.1), var.axes=FALSE,
    xlab=paste("PC1: ", round(sum(pcx.s$sdev[1]^2)/mv.s, 3)),
    ylab=paste("PC2: ", round(sum(pcx.s$sdev[2]^2)/mv.s, 3)),
    scale=0, col="black", xlabs.col=label.col, main="<50% sparse",
    cex.lab=1.8, cex.main=1.5)
abline(v=0, lty=3, col="grey", lwd=1.5)
abline(h=0, lty=3, col="grey", lwd=1.5)
```

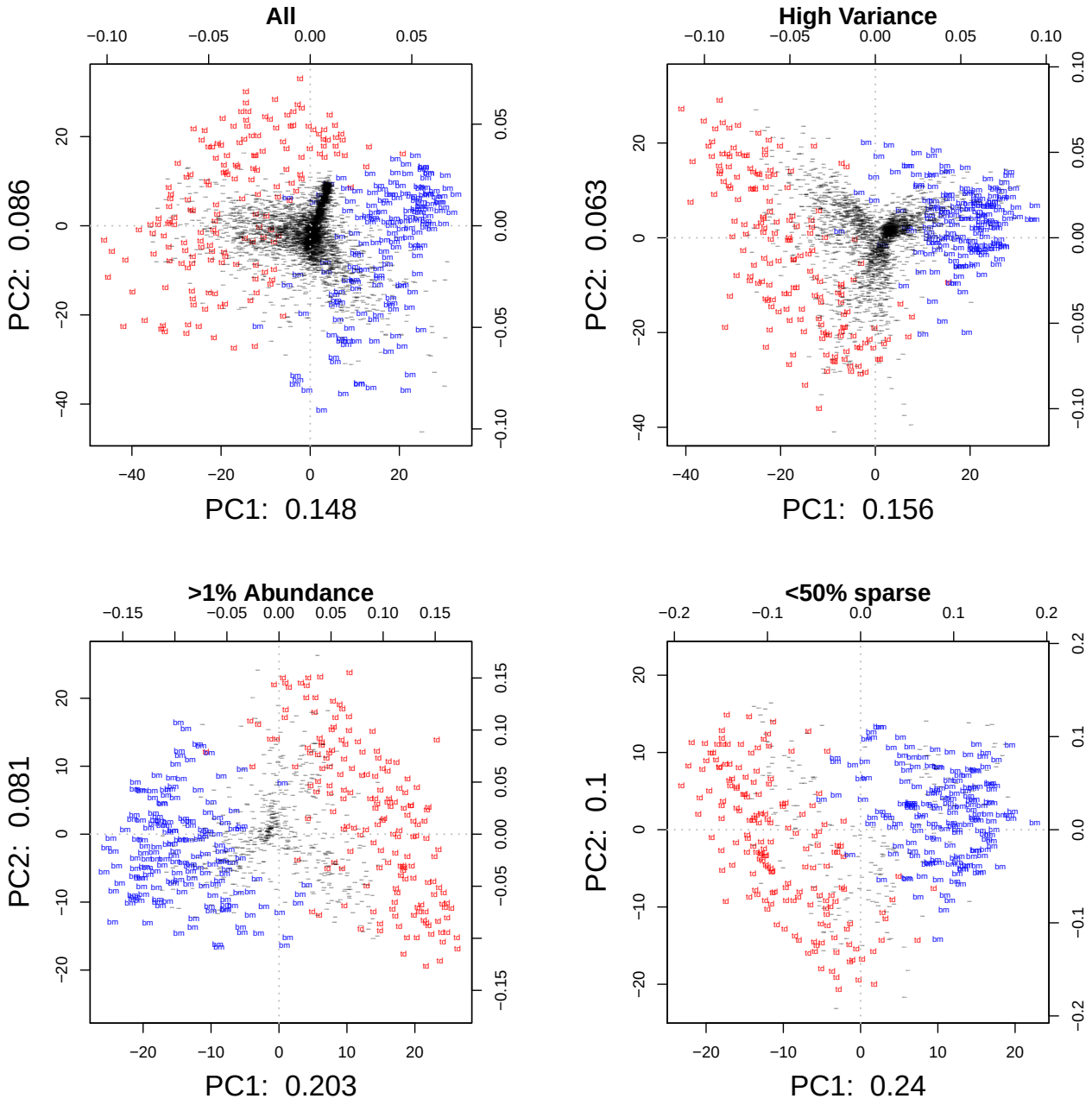

Figure 5: test

```
if(plot==TRUE)  dev.off()

#knitr::knit_exit()
```

ALDEx

```
x <- aldex.clr(d.good.filt, mc.samples=16)
```

```
## [1] "operating in serial mode"
```

```
conds <- c(rep("T", length(grep("^td", colnames(d.good.filt)))),
    rep("C", length(grep("^bm", colnames(d.good.filt)))) )
x.e <- aldex.effect(x, conds)
```

```
## [1] "operating in serial mode"
## [1] "sanity check complete"
## [1] "rab.all  complete"
## [1] "rab.win  complete"
## [1] "rab of samples complete"
## [1] "within sample difference calculated"
## [1] "between group difference calculated"
## [1] "group summaries calculated"
## [1] "effect size calculated"
## [1] "summarizing output"
```

```
if(plot==TRUE) pdf("aldex.pdf", height=5, width=5)
sigT <- which(x.e$effect > 0.8)
sigC <- which(x.e$effect < -0.8)
plot(x.e$diff.win, x.e$diff.btw, pch=19, cex=0.5, col=rgb(0,0,0,0.1),
    xlim=c(0.2,6), xlab="Dispersion (log2)", ylab="Difference (log2)")
text(x.e$diff.win[sigT], x.e$diff.btw[sigT], labels=rownames(x.e)[sigT],
    col="red", cex=0.5)
text(x.e$diff.win[sigC], x.e$diff.btw[sigC], labels=rownames(x.e)[sigC],
    col="blue", cex=0.5)

abline(0,1, lty=3)
abline(0,-1, lty=3)
```

```
if(plot==TRUE) dev.off()
```

```
phi.sma.df <- propr.aldex.phi(x)

# find the set of connections with phi less than some value
d.lo.phi <- subset(phi.sma.df, phi<0.275)

# generate a graphical object
```

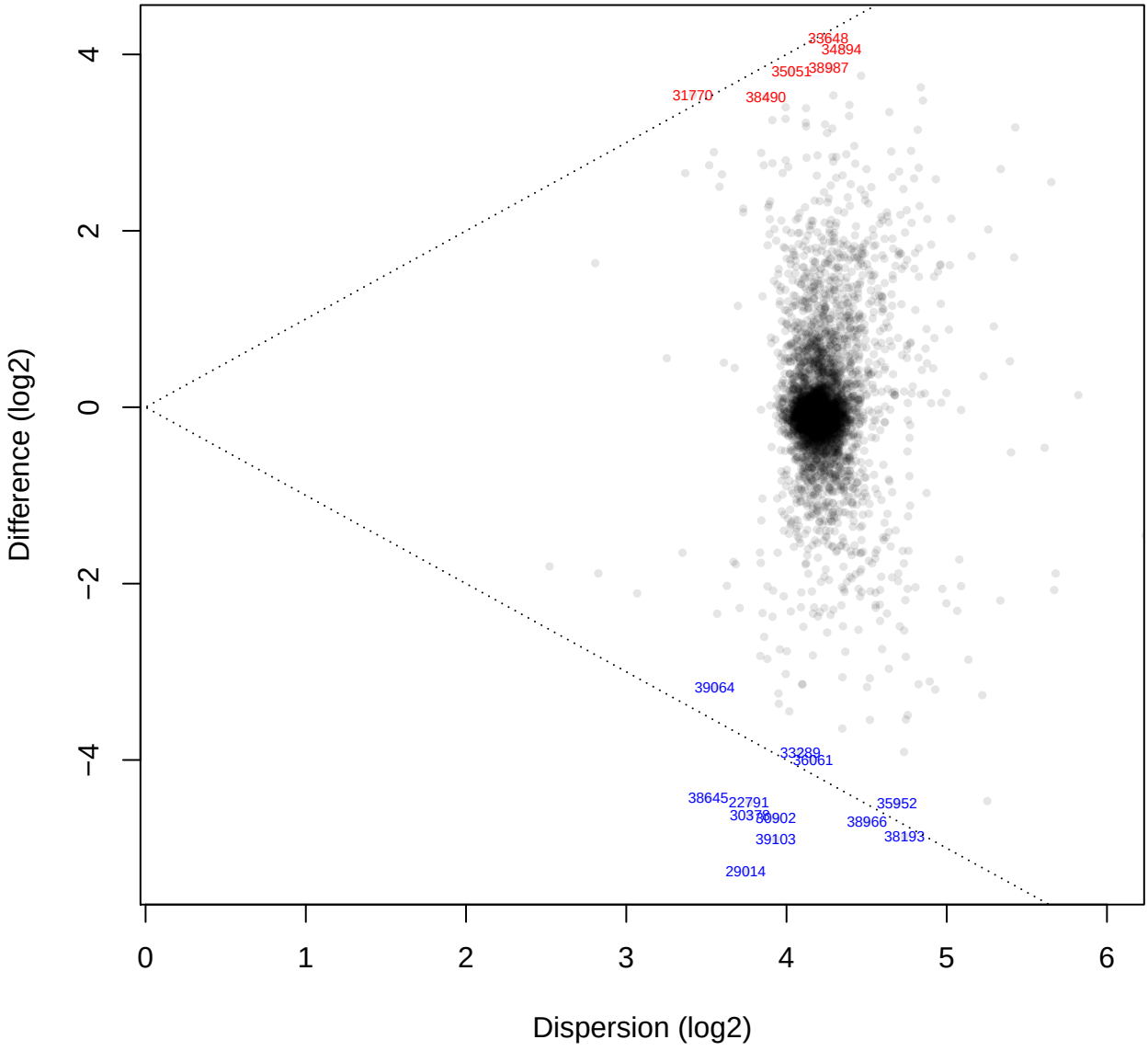

Figure 6: test

```
g <- graph.data.frame(d.lo.phi, directed=FALSE)
# get the clusters from the graph object
g.clust <- clusters(g)

# data frame containing the names and group memberships of each cluster
g.df <- data.frame(ID=V(g)$name, cluster=g.clust$membership,
    size=g.clust$csize[g.clust$membership])

g.df$genus=d.tax[V(g)$name,2]

col=rainbow(max(g.df$cluster))
g.df$col <- col[as.numeric(g.df$cluster)]

g.ids <- V(g)$name
g.names <- d.tax[V(g)$name,2]
V(g)$name <- g.names
V(g)$color <- g.df$col

###
# plot the taxa with phi < 0.25
label.col <- c(rep(rgb(1,0,00,0.25), length(grep("^td", rownames(pcx.g$x)))),
    rep(rgb(0,0,1,0.25), length(grep("^bm", rownames(pcx.g$x)))) )

if(plot==TRUE) pdf("phi_plot.pdf", height=5, width=12)
par(mfrow=c(1,3), mar=c(5,5,4,1))
coloredBiplot(pcx.g, cex=c(0.5,0.1), var.axes=FALSE,
    xlab=paste("PC1: ", round(sum(pcx.g$sdev[1]^2)/mv.g, 3)),
    ylab=paste("PC2: ", round(sum(pcx.g$sdev[2]^2)/mv.g, 3)),
    scale=0, col="black", xlabs.col=label.col, main="All", cex.lab=1.8)
abline(v=0, lty=3, col="grey")
abline(h=0, lty=3, col="grey")
for(i in 1:max(g.df$cluster)){
vec <- as.vector(g.df[g.df$cluster==i,"ID"])
points(pcx.g$rotation[vec,1], pcx.g$rotation[vec,2], col=col[i])
}

plot(g, layout=layout.fruchterman.reingold.grid(g, weight=0.05/E(g)$phi),
    vertex.label.color="black", vertex.size=5, vertex.color=V(g)$color)
# vertex.label.color=V(g)$color,

plot(x.e$diff.win, x.e$diff.btw, pch=19, cex=0.5, col=rgb(0,0,0,0.1),
    xlim=c(0.2,6), xlab="Dispersion (log2)", ylab="Difference (log2)")

for(i in 1:max(g.df$cluster)){
vec <- as.vector(g.df[g.df$cluster==i,"ID"])
points(x.e[vec,"diff.win"], x.e[vec,"diff.btw"], col=col[i])
}
#clip(0,0,5,5)
abline(0,1, lty=3)
abline(0,-1, lty=3)
```

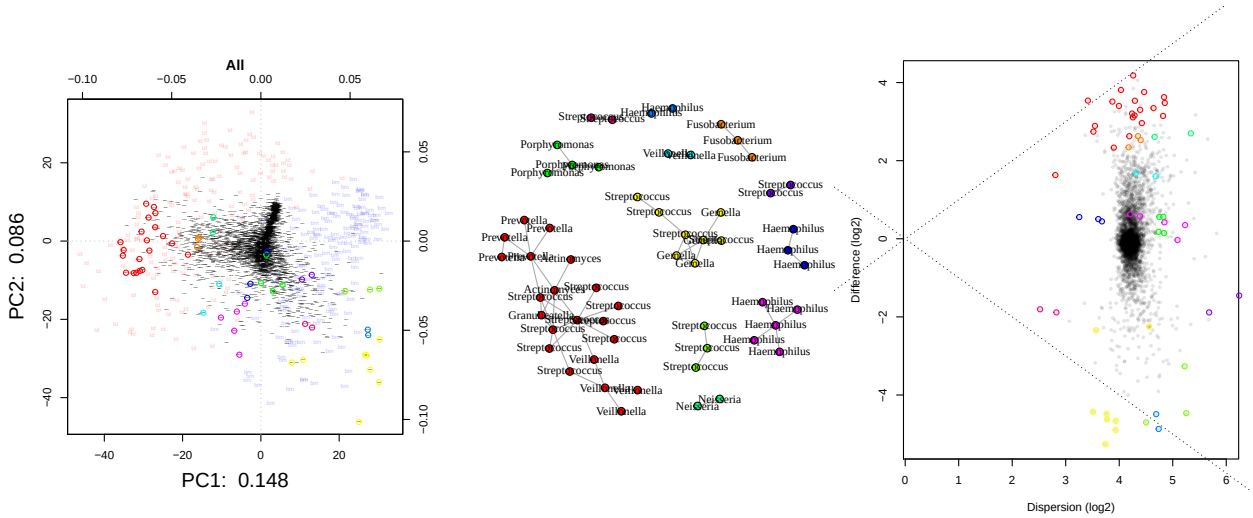

Figure 7: test

```
if(plot==TRUE) dev.off()
```