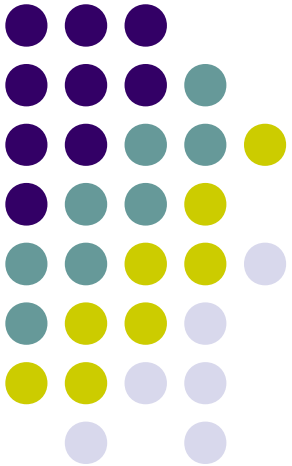
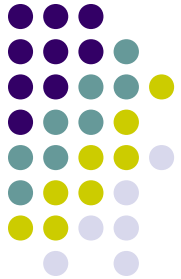
Lecture #3

Data visualizations, outliers, and missing data

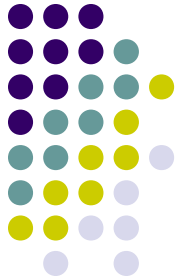


Outline



- Data visualizations
  - Univariate vs. multivariate
- Outliers
  - Detection
  - Visualizations
  - Methods to handle outliers
- Missing data
  - Average value imputation
  - Weighted *k*-nearest neighbor
  - SVD impute method
- Summary

Visualizing microarray data

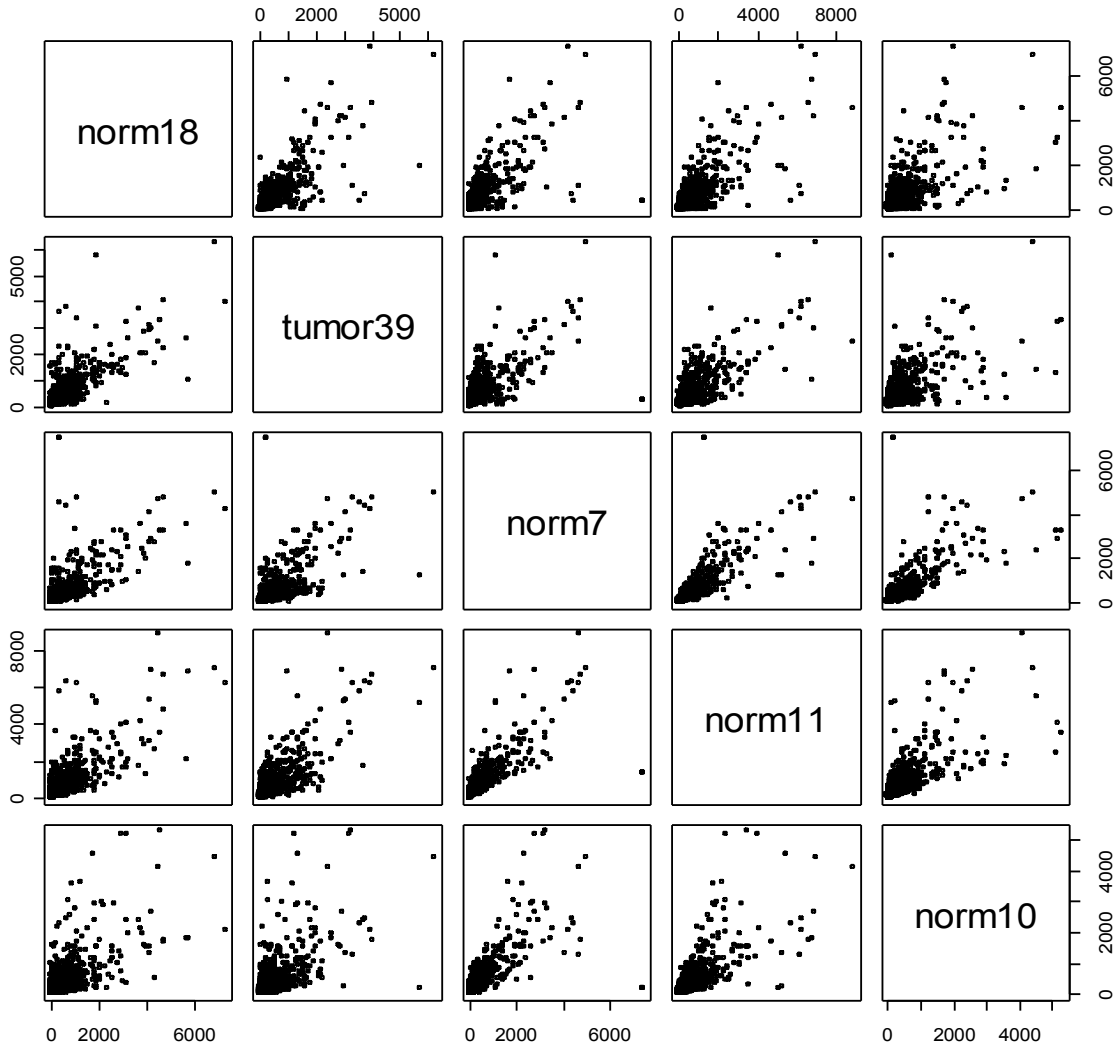


- Visualizing microarray data is a more difficult task than visualizing other single variable data formats
  - Multivariate (genes)
  - Dimensionality

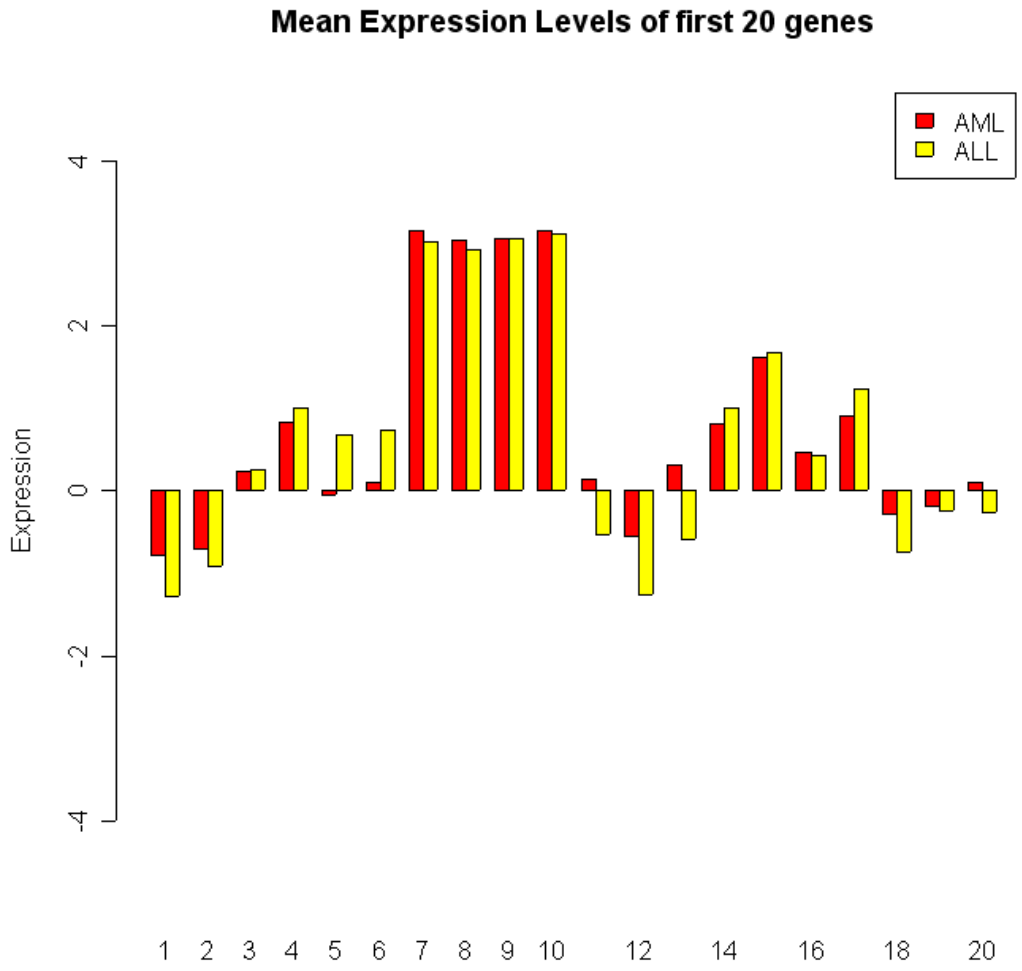
- Many of the traditional methods to view data in the univariate world have to be adjusted to encompass all of the variables in a multivariate space
  - Scatter plots
  - Dot plots
  - Histograms
  - Bar plots

- We can do this in multiple ways
  - Treat variables (genes or samples) as vectors in n-dimensional space
  - Use linear combinations of the variables

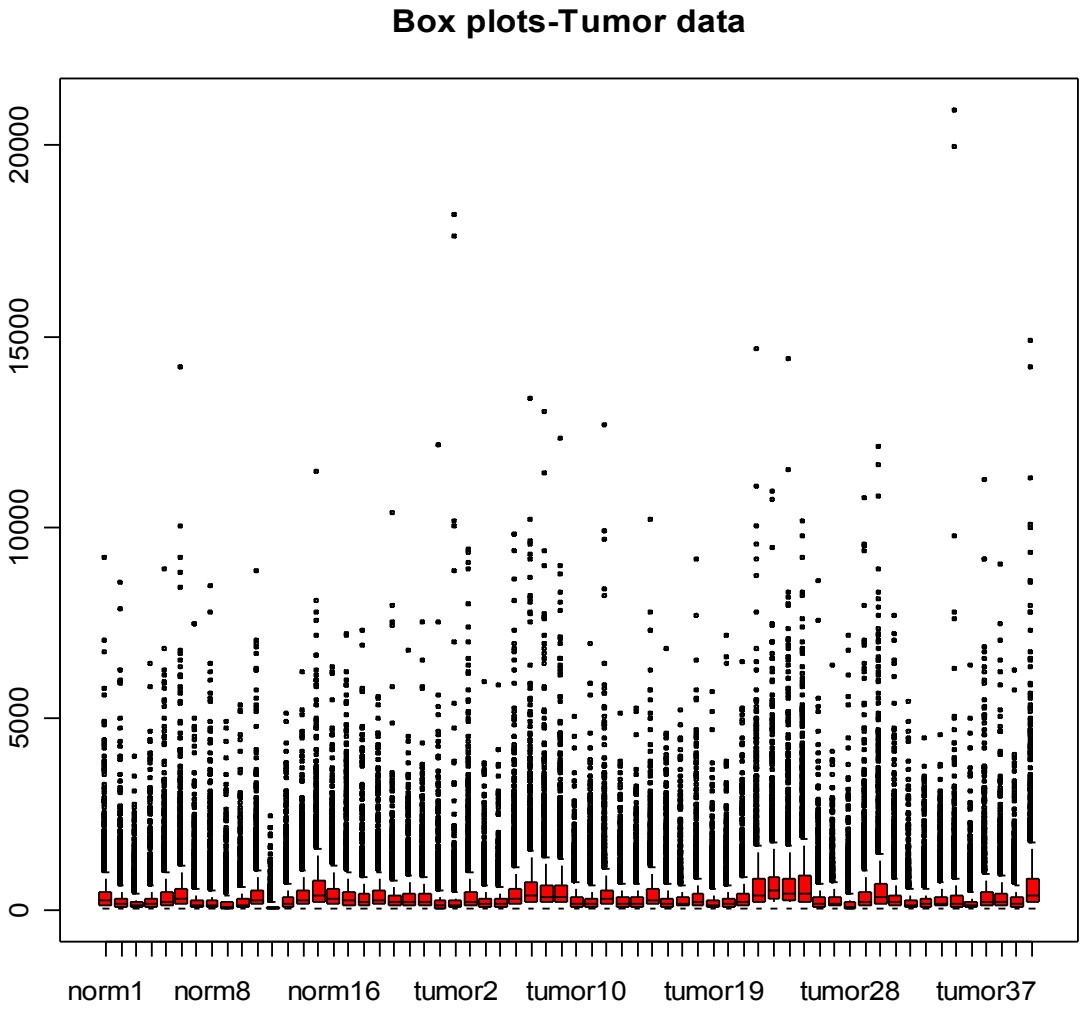
Scatter plot matrix (select samples)



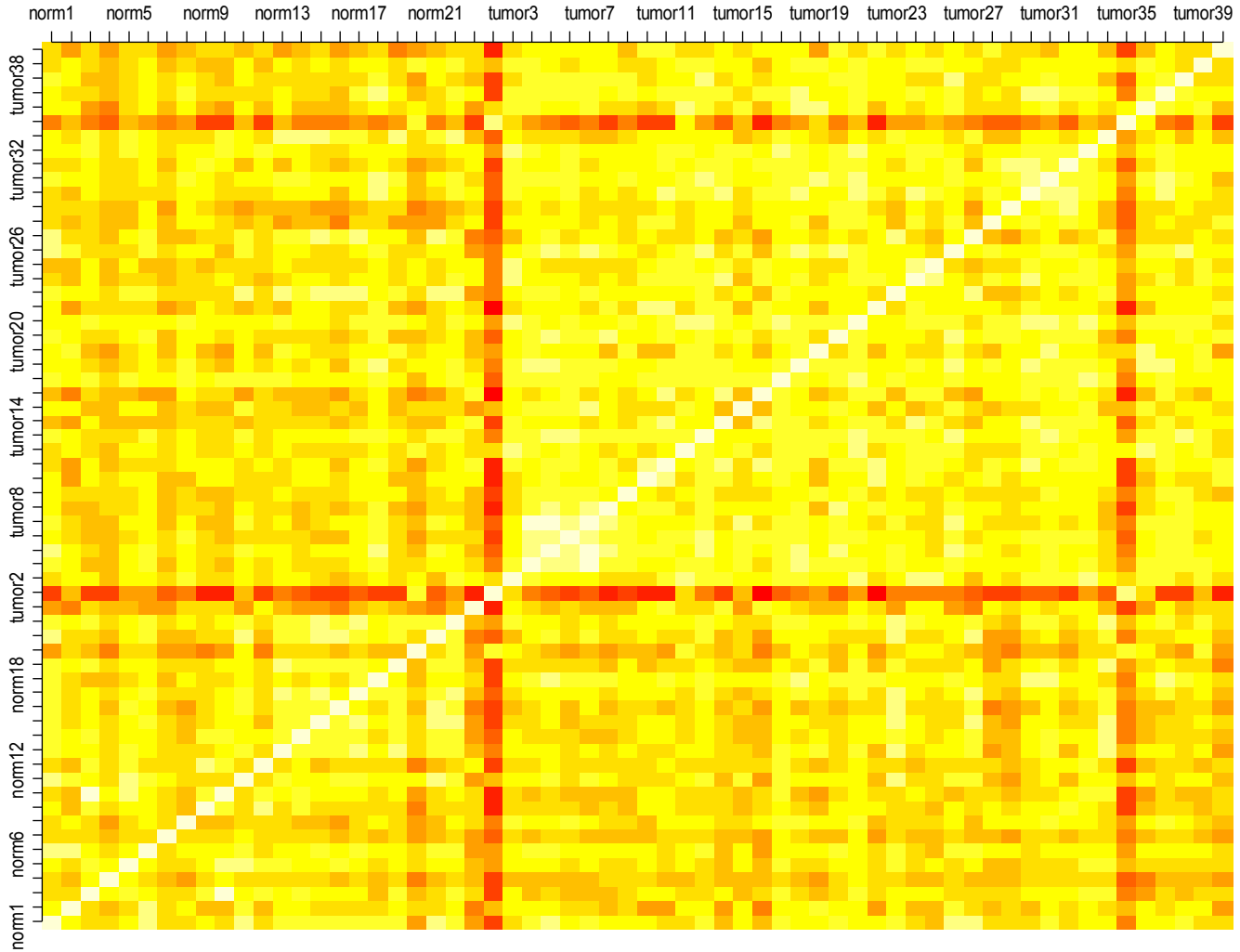
Bar plot matrix (select genes)



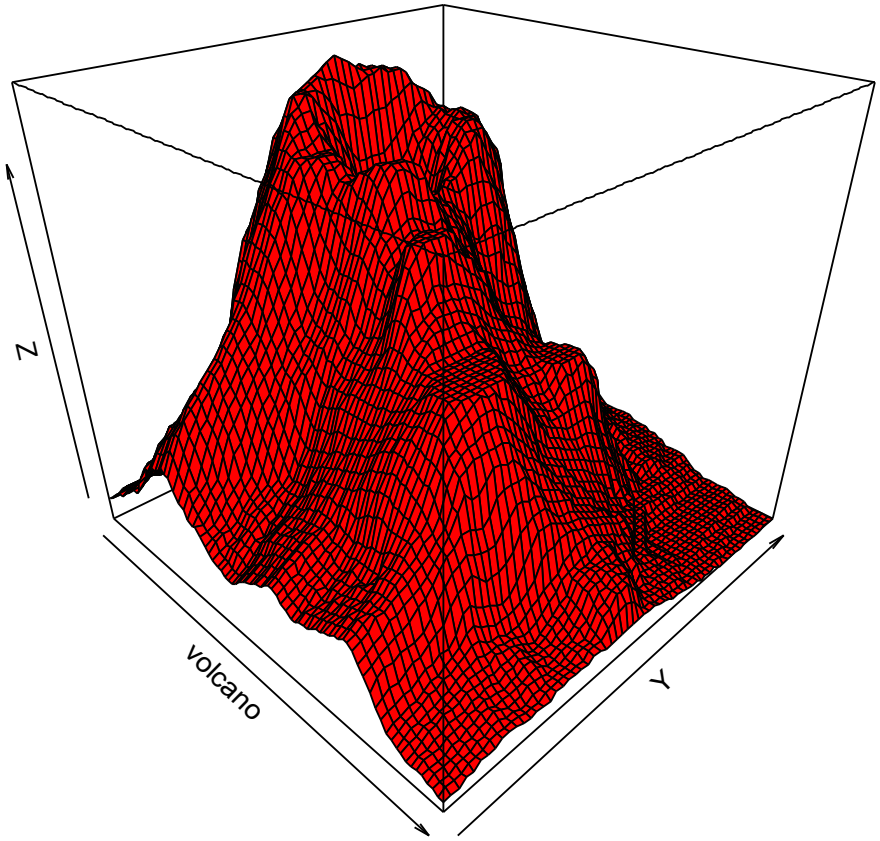
Box-Whisker Plots (all samples)



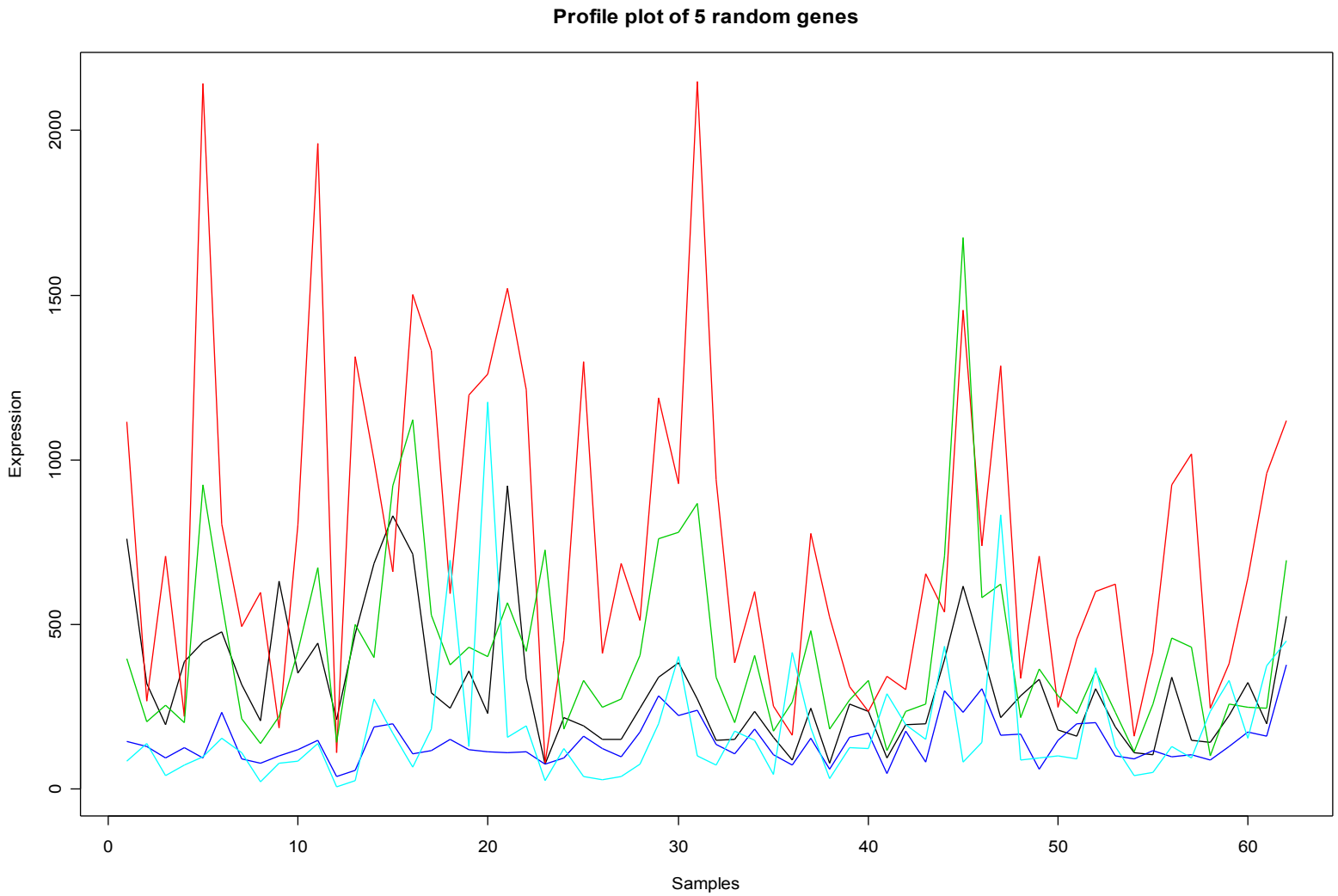
Correlation matrix (all samples)



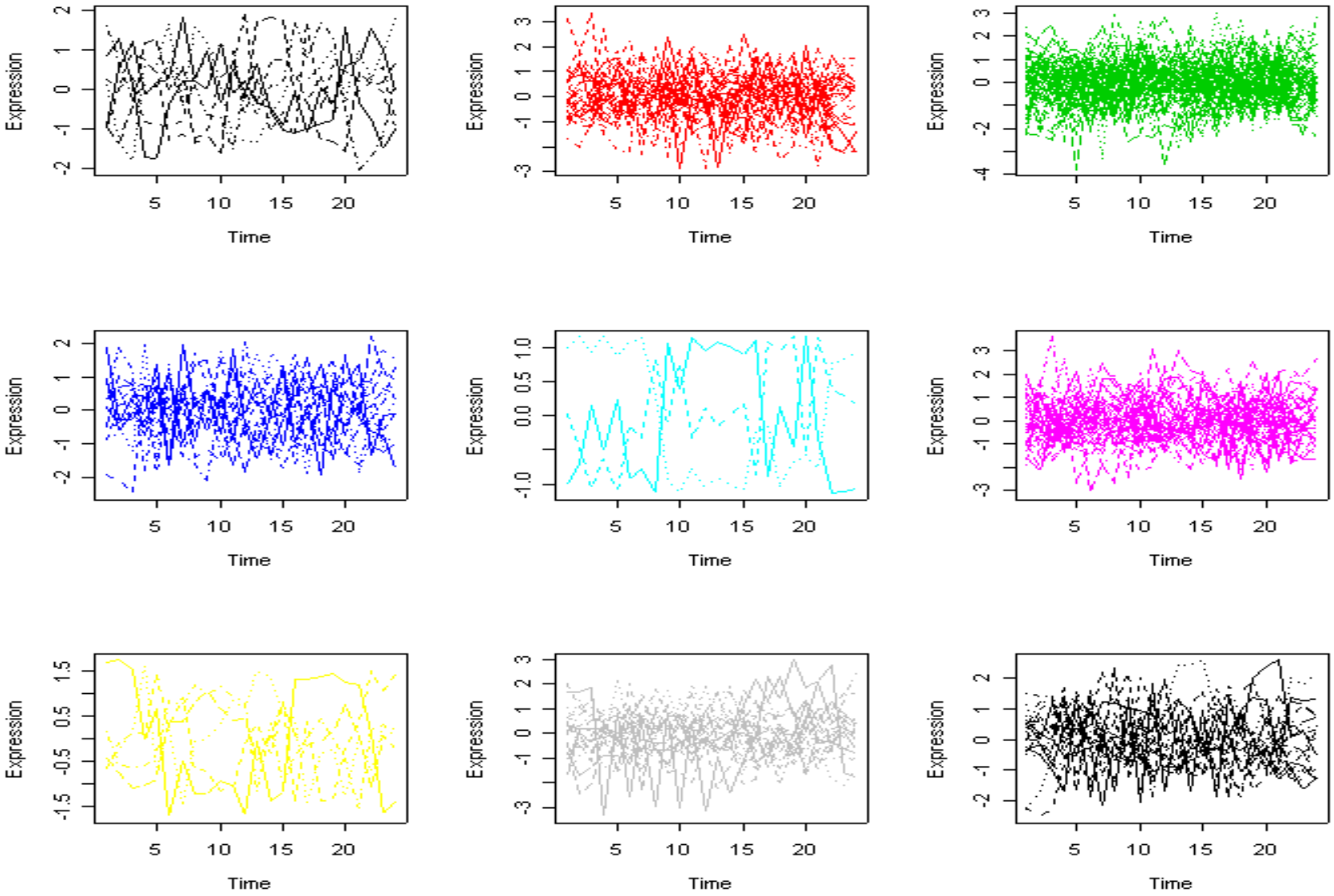
Perspective plot (select samples)



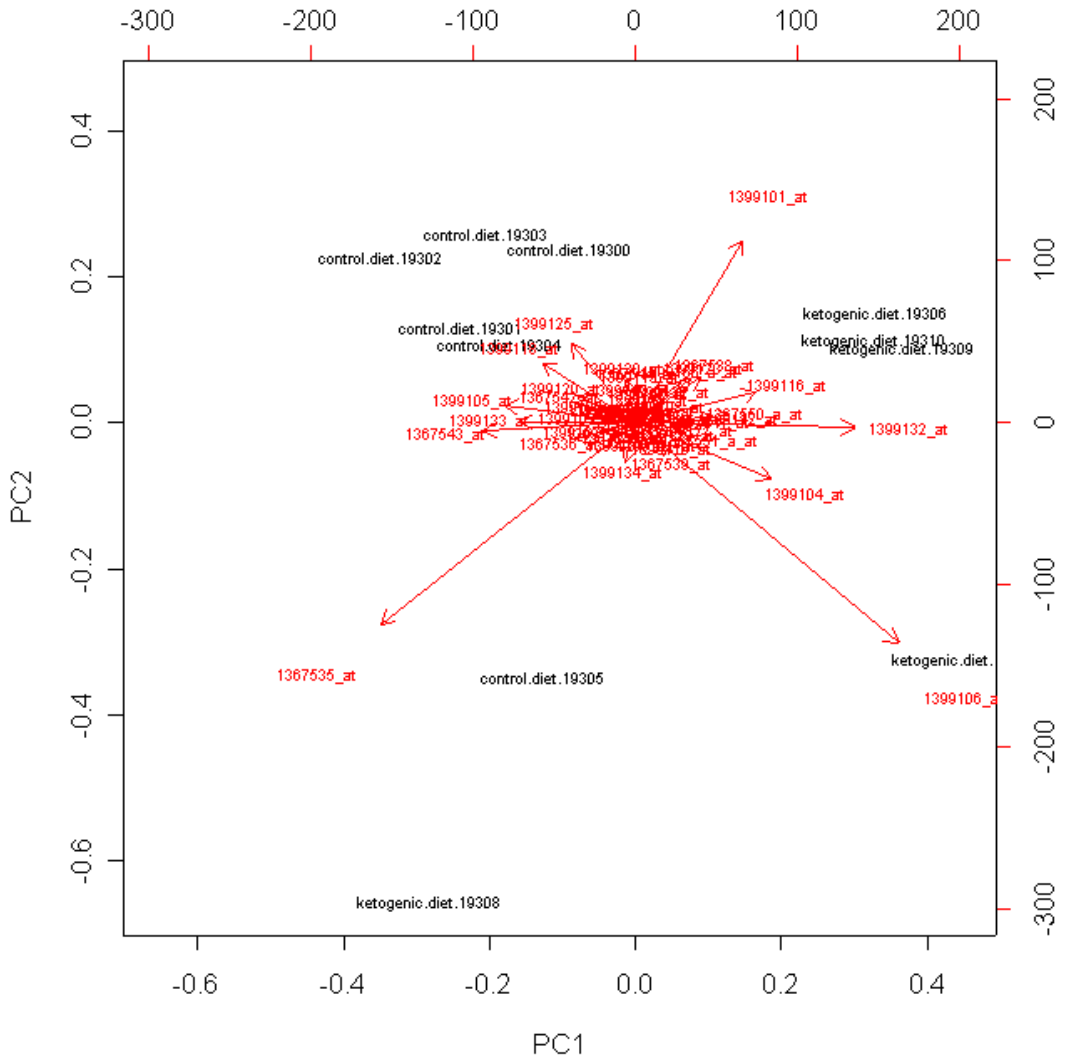
Gene profile plot (select genes)



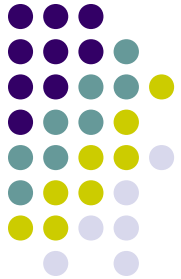
Cluster profile plots (all genes)



PCA biplot (linear combination of samples and genes)



Outlier identification



- Outliers are variables that may have aberrant values, causing undesired effects on the data

- Multiple causes for outlier samples in microarray data
  - Chip manufacturing
  - Degraded RNA sample
  - Non-responsive patient/animal
  - etc.

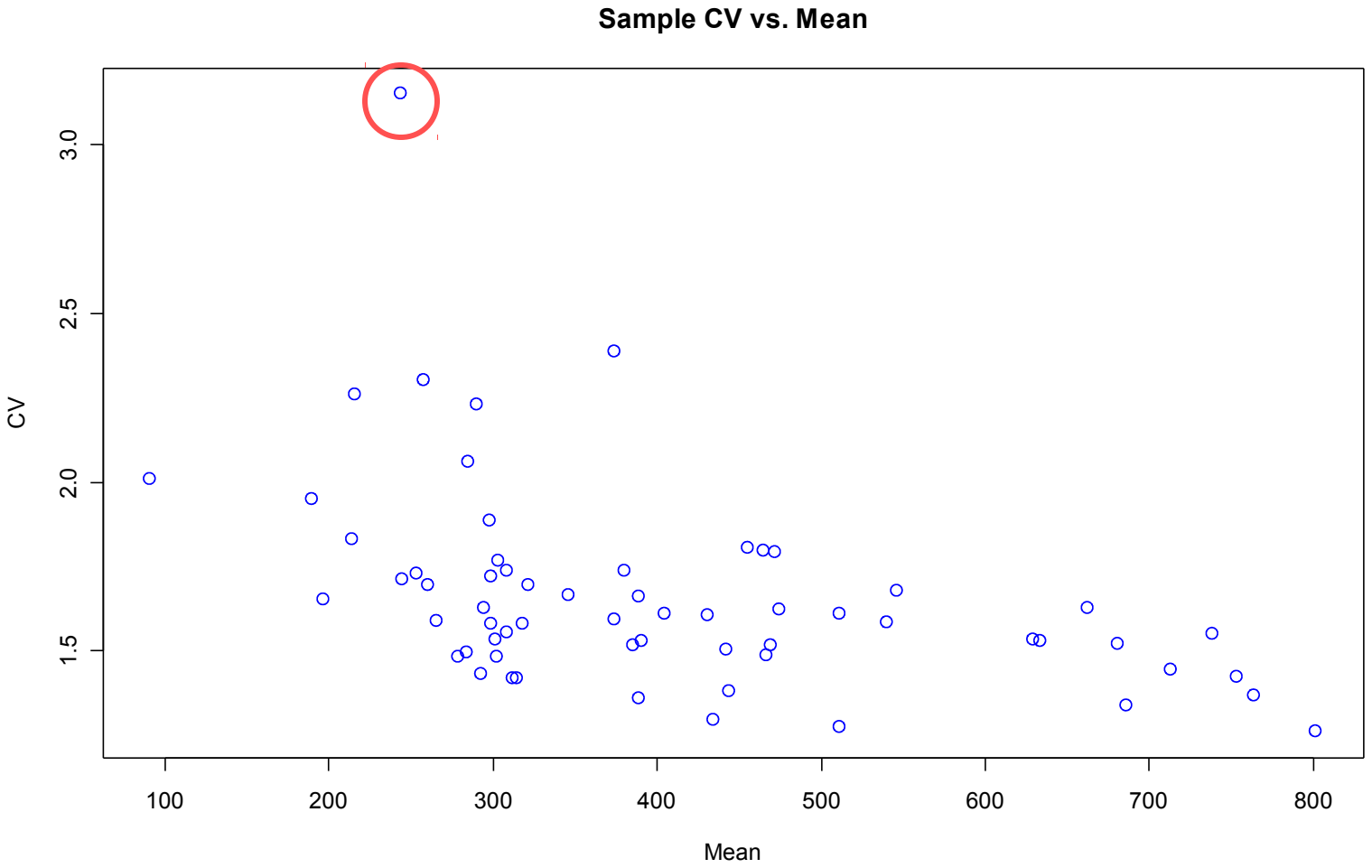
- Equally many causes for outlier genes
  - Probe specificity
  - Chip artifacts in certain regions
  - Signal/noise threshold (low expressers)
  - etc.

Outlier identification

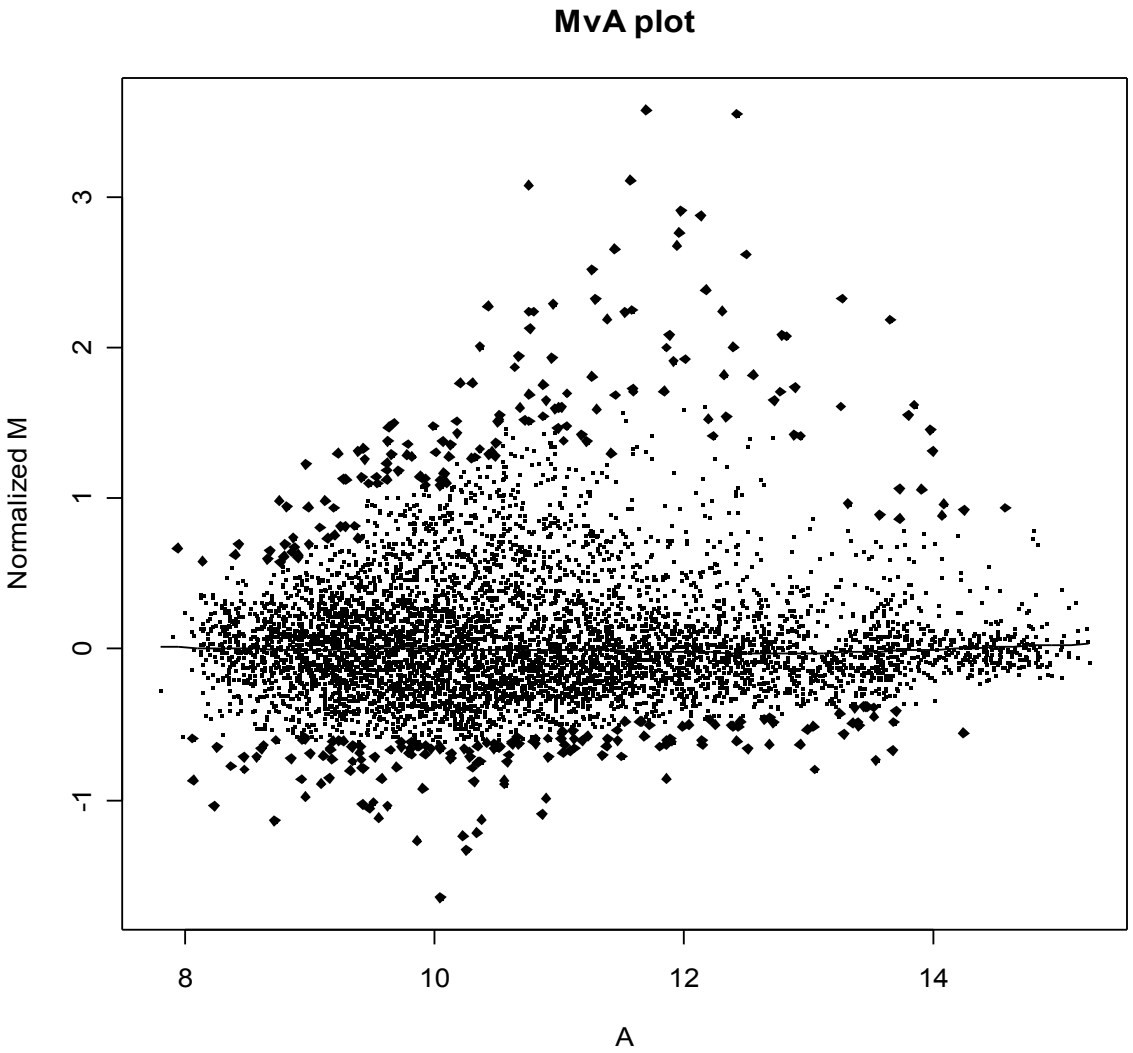
- Multiple visualizations can detect outlier samples
  - Coefficient of variation (cv) plot
  - MvA plot
  - PCA plot
  - Correlation heat map
  - Clustering dendrogram

CV vs. Mean plot

- Sample outlier

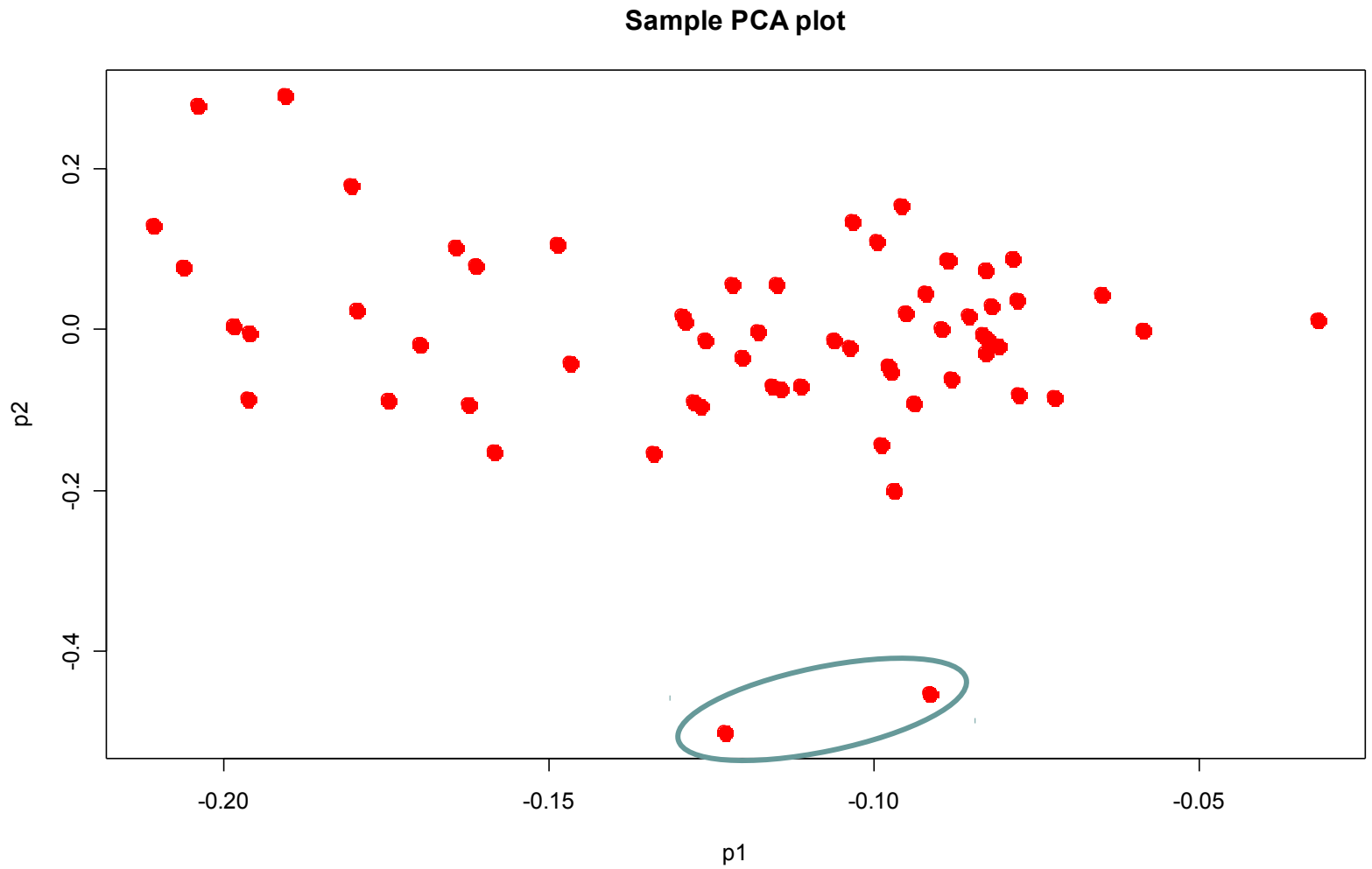


MvA plot

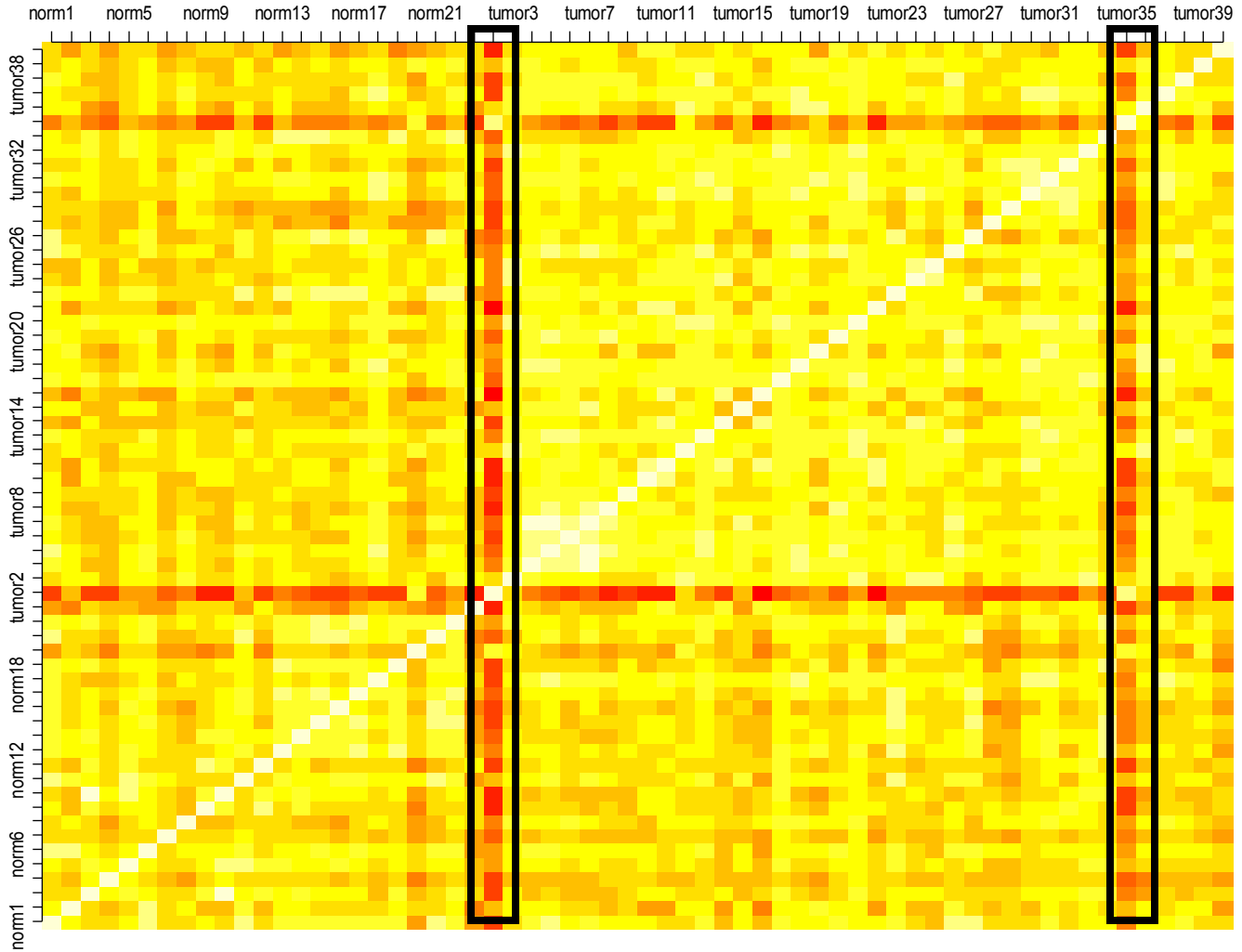


Sample PCA plot

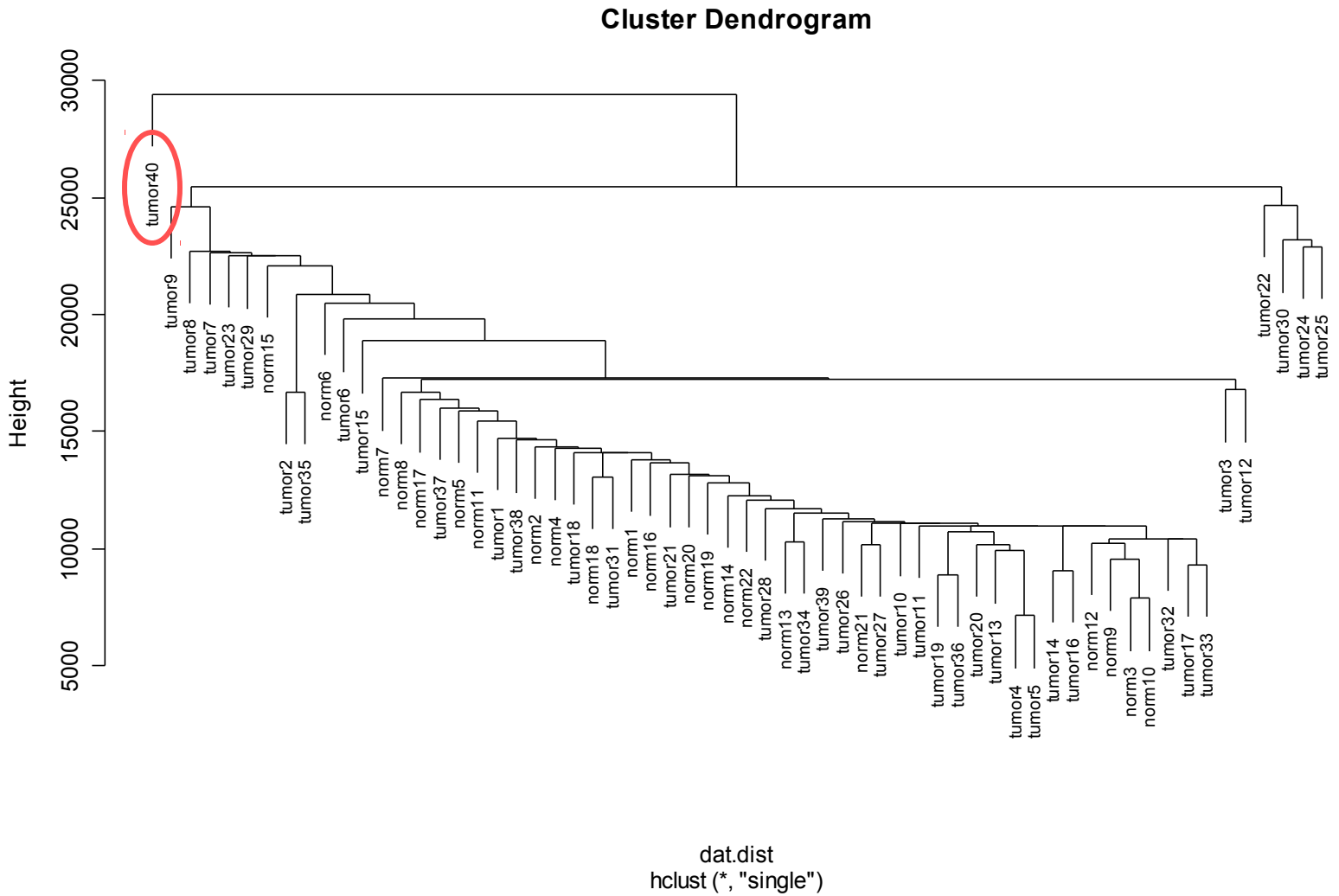
- Sample outliers



Sample correlation matrix



Sample Clustering Dendrogram



Dealing with Outliers

- Sample and gene outliers can contribute to incorrect statistical inference
  - The sample variance is inflated, which is not indicative of the true variance
- Methods for dealing with outliers
  - Trimmed mean/median (exclude a specified percentage of points)
  - Median (less sensitive to very high or low expressing outliers)
  - Quantiles (use the IQR or q1/q3)
  - Simply remove from data prior to calculations

Missing Data

- Missing data occurs when there is no expression value associated with a particular array spot
  - cDNA arrays usually have many missing data points
  - A blank is left in the cell, where the expression value should be
  - Most analysis methods require gene/sample vectors of equal rank to compute distances, scores, etc.
  - Must find method of either extrapolating or interpolating these values

- Three methods discussed is the Troyanskaya O et. al. paper to assess missing values
  - Mean value calculation (majority rules)
  - Weighted *k*-nearest neighbor (KNN) interpolation
  - Singular value decomposition (SVD) impute method

Mean Value

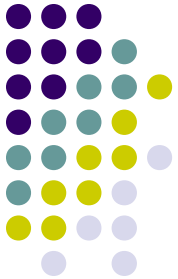
- For a particular gene, compute the average value of the existing data points and use this value as the missing value
  - Quickest and easiest method

- Problems
  - Multiple missing values per gene will have the same expression[1]
  - Ignore correlation structure in the data[1]
  - Outlier samples can alter the mean[1]

k-Nearest Neighbor

- For the missing experiment value in gene A, compute the KNN for all genes across 2-N experiments (where N is total number of experiments)
  - Cluster the genes, omitting experiment #1
  - Use Euclidean distance
- Determine cluster where gene A belongs
  - Call this cluster X
- Use Euclidean distances from genes in cluster X to gene A as percentages
- Calculate weighted mean of genes in cluster X (except gene A), including missing experiment value

k-Nearest Neighbor example



- Data
  - Alon et al. colon cancer data set
- Artificially remove experiment #1 value from gene #2
- Perform KNN missing data imputation to determine value

Predicted value = 4893.953

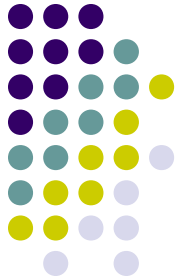
Actual value = 4883.449

Relative error = |(4883.449-4893.953)| / |4883.449| = 2.2e-3

SVD impute method

- First use the row average to fill in the gene with missing values
  - SVD can only be performed on complete matrices
- *Calculate the characteristic roots (eigenvalues) from the gene correlation or covariance matrix
- *Compute the corresponding characteristic vectors (eigenvectors) and sort by most significant characteristic roots
- Regress gene *i* against the *k* characteristic vectors
- Use the coefficients from the regression model to get the missing value in gene *i*
  - Linear combination

*we will visit this later in the semester in PCA



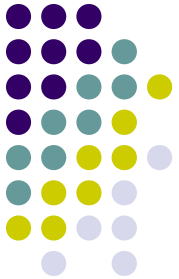
Summary

- Data visualizations
  - Very useful in understanding multivariate data
- Outliers
  - Multiple methods to deal with them to improve statistical inference
- Missing data
  - The examples mentioned here are only a couple of approaches
  - Much continuing work in this area

References

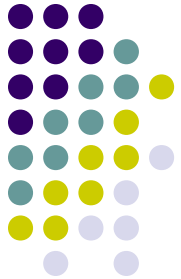
- [1]Troyanskaya O, Cantor M, Sherlock G, Brown P, Hastie T, Tibshirani R, Botstein D, and Altman R. (2001) Missing value estimation methods for DNA microarrays. *Bioinformatics*. **17**, 520-525.

R Code



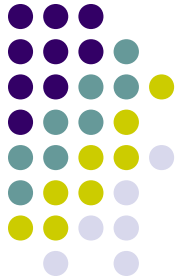
```
# scatter plot matrix
dat <- read.table("gecolon.dat",header=T)
dimnames(dat)[[1]] <- as.character(dat[,1])
dat <- dat[,-1];         dat <- as.data.frame(dat);

# other data sets in R to use
library(Biobase);        library(annotate);       library(golubEsets);
data(golubTrain);         data(golubTest);        data(geneData);
dat <- geneData or dat <- exprs(golubTrain) or dat <- exprs(golubTest)

# box plots
boxplot(dat,cex=0.45,col='red',main="Box plots-Tumor data")

# random selection of 5 samples
rand.sams <- sample(names(dat),5,replace=F)
# plot trellis
pairs(dat[,rand.sams])

# Pearson's correlation matrix
dat.cor <- cor(dat)
image(dat.cor,axes=F)
axis(2,at=seq(0,1,length=ncol(dat.cor)),label=dimnames(dat.cor)[[2]])
axis(3,at=seq(0,1,length=ncol(dat.cor)),label=dimnames(dat.cor)[[2]])

# random sample of 5 genes
rand.genes <- sample(dimnames(dat)[[1]],5,replace=F)

# profile plot
plot(c(1,ncol(dat)),range(dat[rand.genes,]),type='n',main="Profile plot of 5 random
     genes",xlab="Samples",ylab="Expression")
for(i in 1:length(rand.genes)) {
     dat.y <- as.numeric(dat[rand.genes[i],])
     lines(c(1:ncol(dat)),dat.y,col=i)
}
```

R Code



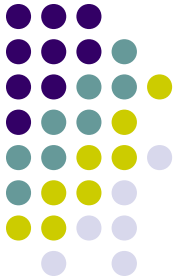
```
# load the yeast cell cycle data set
dat <- read.table("spellman.txt",header=T)
dimnames(dat)[[1]] <- as.character(dat[,1])
dat <- dat[,-1]
dat <- dat[,23:46]
dat[is.na(dat)] <- 0

# pca biplot
biplot(prcomp(t(dat[500:550,])),cex=0.6)

# k-means cluster profiles
dd <- dat[names(f.p)[f.p<0.001],]
d.k <- kmeans(dd,9)
par(mfrow=c(3,3))
for(i in 1:9) {
     tmp <- scale(dd[d.k$cluster==i,])
     matplot(c(1:ncol(dat)),t(tmp),type='l',col=i,xlab='Time',ylab='Expression')
}
# cv vs. mean plot
dat.mean <- apply(dat,1,mean)                    # calculate mean for each gene
dat.sd <- sqrt(apply(dat,1,var))                 # calculate st.deviation for each gene
dat.cv <- dat.sd/dat.mean                        #calculate cv

plot(dat.mean,dat.cv,main="Sample CV vs. Mean",xlab="Mean",ylab="CV",col='blue',cex=1.5)

# 2D sample pca plot
dat.pca <- prcomp(t(dat))
dat.loads <- dat.pca$x[,1:2]
plot(dat.loads[,1],dat.loads[,2],main="Sample PCA plot",xlab="p1",ylab="p2",col='red',cex=1.5,pch=16)
```

R Code



```
# k-means clustering for missing value imputation
dat <- dat[2:30,]                                              # only use 29 genes for example
cl <- kmeans(dat[,-1],centers=5, iter.max=20)                  # cluster into 5 groups
                                                  # we pretend to be missing a value at sample#1 gene #2
groups <- cl$cluster                              # get cluster membership for each gene
groups                                            # look at groups to see where gene 2 is
group.2 <- groups==2                              # since gene 2 is in group 2, get all other members
genes.cluster <- dimnames(dat)[[1]][group.2]
genes.cluster                                     # look at all other genes in cluster #2

gene.dist <- dist(dat[genes.cluster,-1],method="euclidean") # get distances from genes in cluster 2 to
                                                           # gene #2
gene.dist <- as.matrix(gene.dist)
gene.dist <- gene.dist[2:5,1]
gene.weight <- as.numeric(gene.dist/sum(gene.dist))        # get weights for each gene

weight.mean <-  weighted.mean(dat[genes.cluster[-1],1], gene.weight)    # calculate weighted mean for
                                                                        # gene #2


# perspective plot
data(volcano)           # load volcano data set
persp(volcano, theta=45, phi=30, col="red")

# MvA plot
library(sma)
data(MouseArray)
mouse.lratio <- stat.ma(mouse.data, mouse.setup)
plot.mva(mouse.data, mouse.setup, norm="l", 2, extra.type="pci", plot.type="n",main="MvA plot")
```

R Code



```
# calculate mean for some genes, with respect to class
library(multtest)
data(golub)
dat <- as.data.frame(golub)
ann <- golub.cl
dat.aml <- apply(dat[,ann==1],1,mean)
dat.all <- apply(dat[,ann==0],1,mean)
tab <- data.frame(rbind(dat.aml[1:20],dat.all[1:20]))
dimnames(tab)[[1]] <- c("AML","ALL")
names(tab) <- dimnames(dat)[[1]][1:20]
mp <- barplot(tab)
tot <- colMeans(tab)
text(mp, tot + 3, format(tot), xpd = TRUE, col = "blue")
barplot(as.matrix(tab),beside=T,col=c("red","yellow"),legend=rownames(as.matrix(tab)),ylim=c(-
    5,5),ylab="Expression")
title(main = "Mean Expression Levels of first 20 genes")

# cluster tree
dat <- t(dat)                                    #transpose dat
dat.dist <- dist(dat,method="euclidean")         # calculate distance
dat.clust <- hclust(dat.dist,method="single")    # calculate clusters
plot(dat.clust,labels=names(dat),cex=0.75)                  # plot cluster tree
```