Clustering and Visualization using R

陽明大學生物資訊研究所
2005 微陣列數據分析暑期課程

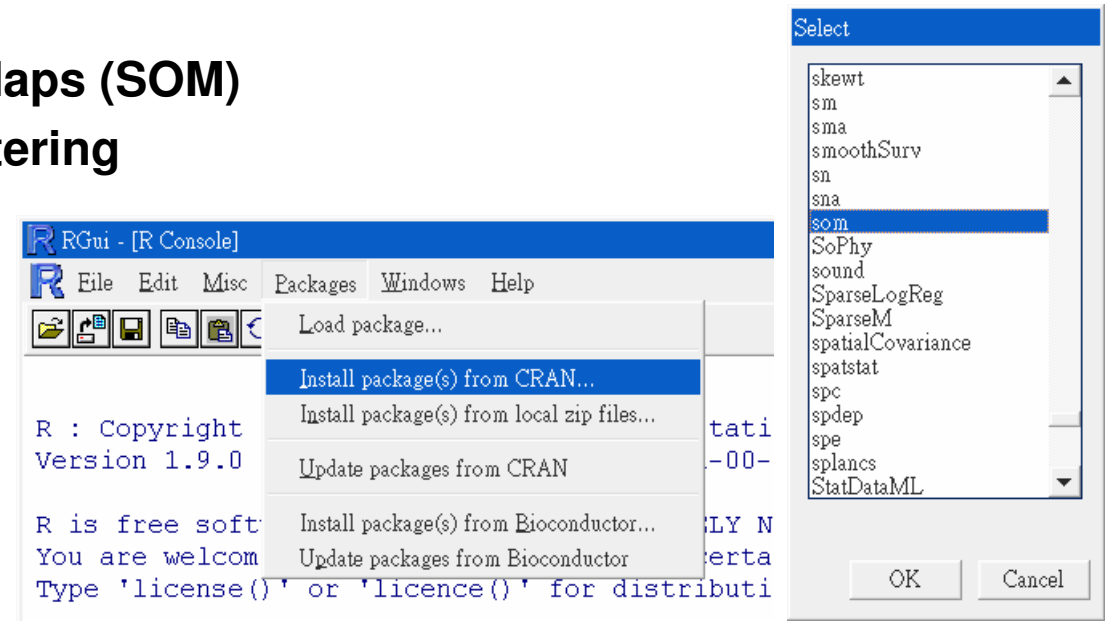
吳漢銘
hmwu@stat.sinica.edu.tw
http://www.sinica.edu.tw/~hmwu

2005年7月27日

中央研究院 統計科學研究所
Institute of Statistical Science, Academia Sinica

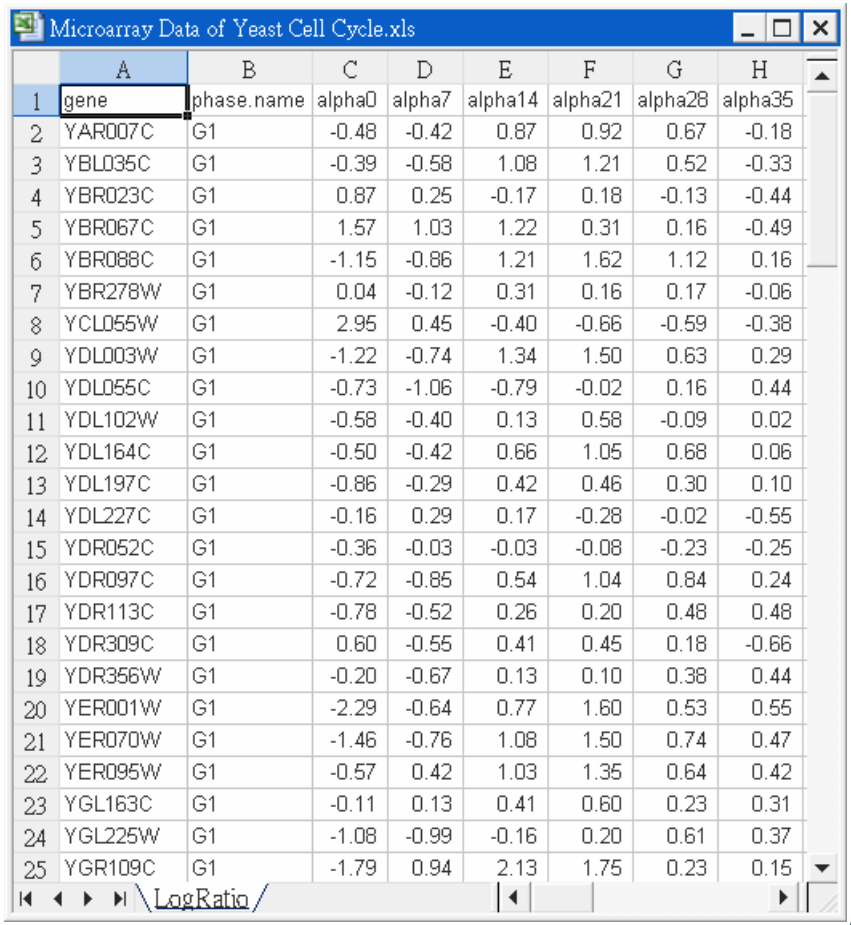
Outlines

- **Microarray Data of Yeast Cell Cycle**
- **Graphics:**
  - histogram, boxplot, scatterplot matrix, data image, line plots.
- **Principal Component Analysis (PCA)**
- **Multidimensional Scaling (MDS)**
- **K-Means**
- **Self-Organizing Maps (SOM)**
- **Hierarchical Clustering**



Microarray Data of Yeast Cell Cycle

- Spellman et al., (1998). Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast Saccharomyces cerevisiae by Microarray Hybridization. Molecular Biology of the Cell 9, 3273-3297.
- The data consists of several sub-sets collected under different conditions: alpha factor arrest, elutriation, arrest of cdc15 and cdc28 temperature-sensitive mutant.
- Each of these sub-sets is a single experiment.
- These experiment methods are used to synchronize the yeast cell cycle.
- Synchronized by alpha factor arrest method: Spellman et al. (1999).
- Time course data: every 7 minutes and totally 18 time points.
- Known genes: there are 103 cell cycle-regulated genes by traditional method in G1, S, S/G2, G2/M, or M/G1.

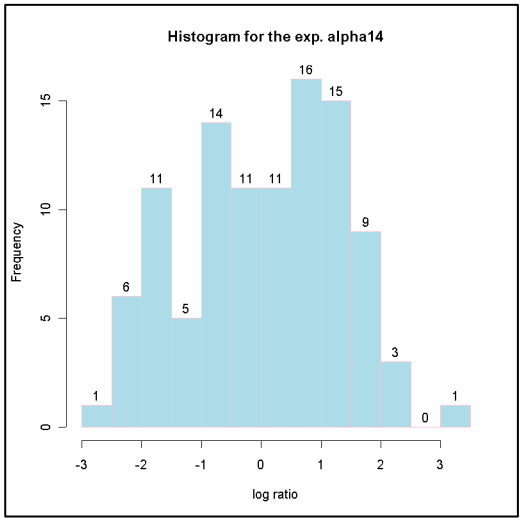


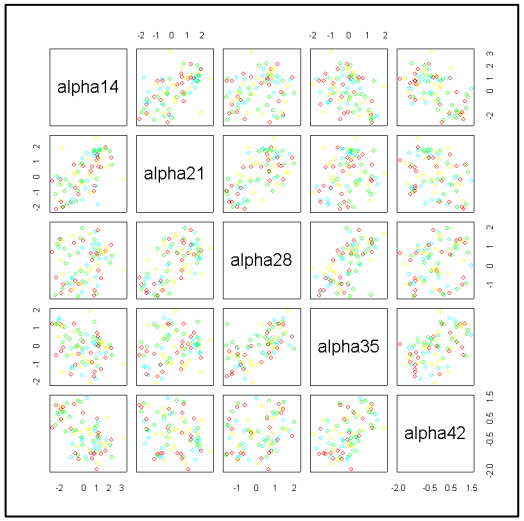
Microarray Data of Yeast Cell Cycle.xls

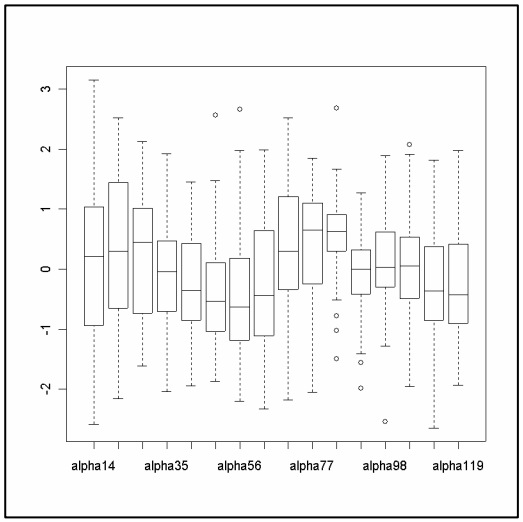
	A	B	C	D	E	F	G	H
1	gene	phase.name	alpha0	alpha7	alpha14	alpha21	alpha28	alpha35
2	YAR007C	G1	-0.48	-0.42	0.87	0.92	0.67	-0.18
3	YBL035C	G1	-0.39	-0.58	1.08	1.21	0.52	-0.33
4	YBR023C	G1	0.87	0.25	-0.17	0.18	-0.13	-0.44
5	YBR067C	G1	1.57	1.03	1.22	0.31	0.16	-0.49
6	YBR088C	G1	-1.15	-0.86	1.21	1.62	1.12	0.16
7	YBR278W	G1	0.04	-0.12	0.31	0.16	0.17	-0.06
8	YCL055W	G1	2.95	0.45	-0.40	-0.66	-0.59	-0.38
9	YDL003W	G1	-1.22	-0.74	1.34	1.50	0.63	0.29
10	YDL055C	G1	-0.73	-1.06	-0.79	-0.02	0.16	0.44
11	YDL102W	G1	-0.58	-0.40	0.13	0.58	-0.09	0.02
12	YDL164C	G1	-0.50	-0.42	0.66	1.05	0.68	0.06
13	YDL197C	G1	-0.86	-0.29	0.42	0.46	0.30	0.10
14	YDL227C	G1	-0.16	0.29	0.17	-0.28	-0.02	-0.55
15	YDR052C	G1	-0.36	-0.03	-0.03	-0.08	-0.23	-0.25
16	YDR097C	G1	-0.72	-0.85	0.54	1.04	0.84	0.24
17	YDR113C	G1	-0.78	-0.52	0.26	0.20	0.48	0.48
18	YDR309C	G1	0.60	-0.55	0.41	0.45	0.18	-0.66
19	YDR356W	G1	-0.20	-0.67	0.13	0.10	0.38	0.44
20	YER001W	G1	-2.29	-0.64	0.77	1.60	0.53	0.55
21	YER070W	G1	-1.46	-0.76	1.08	1.50	0.74	0.47
22	YER095W	G1	-0.57	0.42	1.03	1.35	0.64	0.42
23	YGL163C	G1	-0.11	0.13	0.41	0.60	0.23	0.31
24	YGL225W	G1	-1.08	-0.99	-0.16	0.20	0.61	0.37
25	YGR109C	G1	-1.79	0.94	2.13	1.75	0.23	0.15

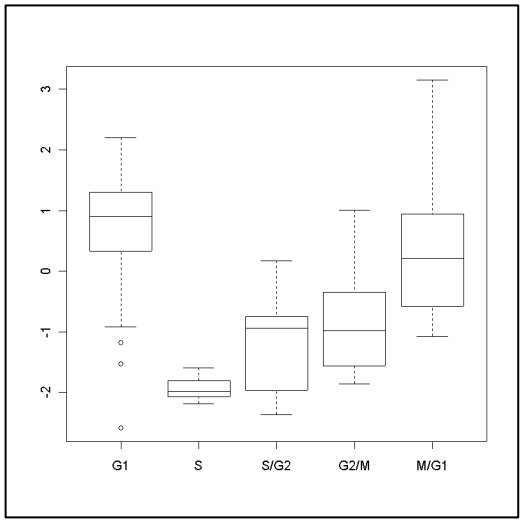
LogRatio

Histogram, Boxplot and Scatterplot matrix









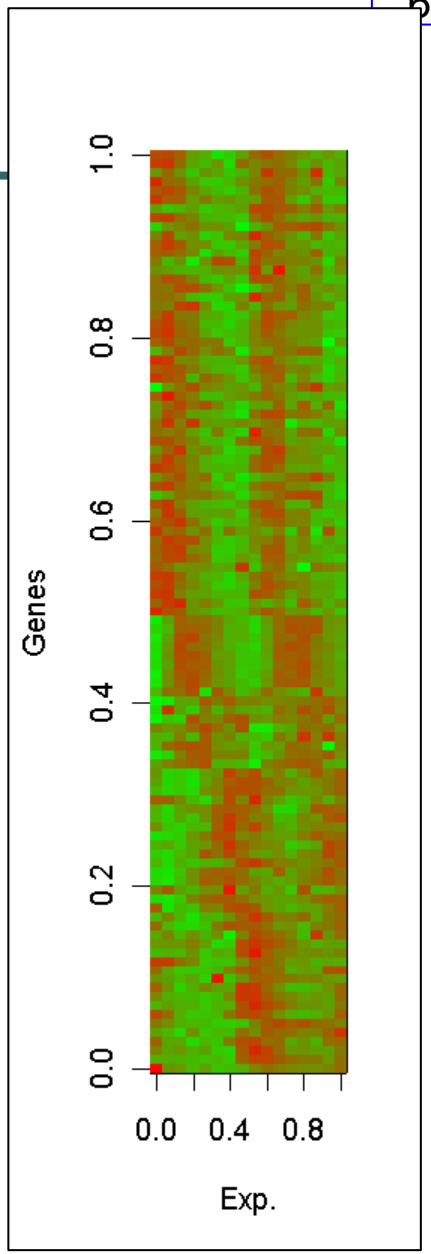
Plots

```
## Read Data
setwd("C:\\Program Files\\R\\rw2001\\WorkingData")
library(stats)
cell.matrix  <- read.table("TradCellCycle103_alpha.txt", header=TRUE)
n <- dim(cell.matrix)[1]
p <- dim(cell.matrix)[2]-2
cell.data <- cell.matrix[,3:p+2]
gene.name <- cell.matrix[,1]
gene.phase <- cell.matrix[,2]
phase <- unique(gene.phase)
phase.name <- c("G1", "S", "S/G2", "G2/M", "M/G1")

## standardized data
cell.sdata <- (cell.data-apply(cell.data, 1, mean))/sqrt(apply(cell.data, 1, var))

#Histogram
hist(cell.sdata[,1], br=12, col="lightblue", border="pink", labels = TRUE, main="Histogram for the
exp. alpha14", xlab="log ratio")

#Boxplot
boxplot(cell.sdata)
boxplot(cell.sdata[,1]~gene.phase, names=phase.name)

#Scatterplot matrix
pairs(cell.sdata[,1:5], col=phase)
```

Data image

Data Image
cell.image <- as.matrix(t(cell.sdata[n:1,]))
RGcol <- maPalette(low = "green", high = "red", k = 50)
image(cell.image, xlab="Exp.", ylab="Genes", col = RGcol)

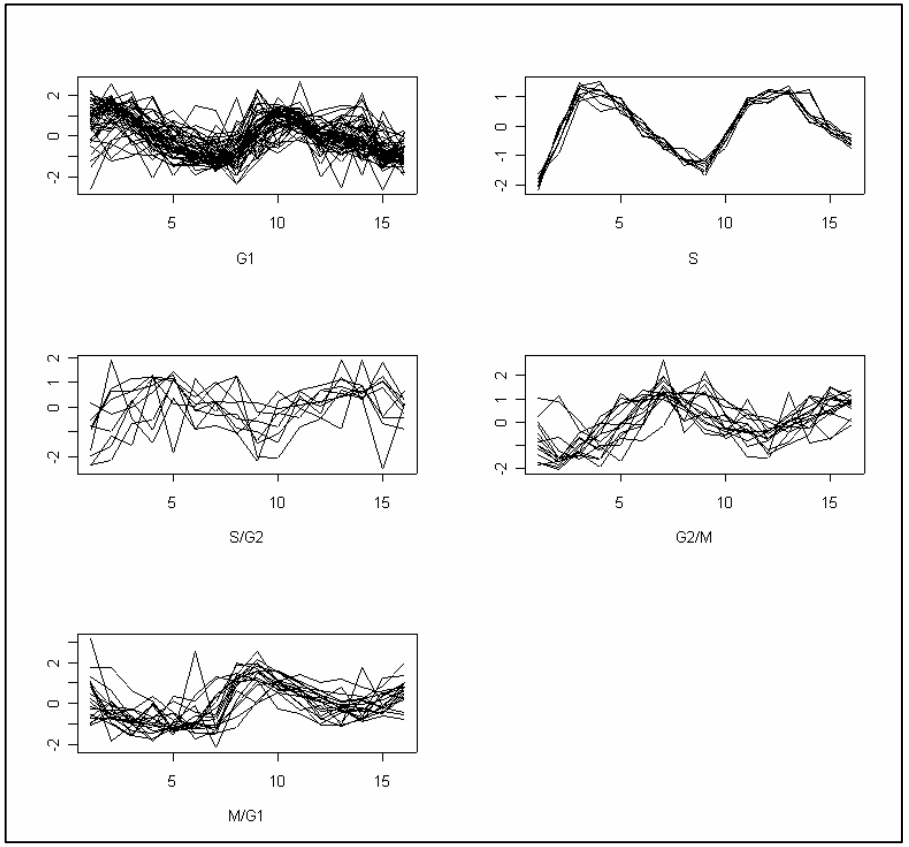
```
maPalette <- function(low = "white",
                      high = c("green", "red"),
                      mid=NULL,
                      k =50)
{
     low <- col2rgb(low)/255
     high <- col2rgb(high)/255

     if(is.null(mid)){
          r <- seq(low[1], high[1], len = k)
          g <- seq(low[2], high[2], len = k)
          b <- seq(low[3], high[3], len = k)
       }
     if(!is.null(mid)){
       k2 <- round(k/2)
       mid <- col2rgb(mid)/255
       r <- c(seq(low[1], mid[1], len = k2),
              seq(mid[1], high[1], len = k2))
       g <- c(seq(low[2], mid[2], len = k2),
              seq(mid[2], high[2], len = k2))
       b <- c(seq(low[3], mid[3], len = k2),
              seq(mid[3], high[3], len = k2))
     }
     rgb(r, g, b)
}
```



Note: should source “maPalette function” first (Dudoit and Yang).

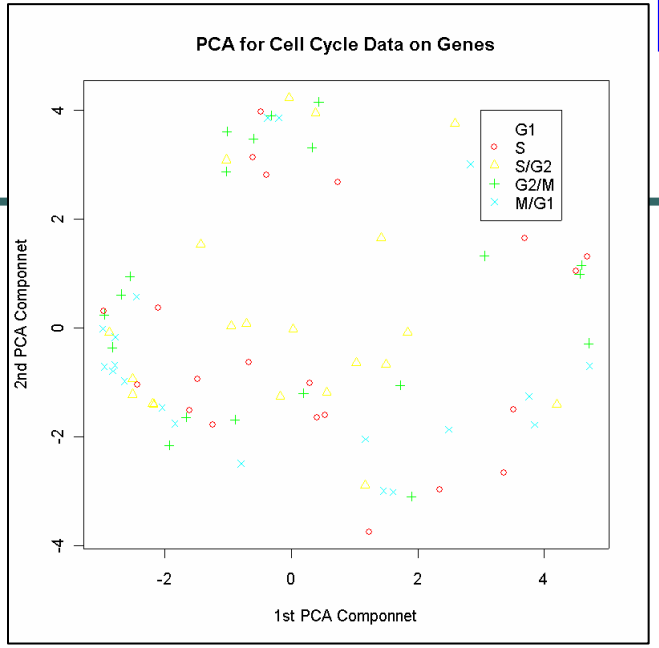
Time Series Plots

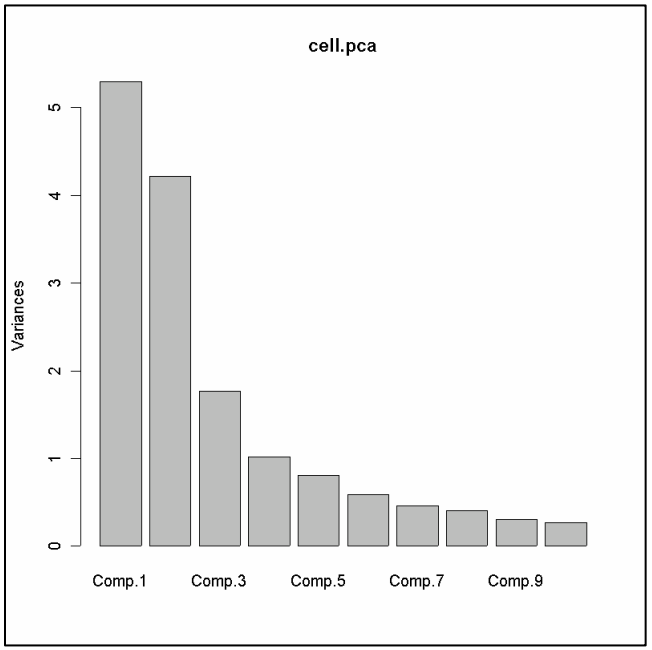
```
## Time Series Plots
number <- 1:n

## standardized data
cell.sdata <- (cell.data-apply(cell.data, 1, mean))
/sqrt(apply(cell.data, 1, var))

par(mfrow=c(3,2))
for(i in 0:4){
   ts.plot(t(cell.sdata[number[gene.phase==i],]),
           xlab=phase.name[i+1])
}
```



PCA

```
cell.pca <- princomp(cell.sdata, cor=TRUE,
scores=TRUE)

# 2D plot for first two components
pca.dim1 <- cell.pca$scores[,1]
pca.dim2 <- cell.pca$scores[,2]
plot(pca.dim1, pca.dim2,
main="PCA for Cell Cycle Data on Genes", xlab="1st
PCA Componnet", ylab="2nd PCA Componnet",
col=c(phase), pch=c(phase))
legend(3, 4, phase.name, pch=c(phase), col=c(phase))

# shows a screeplot.
plot(cell.pca)
biplot(cell.pca)
```



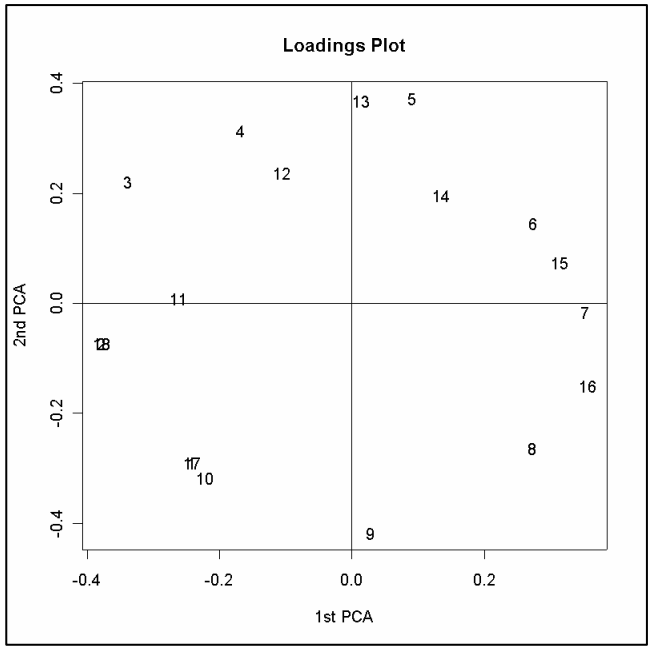


PCA

```
## loadings plot
plot(loadings(cell.pca)[,1], loadings(cell.pca)[,2], xlab="1st PCA", ylab="2nd PCA", main="Loadings Plot", type="n")
text(loadings(cell.pca)[,1], loadings(cell.pca)[,2], labels=paste(1:p))
abline(h=0)
abline(v=0)
```

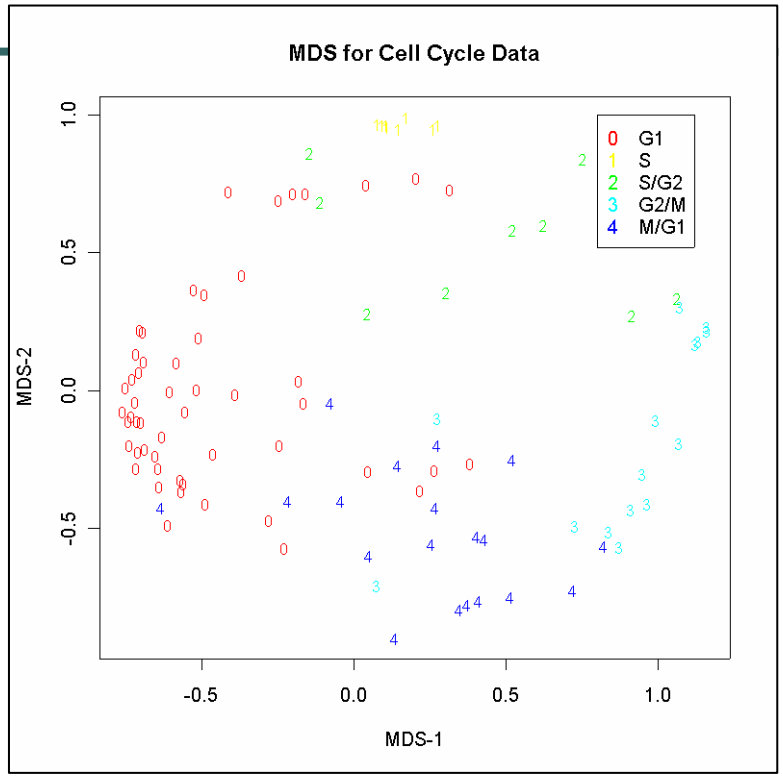
```
> summary(cell.pca)
Importance of components:
                          Comp.1    Comp.2    Comp.3     Comp.4     Comp.5         Comp.15
Standard deviation     2.3012110 2.0542795 1.3300507 1.00895544 0.90053289    0.308577283
Proportion of Variance 0.3309732 0.2637540 0.1105647 0.06362444 0.05068497 ... 0.005951246
Cumulative Proportion  0.3309732 0.5947272 0.7052919 0.76891637 0.81960134    1.000000000
```

```
# print loadings
loadings(cell.pca)
summary(cell.pca)
```



Loadings:	Comp.1	Comp.2	Comp.3	Comp.4
alpha14	-0.283	-0.21	0.283	0.136
alpha21	-0.374	0.211	-0.135	-0.16
alpha28	-0.26	0.298	0.161	-0.168
alpha35	-0.102	0.372	0.165	-0.321
alpha42	0.161	0.355	0.2	-0.317
alpha49	0.287	0.167	0.116	-0.515
alpha56	0.35	0.172	-0.274	-0.115
alpha63	0.251	-0.258	-0.275	-0.37
alpha70	-0.372	-0.217	-0.382	-0.159
alpha77	-0.253	-0.221	-0.321	-0.32
alpha84	-0.249	-0.437	-0.309	-0.256
alpha91	-0.115	0.279	-0.436	0.114
alpha98	0.36	-0.284	0.186	-0.138
alpha105	0.16	0.257	-0.283	-0.125
alpha112	0.347	0.319	-0.178	-0.276
alpha119	0.348	-0.164	-0.201	0.11

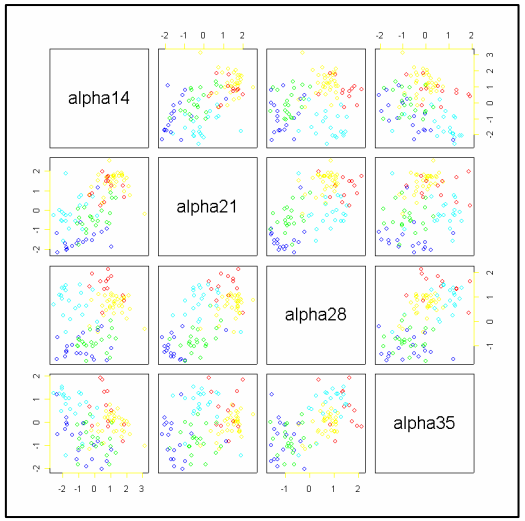
MDS

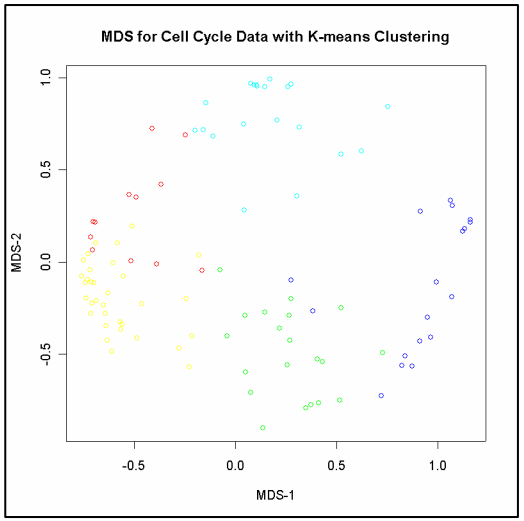


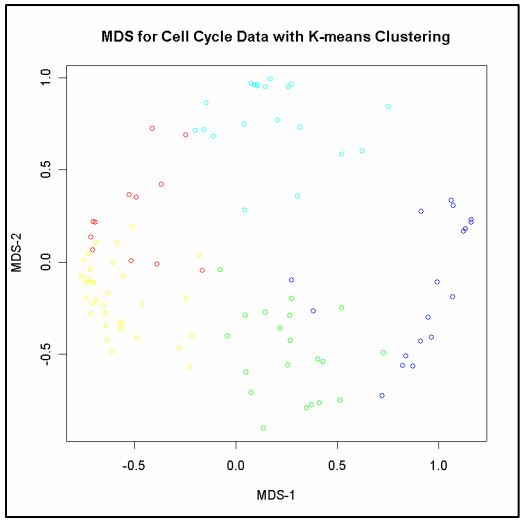

```
#correlation matrix
cell.cor<- cor(t(cell.sdata))

#distance matrix
cell.dist<- sqrt(2*(1-cell.cor))

cell.mds<- cmdscale(cell.dist)
mds.dim1 <- cell.mds[,1]
mds.dim2 <- cell.mds[,2]

plot(mds.dim1, mds.dim2, type="n", xlab="MDS-1", ylab="MDS-2", main="MDS for Cell Cycle Data")
for(i in 0:4){
   text(mds.dim1[number[gene.phase==i]], mds.dim2[number[gene.phase==i]],
gene.phase[number[gene.phase==i]] , cex=0.8, col= i+1)
}
legend(0.8, 1.0, phase.name, pch="01234", col=c(1,2,3,4,5))
```

K-Means

```
no.group <- 5
no.iter <- 20
cell.kmeans <- kmeans(cell.sdata, no.group, no.iter)
plot(cell.sdata[,1:4], col = cell.kmeans$cluster)

## PCA
plot(pca.dim1, pca.dim2,  main="PCA for Cell Cycle Data with K-means Clustering", xlab="PCA-1",
ylab="PCA-2", col=cell.kmeans$cluster)

## MDS
plot(mds.dim1, mds.dim2, xlab="MDS-1", ylab="MDS-2", main="MDS for Cell Cycle Data with K-means
Clustering", col = cell.kmeans$cluster)
```

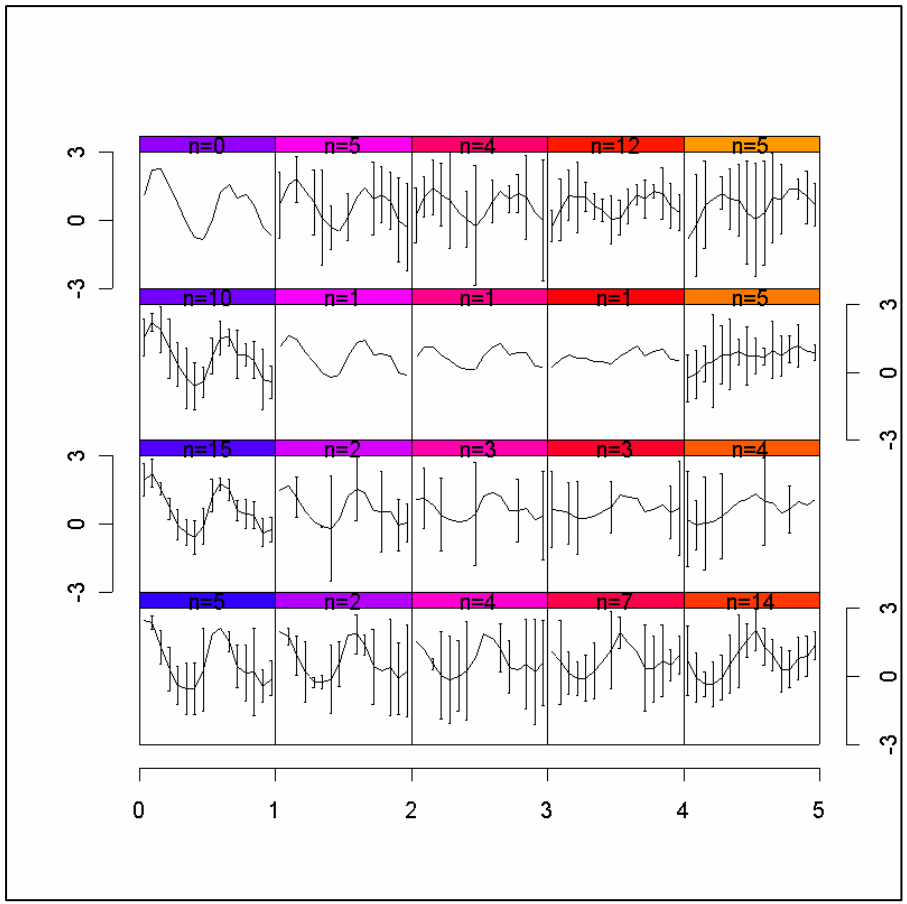






SOM

```
library(som)
cell.som <- som(cell.sdata, xdim=5, ydim=4, topol="rect", neigh="gaussian")
plot(cell.som)
```



Hierarchical Clustering

Hierarchical Clustering on genes
cell.gene.hc.ave <- hclust(dist(cell.sdata), method = "ave")
plot(cell.gene.hc.ave, hang = -1, cex=0.5, labels=gene.name)

Hierarchical Clustering on experiments
cell.exp.hc.ave <- hclust(dist(t(cell.sdata)), method = "ave")
plot(cell.exp.hc.ave, cex=0.8)