Clustering, dimension reduction and the yeast cell cycle example.

Clustering and Dimension Reduction

Dimension reduction techniques (e.g. PCA/SVD) do not produce clusters. However:

1. Dimension reduction can be used to form groups of genes, e.g.:
   - the closest to the first, second, third etc. direction;
   - the closest or furthest from the first direction, plane, 3D space, etc.

2. Dimension reduction can be used to visualize the outcome of clustering, e.g.:
   (if T>2) plot points color-coded according to cluster membership on the 1st principal components plane. This 2D projection view is “most representative” of the data (in that it maximizes the share of captured overall variation), but it is not necessarily the best view for “separating” clusters.

3. Dimension reduction can be used prior to clustering, e.g.:
   (if T>2) cluster points after projection on the 1st principal components plane. This may eliminate noise and artifacts from the clustering exercise, and facilitate cluster computation (some algorithms depend strongly on dimension), but care is needed on how much and what we are willing to “throw away”.

Alternatives to PCA in relation to clustering

Linear classifiers (Discriminant Analysis, Sliced Inverse Regression):

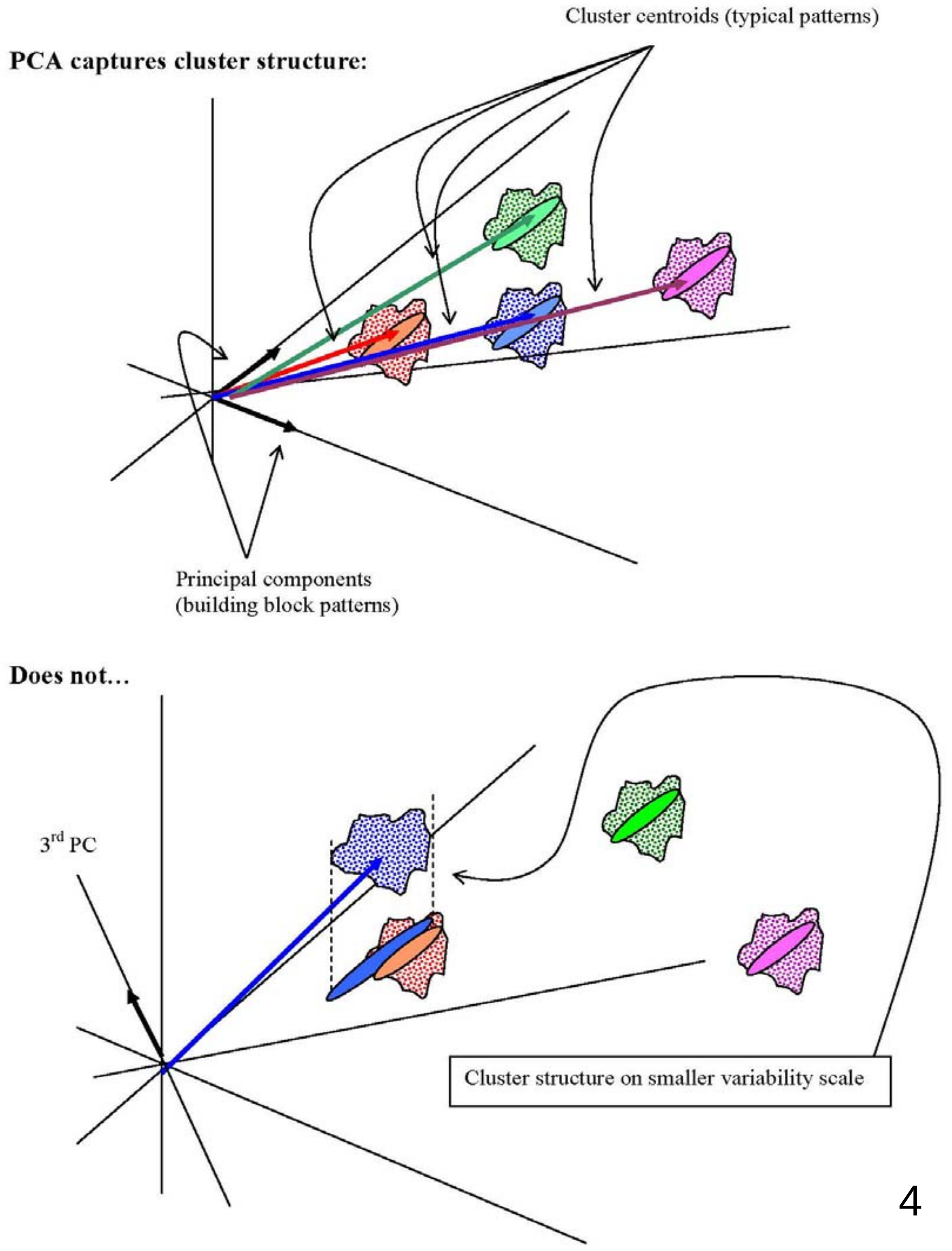
After clustering, treat cluster memberships as a (known) classification response. Find directions, and thus low-dimensional projections, that preserve separation ("distinguishability") among the now given classes. The resulting 2D projection views are (linearly) optimized for cluster separation.

Multidimensional scaling:

Find directions, and thus low-dimensional projections, that preserve distances among data points. After clustering, 2D projection views obtained with multidimensional scaling may provide a better cluster visualization (in terms of separation) than 2D views obtained from PCA. Before clustering, reducing the data with multidimensional scaling aims at preserving the "basis" for clustering (distances), and thus may be more effective than PCA.

Yeung K.Y., Ruzzo W.L. (2001): Principal component analysis for clustering gene expression data. *Bioinformatics* 17 (9) 762-744.

Using several clustering methods and distance choices, and both actual and simulated data, they show how clustering based on the first few principal components may significantly degrade the clustering results.


Cluster centroids (typical patterns)
PCA captures cluster structure:
Principal components (building block patterns)
Does not...
3rd PC
Cluster structure on smaller variability scale

Example, and clustering in R

Yeast cell-cycle data

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.

Recall

- 678 genes (exhibiting periodic behavior, no missing values)
- first 12 time points from cdc-synchronized time course
- normalized (to green; not synchronized) log-ratios, from spotted arrays
- row-standardized values (relative to first 12 columns)

```
> library(stats)
> yeast <- read.table("proc_yeast_cycle.txt",header=TRUE)
> dim(yeast)            #contains only numerical data
[1] 678  12
```

```
> help(hclust)                  #selected text from:
```

hclust(d, method = "complete")

Arguments:

d: a dissimilarity structure as produced by 'dist'.

method: the agglomeration method to be used: “single”, “complete”, “average”, etc.

Value: An object of class hclust which describes the tree produced by the clustering process. It has many components, among them

height: a set of n-1 non-decreasing real values expressing the clustering heights: that is, the value of the criterion associated with the clustering 'method' at each agglomeration.

```
> help(dist)                    #selected text from:
```

dist(x, method = "euclidean", diag = FALSE, upper = FALSE)

Arguments:

x: a numeric matrix or data frame

method: the distance measure to be used: “euclidean”, “maximum”, “manhattan”, etc.

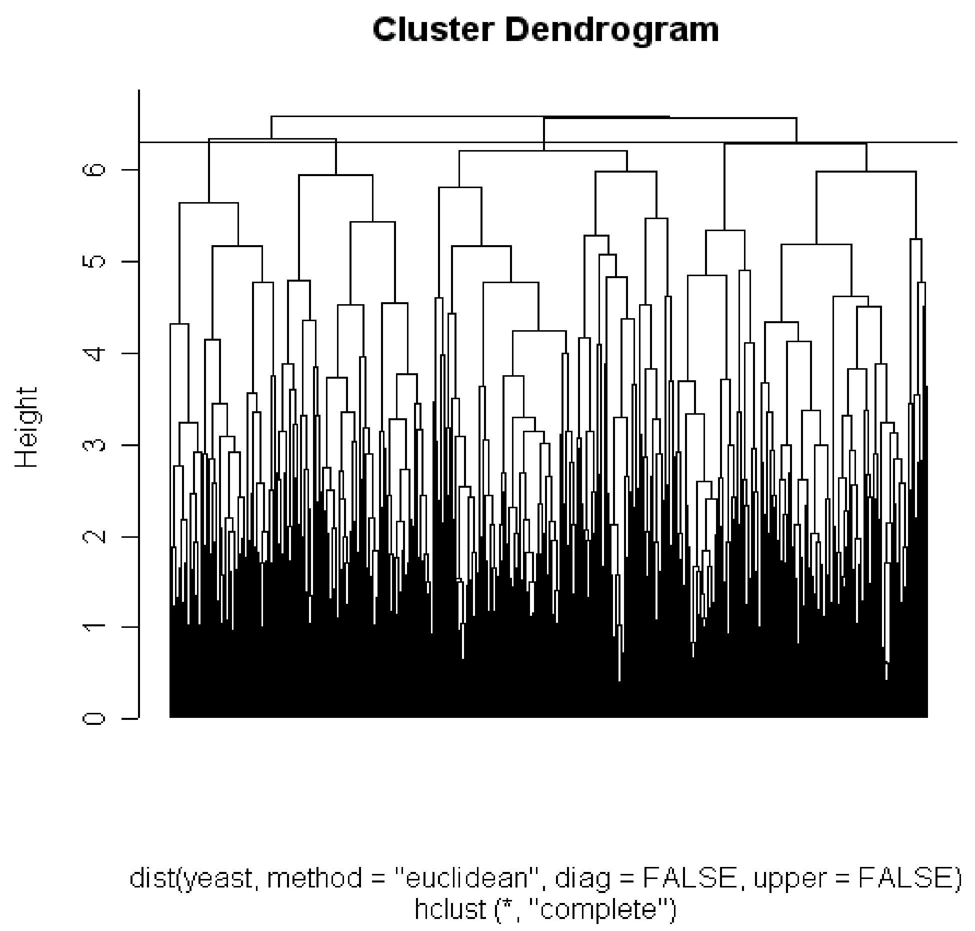
diag: logical, whether the diagonal of the distance matrix is given by 'print.dist'.

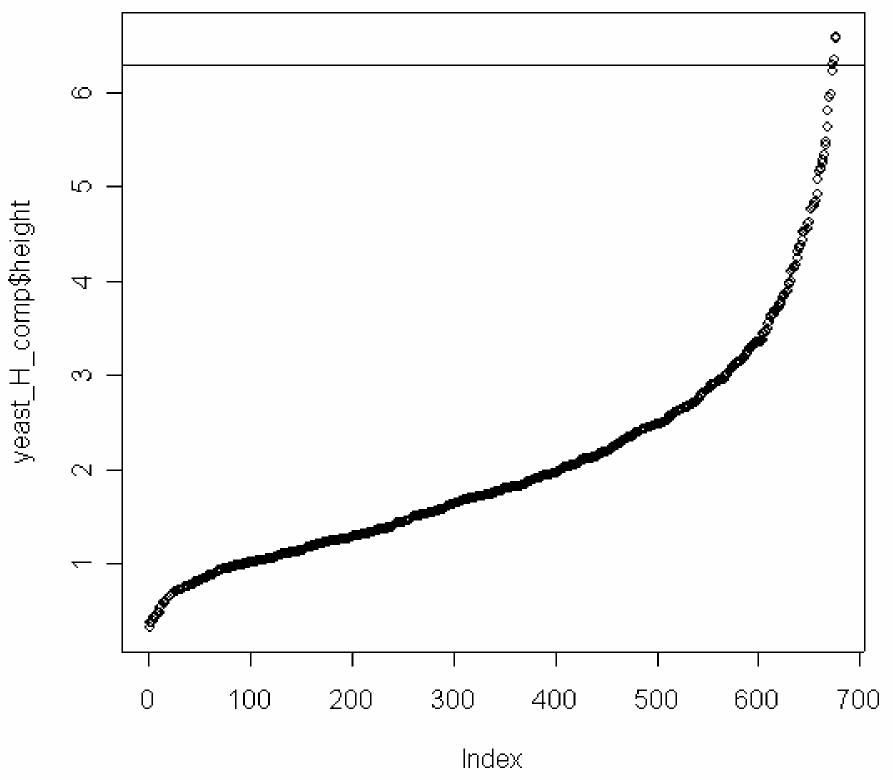
upper: logical, whether the upper triangle of the distance matrix is given by 'print.dist'.

Value: An object of class dist, with various components.

```
> yeast_H_comp <- hclust(
+ dist(yeast, method = "euclidean", diag = FALSE, upper = FALSE),
+ method = "complete")
> plot(yeast_H_comp,labels=FALSE,hang=-1)  #dendrogram
> abline(6.3,0)           #threshold for 5 clusters (cell cycle phases)
> plot(yeast_H_comp$height)  #heights=distances at mergers
> abline(6.3,0)
```

NOT VERY CONVINCING EVIDENCE FOR 5 CLUSTERS; EVIDENCE OF CLUSTERING?





```
> help(cutree)                    #selected text from:
```

cutree(tree, k = 5, h = NULL)

Arguments:

tree: as produced by hclust.

k: an integer, or vector of integers, with the desired number of groups

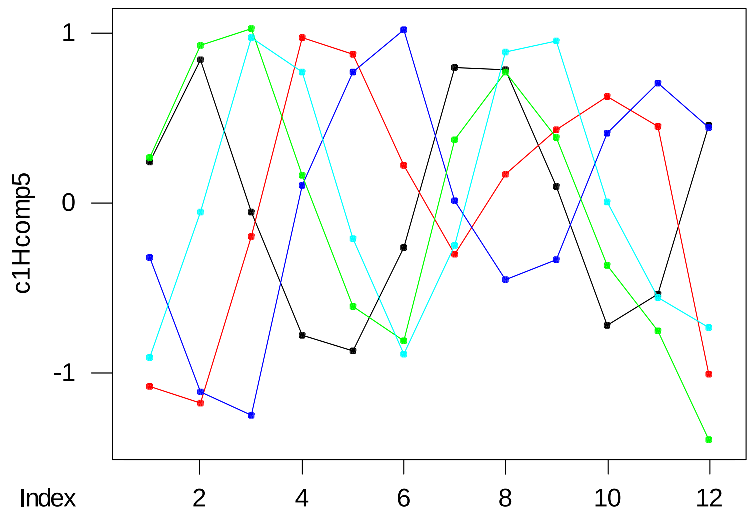
h: a scalar, or vector of scalars, with height(s) to cut.

(at least one must be specified, 'k' overrides 'h' if both are given.)

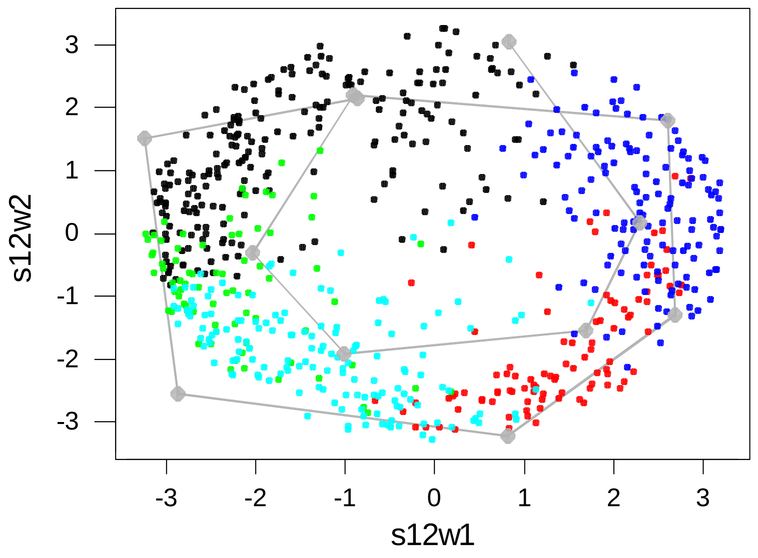
Value: a vector (matrix if several k's or h's) with memberships (k's or h's used as clmn names).

```
> yeast_H_comp_M5 <- cutree(yeast_H_comp,k=5) #get memberships
```

Cluster centroids, Hcomp5
black=1 red=2 blue=3 green=4 cyan=5



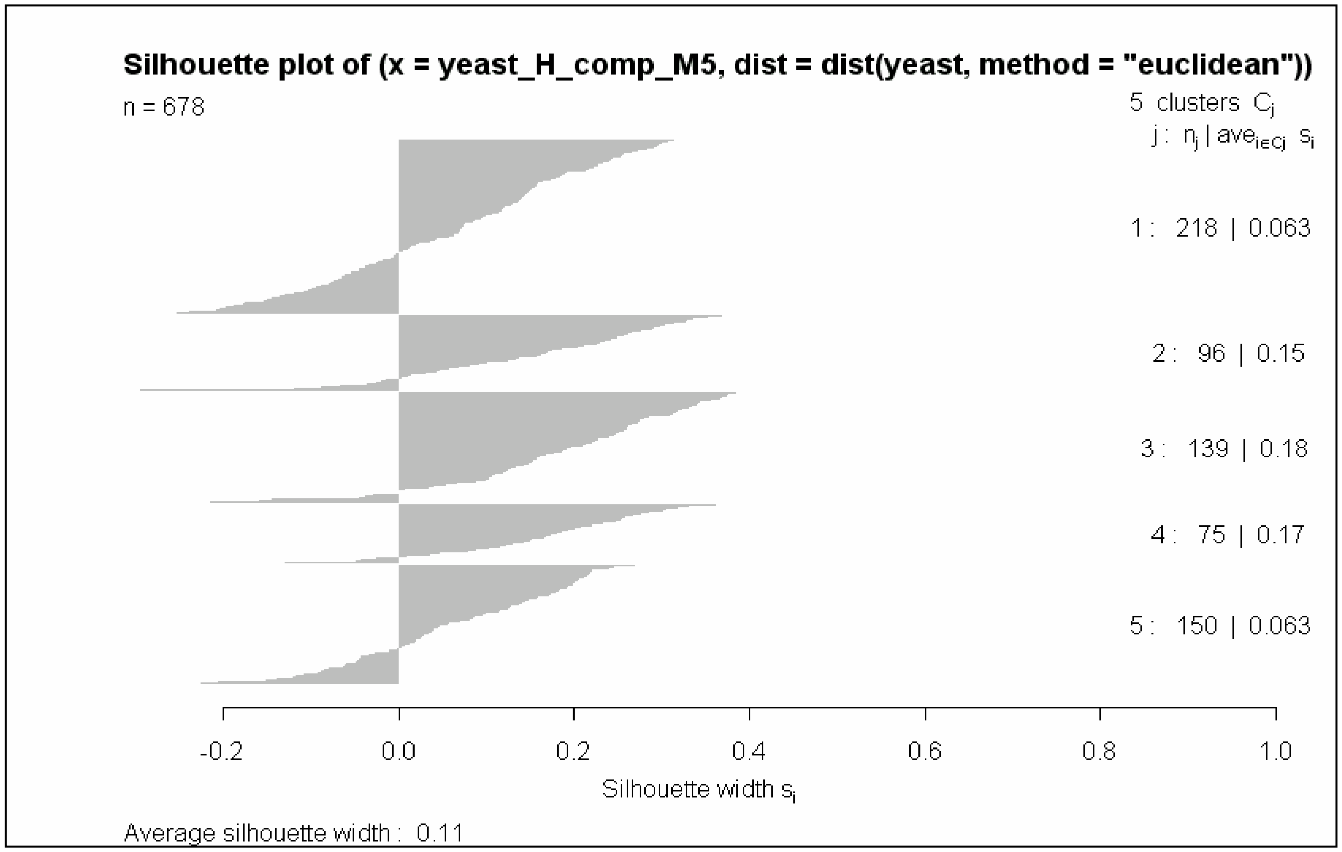
Clusters from Hcomp5 (1st PCA plane)
black=1 red=2 blue=3 green=4 cyan=5



(1)=218 (2)=96 (3)=139 (4)=75 (5)=150 genes

(gray: original coord's projected on the plane, joined in time order and magnified)

```
> #silhouette (see help) plot using memberships,and distance matrix
> #previously used in clustering.
> plot(silhouette(yeast_H_comp_M5,dist(yeast, method = "euclidean")))
```



Cluster centroids and locations seem to make sense, but do we really have evidence of clustering (do the data present “lumps”), or justification for 5 clusters?

```
> help(kmeans)                           #selected text from:
```

kmeans(x, centers, iter.max = 100)

Arguments:

x: a numeric matrix of data, or data frame with all numeric columns.

centers: number of clusters, or matrix with initial cluster centers (each row a center). If the first, a random set of rows in 'x' are chosen as the initial centers.

iter.max: maximum number of iterations allowed.

Value: list including

cluster: a vector of integers containing memberships.

centers: a matrix with final cluster centroids.

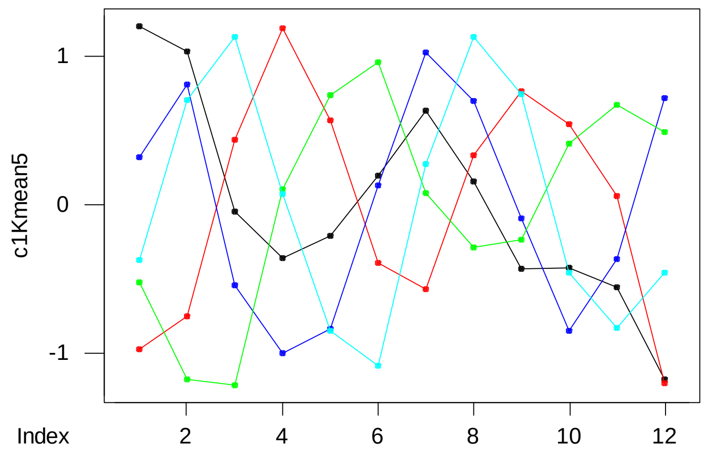
withinss: within-cluster sum of squares for each cluster.

size: the number of points in each cluster.

```
> #very sensitive to initialization!
> yeast_K_5 <- kmeans(yeast,5,1000)          #random initialization
> yeast_K_5 <- kmeans(yeast,K5init,1000)     #giving initial centroids
```

Cluster centroids, Kmeans5

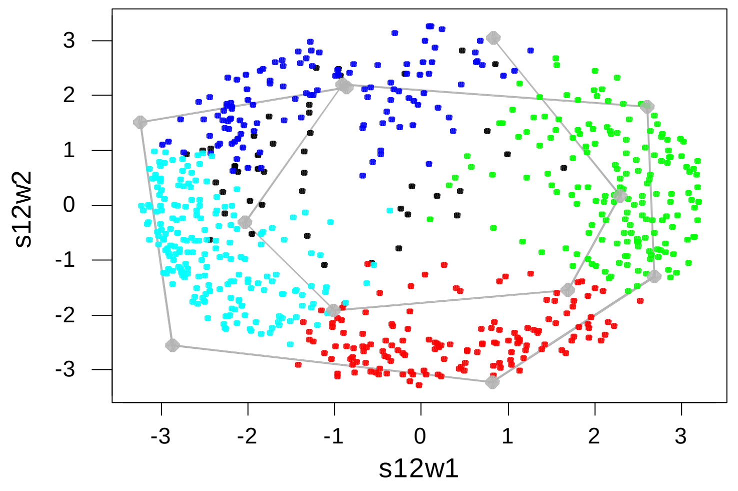
black=1 red=2 blue=3 green=4 cyan=5



(1)=45 (2)=143 (3)=119 (4)=169 (5)=202 genes

Clusters from Kmean5 (1st PCA plane)

black=1 red=2 blue=3 green=4 cyan=5



(gray: original coord's projected on the plane, joined in time order and magnified)

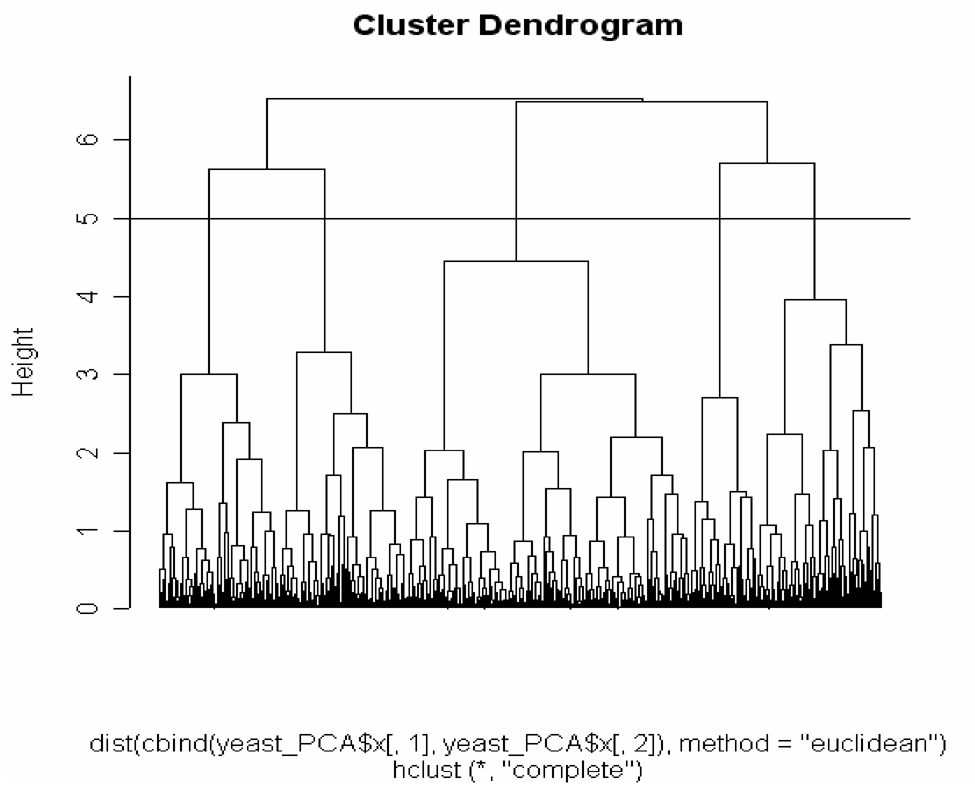
Aside: How to get PCA 2D projection views of the clusters

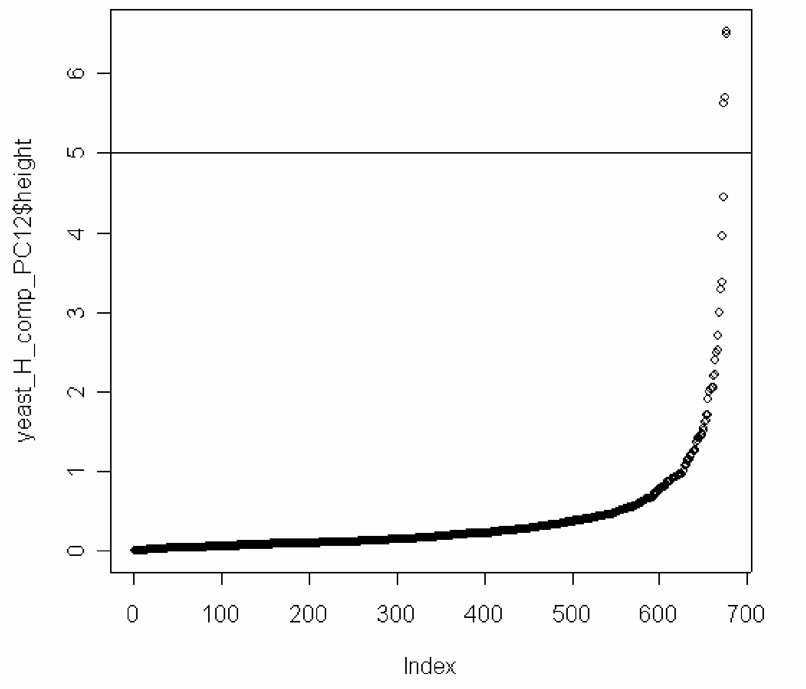
```
> yeast_PCA <- prcomp(yeast,retx=TRUE,center=TRUE,scale.=FALSE,tol=NULL)
> plot(yeast_PCA$x[,1],yeast_PCA$x[,2],col = yeast_H_comp_M5)
> plot(yeast_PCA$x[,1],yeast_PCA$x[,2],col = yeast_K_5$cluster)
```

Note linear trend attributable to one cluster (**hier**, **kmeans**; centroids plot; points growing inwards from the rim in the PCA 2D projection view – 3rd PC).

… Could perform clustering in the PC plane:

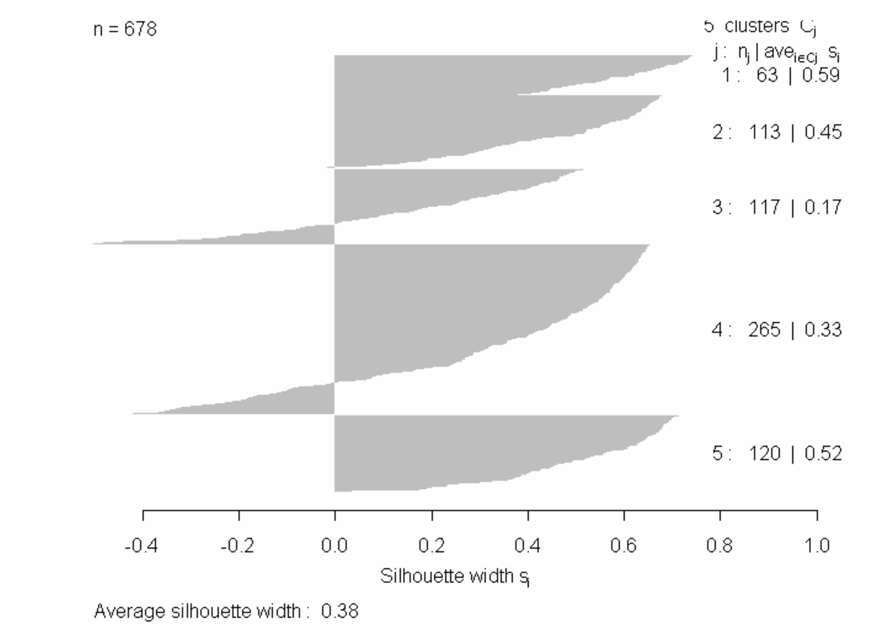
```
> yeast_H_comp_PC12 <- hclust(dist(cbind(yeast_PCA$x[,1],
+ yeast_PCA$x[,2]),method = "euclidean"),method = "complete")
```

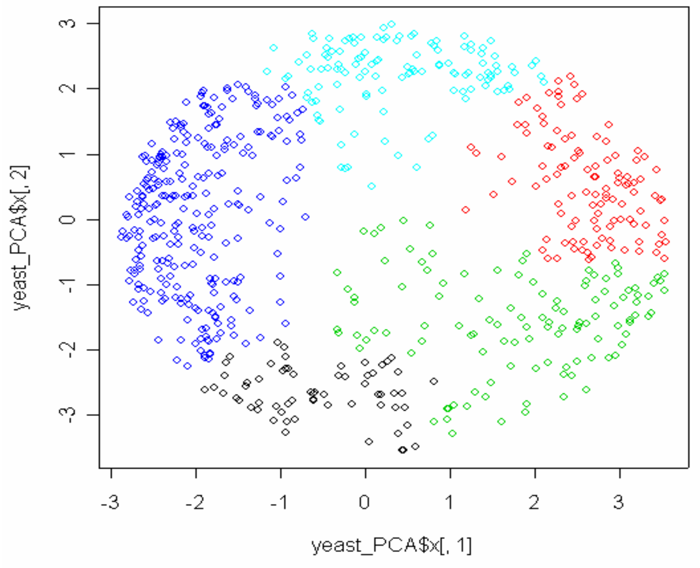




Reducing artifact effects (trend!) seems to help; 5th cluster is “free” to locate on the rim, better evidence for clustering and justification for 5. But careful! Yeung & Ruzzo (2001)

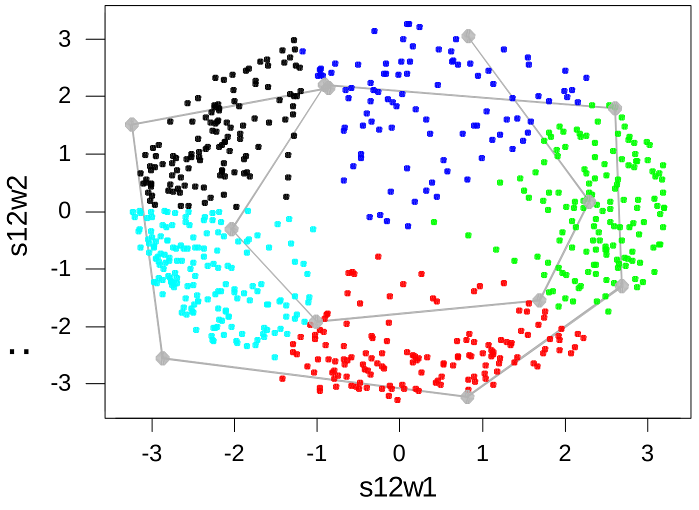
```
> yeast_H_comp_M5_PC12 <- cutree(yeast_H_comp_PC12,k=5)
> plot(yeast_PCA$x[,1],yeast_PCA$x[,2],col = yeast_H_comp_M5_PC12)
> plot(silhouette(yeast_H_comp_M5_PC12,dist(cbind(yeast_PCA$x[,1],
+ yeast_PCA$x[,2]),method="euclidean")))
```





Similarly with K-means

```
> yeast_K_5_PC12 <-kmeans(
+ cbind(yeast_PCA$x[,1],yeast_PCA$x[,2]),
+ K5initPC12,1000)
```



Yet note how “arbitrary” the clusters still seem to be…

F. Chiaromonte Sp 06