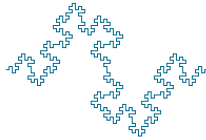


High-Throughput Sequencing Course

Unsupervised Learning

Biostatistics and Bioinformatics

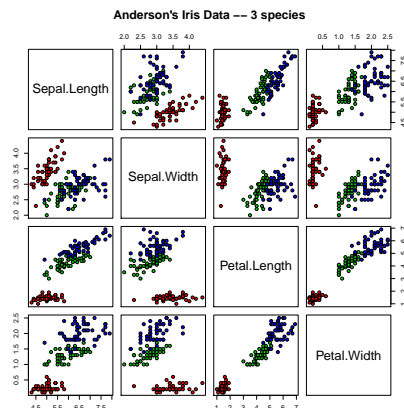


Summer 2019

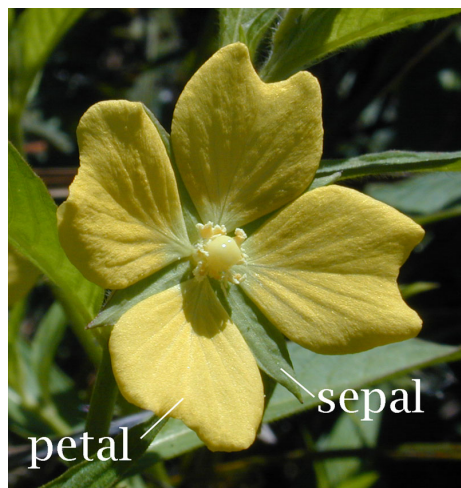
SCOPE

- ▶ Let X denote the genetic/genomic profile of a sample
- ▶ Often we would like to discover groups, clusters or outliers based on the genetic profiles of the samples
- ▶ These are *unsupervised* methods in the sense that the algorithm knows nothing about the grouping/clustering
- ▶ The method is only aware of the genetic profile (X) and not the outcome Y

FISHER'S IRIS DATA

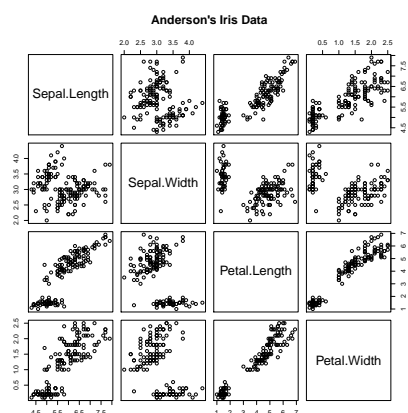


ON PETALS AND SEPALS

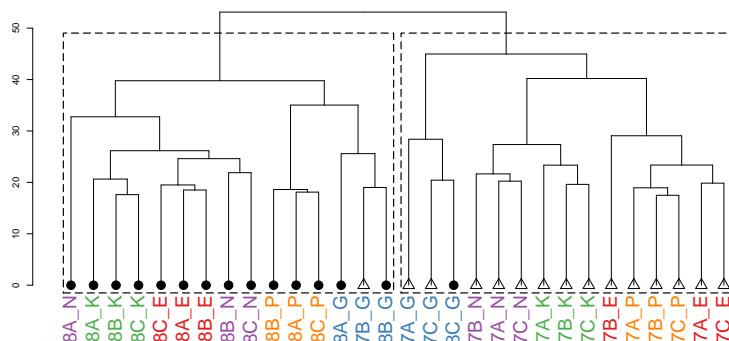


<https://en.wikipedia.org/wiki/Sepal>

FISHER'S IRIS DATA



2015 DATA: AGGLOMERATIVE HIERARCHICAL CLUSTERING



A SELF-FULFILLING PROPHECY

- ▶ Statistical methods for unsupervised learning guarantee one thing
- ▶ They will return a clustering of your data
- ▶ What they do not guarantee and are invariably unable to verify, is the biological relevance or reproducibility of the clustering
- ▶ In light of this Self-fulfilling Prophecy, these methods should be used with utmost care

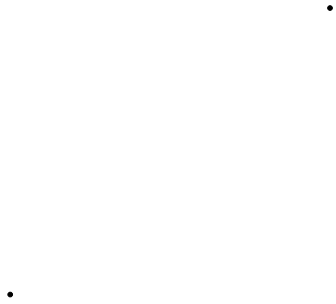
METHODS TO BE DISCUSSED

- ▶ There are many methods for unsupervised class discovery.
- ▶ We will consider three types of methods:
 - ▶ Hierarchical Clustering
 - ▶ k -means Clustering
 - ▶ Ordination Methods (e.g., Multi-Dimensional Scaling (MDS) and Principal Components (PC))
- ▶ Note that there are many variations of these methods
- ▶ Most mathematical details will be left out
- ▶ We focus on discovering classes among samples (not genes)

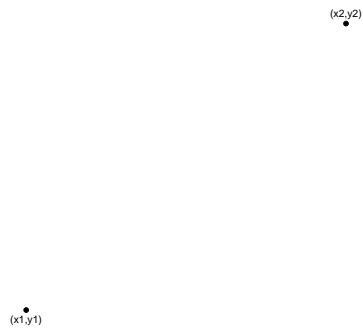
DISTANCE BETWEEN TWO POINTS

- ▶ Many class discover methods aim to quantify the similarity (or dissimilarity) among patients
- ▶ For each patient, the vector of gene expression can be thought of a "point" in an m -dimensional space
- ▶ For many class discovery methods, one has to be able to quantify the "distance" between two points (the expression profiles between two individuals)
- ▶ A common distance measure is the Euclidean distance

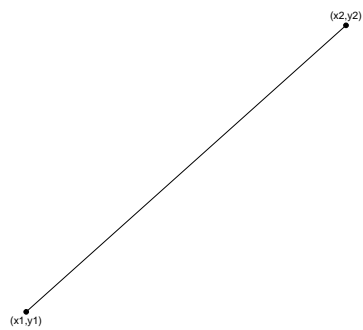
DISTANCE (TWO POINTS ON THE PLANE)



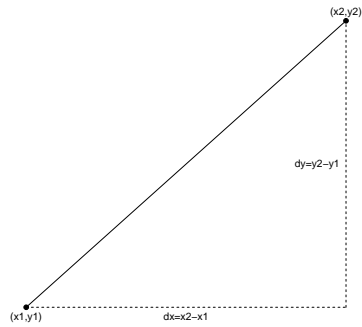
DISTANCE (COORDINATES)



DISTANCE



DISTANCE (HORIZONTAL/VERTICAL SHIFTS)



PYTHAGOREAN THEOREM (ON THE PLANE)

- According to the Pythagorean theorem

$$h^2 = dx^2 + dy^2 = (x_2 - x_1)^2 + (y_2 - y_1)^2$$

- h is called the hypotenuse
- The distance between (x_1, y_1) and (x_2, y_2) is given by

$$h = \sqrt{dx^2 + dy^2} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

PYTHAGOREAN THEOREM (ON THE PLANE)

- Can be extended to higher dimensions
- In a three-dimensional space the distance between (x_1, y_1, z_1) and (x_2, y_2, z_2) is given by

$$\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

- For any given dimension, the distance is obtained as the square root of the sum of the square of the coordinate-wise differences

GOLUB *et al* LEUKEMIA DATA

- ▶ 47 patients with acute lymphoblastic leukemia (ALL)
- ▶ 25 patients with acute myeloid leukemia (AML)
- ▶ Platform: Affymetrix Hgu6800
- ▶ 7129 probe sets
- ▶ Golub *et al.* (1999). Molecular classification of cancer: class discovery and class prediction by gene expression monitoring, Science, Vol. 286:531-537.

GOLUB *et al* LEUKEMIA DATA

Expression data from first three features and 5 patients

```
dim(exprs(Golub_Merge))  
  
## [1] 7129 72  
  
exprs(Golub_Merge)[1:3, 1:5]  
  
##           39  40  42  47  48  
## AFFX-BioB-5_at -342 -87  22 -243 -130  
## AFFX-BioB-M_at -200 -248 -153 -218 -177  
## AFFX-BioB-3_at  41  262  17 -163 -28
```

GOLUB *et al* LEUKEMIA DATA: DISTANCE

Expression vector for patients 39 and 40

```
x <- exprs(Golub_Merge)[, "39"]  
y <- exprs(Golub_Merge)[, "40"]
```

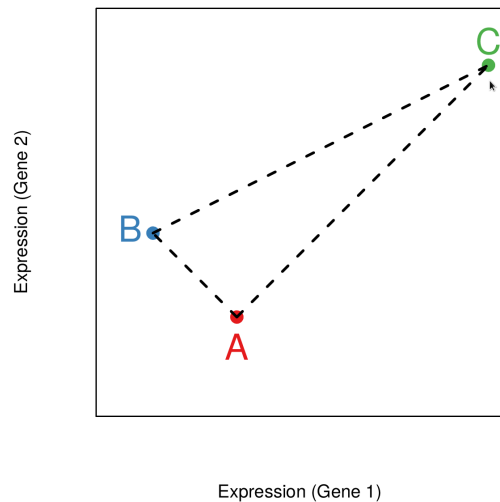
Lengths of these vectors

```
length(x)  
  
## [1] 7129  
  
length(y)  
  
## [1] 7129
```

Distance between these two vectors

```
sqrt(sum((x - y)^2))  
  
## [1] 101530.8
```

RELATIVE DISTANCE (FROM CST 2011 PAPER)



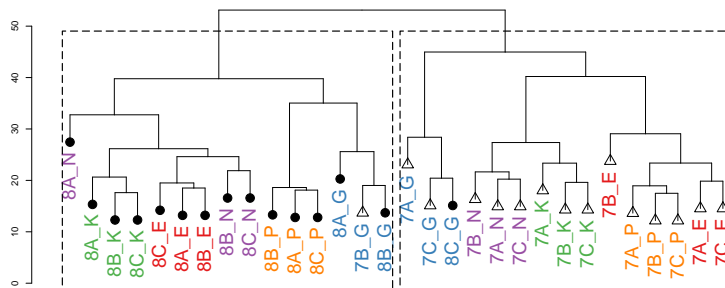
DISSIMILARITY MATRIX

- Use pairwise distances to quantify similarity (or dissimilarity) among patients
- Construct a matrix containing all pairwise distances
- Take the first three patients in the Golub data set

```
dist(t(expr(Golub_Merge[, 1:3])))
##          39          40
## 40 101530.75
## 42  94405.04  89502.29
```

- Patient 42 is more similar (closer) to patient 39 than patient 40 (distance of 94405.04 vs 101530.75)
- Patient 39 is more similar (closer) to 42 than patient 40 (distance of 94405.04 vs 101530.75)

2015 DATA: AGGLOMERATIVE HIERARCHICAL CLUSTERING



CLUSTERS

- ▶ Let c_1, c_2, \dots, c_n denote the n samples
- ▶ Define a cluster to be a set of patients
 - ▶ (c_1) is a cluster with one member: c_1
 - ▶ (c_1, c_3) is a cluster of two members: c_1 and c_3
 - ▶ (c_1, c_2, c_3) is a cluster of three members of c_1, c_2 and c_3
- ▶ Note that c_1 and (c_1) are different entities

NOTION OF A LINKAGE

- ▶ The distance measure quantified the distance between two points
- ▶ In clustering, you need to think about the criterion to link (merge) the clusters
- ▶ maximum distance (aka complete linkage)
- ▶ average distance (aka average linkage)
- ▶ minimum distance (aka single linkage)

AGGLOMERATIVE HIERARCHICAL CLUSTERING

- ▶ Agglomerate: To form clusters
- ▶ Let each of the n points be its own cluster (n clusters each with one single member)
- ▶ Find the pair of clusters that is most similar
- ▶ Merge these two
- ▶ Now you have $n - 1$ clusters (1 cluster with two members and $n - 2$ clusters each with a single member)
- ▶ Compute the similarities between the $n - 2$ "old" clusters with the new cluster
- ▶ Repeat the last two steps until all members have been merged into a single cluster.

CLUSTERING CITIES BY DISTANCES

	ATL	BOS	ORD	DCA
ATL	0	934	585	542
BOS	934	0	853	392
ORD	585	853	0	598
DCA	542	392	598	0

CLUSTERING CITIES BY DISTANCES (SINGLE LINKAGE)

	ATL	BOS	ORD	DCA
ATL	0	934	585	542
BOS	934	0	853	392
ORD	585	853	0	598
DCA	542	392	598	0

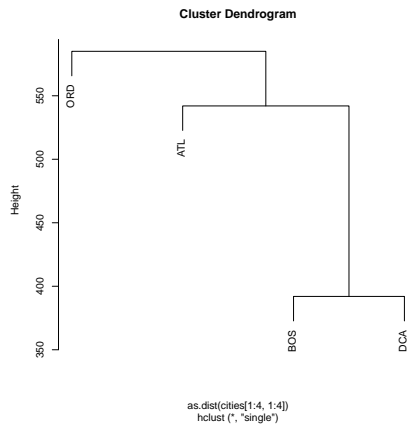
	DCA-BOS	ATL	ORD
DCA-BOS	0	542	598
ATL	542	0	585
ORD	598	585	0

CLUSTERING CITIES BY DISTANCES (SINGLE LINKAGE)

	DCA-BOS	ATL	ORD
DCA-BOS	0	542	598
ATL	542	0	585
ORD	598	585	0

	DCA-BOS-ATL	ORD
DCA-BOS-ATL	0	585
ORD	585	0

FOUR AIRPORTS (SINGLE LINKAGE)



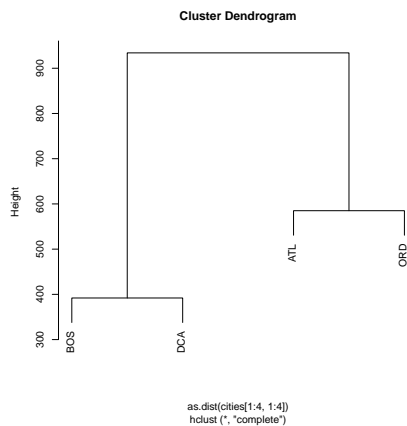
CLUSTERING CITIES BY DISTANCES (COMPLETE LINKAGE)

	ATL	BOS	ORD	DCA
ATL	0	934	585	542
BOS	934	0	853	392
ORD	585	853	0	598
DCA	542	392	598	0

	DCA-BOS	ATL	ORD
DCA-BOS	0	934	853
ATL	934	0	585
ORD	853	585	0

	DCA-BOS	ATL-ORD
DCA-BOS	0	934
ATL-ORD	934	0

FOUR AIRPORTS (COMPLETE LINKAGE)



FOUR AIRPORTS (SIDE BY SIDE)

	ATL	BOS	ORD	DCA
ATL	0	934	585	542
BOS	934	0	853	392
ORD	585	853	0	598
DCA	542	392	598	0

	DCA-BOS	ATL	ORD
DCA-BOS	0	934	853
ATL	934	0	585
ORD	853	585	0

	DCA-BOS	ATL-ORD
DCA-BOS	0	934
ATL-ORD	934	0

Table: Complete Linkage

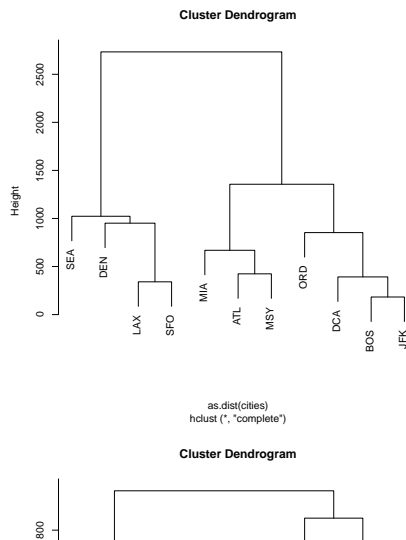
	ATL	BOS	ORD	DCA
ATL	0	934	585	542
BOS	934	0	853	392
ORD	585	853	0	598
DCA	542	392	598	0

	DCA-BOS	ATL	ORD
DCA-BOS	0	542	598
ATL	542	0	585
ORD	598	585	0

	DCA-BOS-ATL	ORD
DCA-BOS-ATL	0	585
ORD	585	0

Table: Single Linkage

ALL AIRPORTS (COMPARISON)

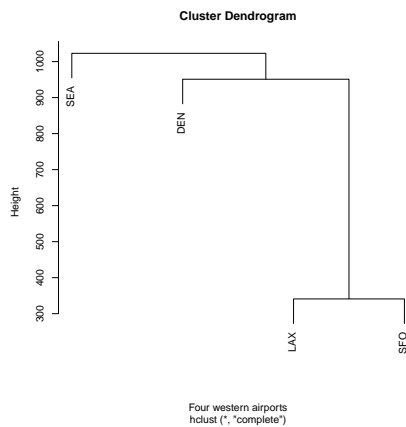


WESTERN AIRPORTS: EXERCISE

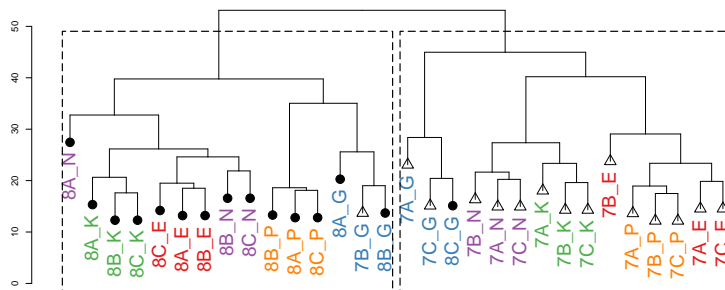
Carry out hierarchical clustering with complete linkage

```
##      DEN LAX  SEA SFO
## DEN    0 836 1023 951
## LAX  836  0  957 341
## SEA 1023 957  0  681
## SFO  951 341 681  0
```

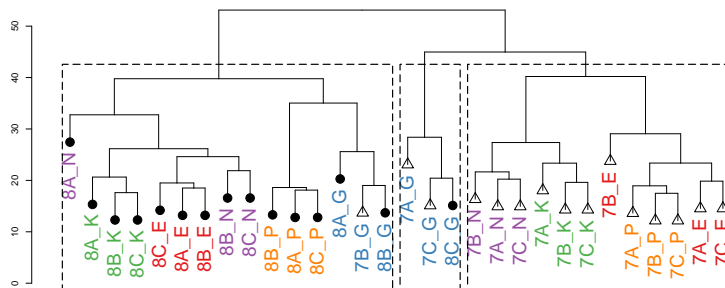
WESTERN AIRPORTS: SOLUTION



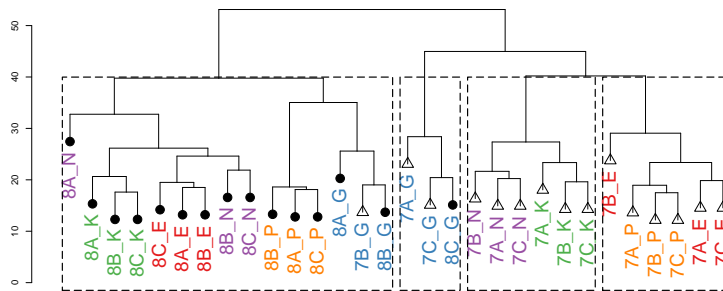
2015 DATA: AGGLOMERATIVE HIERARCHICAL CLUSTERING COMPLETE LINKAGE



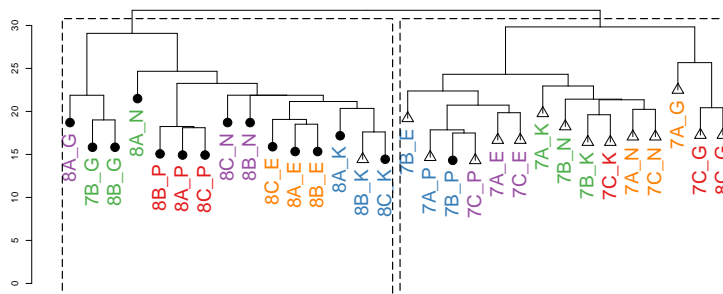
2015 DATA: AGGLOMERATIVE HIERARCHICAL CLUSTERING COMPLETE LINKAGE



2015 DATA: AGGLOMERATIVE HIERARCHICAL CLUSTERING COMPLETE LINKAGE



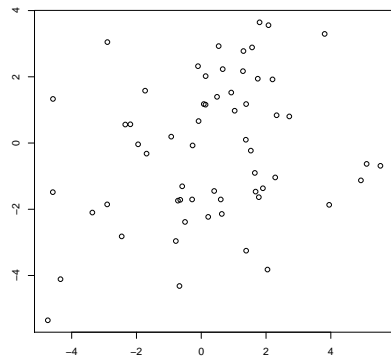
2015 DATA: AGGLOMERATIVE HIERARCHICAL CLUSTERING SINGLE LINKAGE



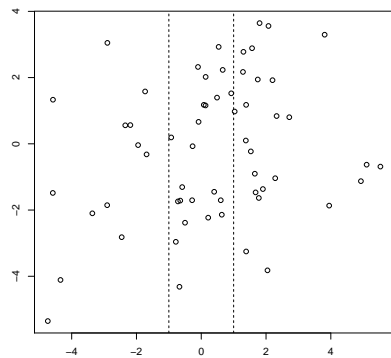
k -MEANS CLUSTERING

- Specify a number of potential clusters (k)
- Split of the data (either randomly or based on some previous results) into k partitions
- Compute the mean (aka centroid) for each partition
- For the first point (sample) determine the *nearest* centroid
- The closeness is typically quantified using the Euclidean distance
- Assign that point to that center
- Repeat for points 2 through n
- Assess the fit using the intra-cluster variance
- Repeat as needed.

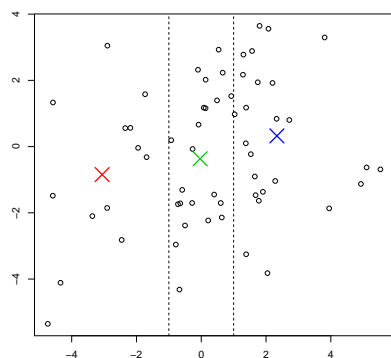
k -MEANS CLUSTERING: DATA



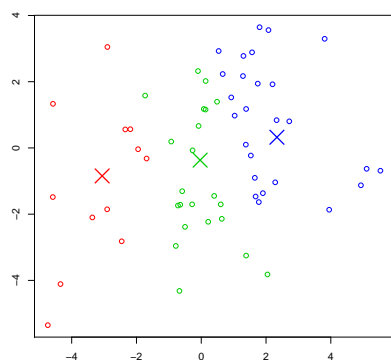
k -MEANS CLUSTERING: INITIAL CLUSTERS



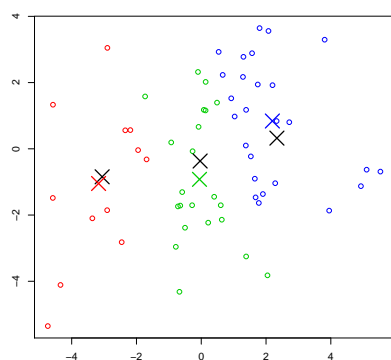
k -MEANS CLUSTERING: INITIAL CENTERS



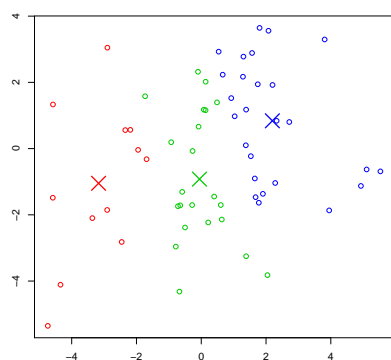
k -MEANS CLUSTERING: LABEL POINTS ACCORDING TO CENTERS



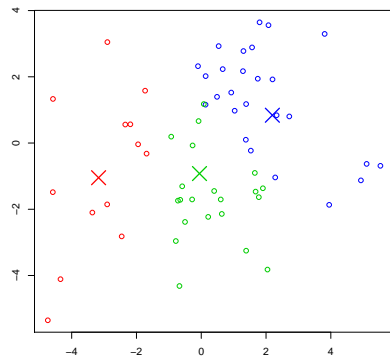
k -MEANS CLUSTERING: UPDATE CENTERS



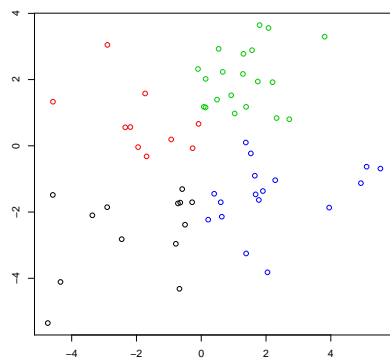
k -MEANS CLUSTERING: UPDATE CENTERS



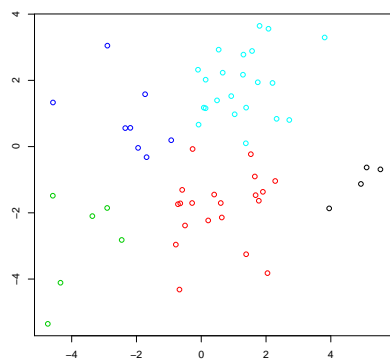
k -MEANS CLUSTERING: UPDATE POINTS



WHY NOT 4 CLUSTERS?



WHY NOT 5 CLUSTERS?



k -MEANS

- ▶ This is an example of *non-hierarchical* clustering
- ▶ Need to specify the number of clusters up front
- ▶ Need to specify (deterministically or randomly) the centers of the clusters up front
- ▶ Results are sensitive to the choice of k and initial partitions
- ▶ Note: All the data points were simulated from a single cluster!
- ▶ 3-means divides this single cluster into 3 subclusters
- ▶ 5-means divides this single cluster into 5 subclusters

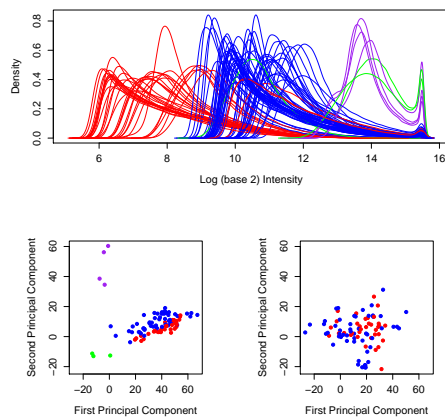
ORDINATION ANALYSIS AND DIMENSION REDUCTION

- ▶ Genome-wide profiling platforms are high-dimensional (m is large)
- ▶ Visualization beyond $m = 3$ not possible (for mortals)
- ▶ Representing the data by a lower dimensional format without losing too much information is desired.
- ▶ Two guiding principles:
 - ▶ Keep variables with highest variability
 - ▶ Reduce redundancy
- ▶ Two commonly used ordination analysis methods
 - ▶ Principal Components (plural) Analysis (PCA)
 - ▶ Multidimensional Scaling (MDS)

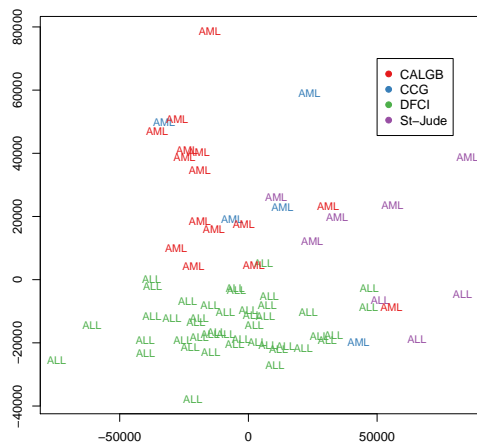
BATCH EFFECT DISCOVERY

- ▶ PCA and MDS methods are very useful for detecting batch effects
- ▶ Batch effects tend to be stronger than biological effects
- ▶ They also affect most probe sets (the biological effect may only be captured by a few)
- ▶ This can be an effective weapon in your QC arsenal (this is how I start any new analysis)

FROM CCR 2008 PAPER



ALL/AML DATA (GOLUB *et. al*; SCIENCE, 1999)



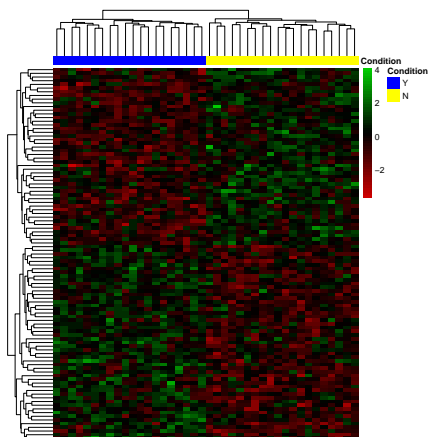
SEMI-SUPERVISED LEARNING

- ▶ Heatmap illustration:
 - ▶ Select a panel of probe-sets based on the two-sample t -test
 - ▶ Carry out hierarchical clustering with respect to the patients (the columns)
 - ▶ Carry out hierarchical clustering with respect to the probe sets in the panel (the rows)
 - ▶ Present the results using a heatmap
- ▶ Some consider this an *unsupervised* analysis as the hierarchical clustering algorithm is unaware of the classes
- ▶ This is not an accurate assessment: It is semi-supervised in the sense that we are picking genes based on the phenotype
- ▶ A procedure is *unsupervised* if the class info is only used for annotation

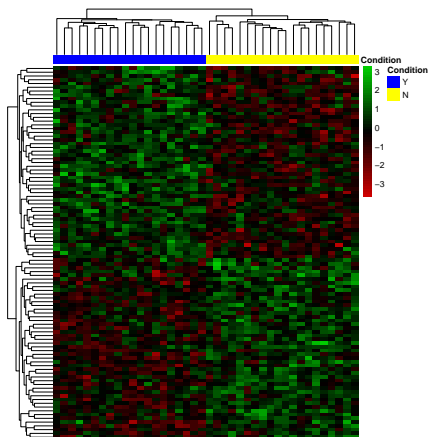
R CODE TO SIMULATE HEATMAP

```
simulate.noise.heatmap = function(n, m, alpha) {  
  # Simulate Expression Matrix  
  EXPRS = matrix(rnorm(2 * n * m), m, 2 * n)  
  grp = factor(rep(0:1, c(n, n)))  
  rownames(EXPRS) = paste("Gene", 1:m, sep = "")  
  colnames(EXPRS) = paste("patient id", 1:(2 * n), sep = "")  
  
  # Get the two sample t-statistics  
  pvals = rowttests(EXPRS, grp)$p.value  
  topgenes = which(pvals < alpha)  
  EXPRS = EXPRS[topgenes, ]  
  annodat = data.frame(Condition = ifelse(grp == 0, "N", "Y"), row.names = colnames(EXPRS))  
  pheatmap(EXPRS, border_color = NA, show_rownames = FALSE, show_colnames = FALSE,  
    annotation_col = annodat, color = colorRampPalette(c("red3", "black",  
      "green3"))(50), annotation_colors = list(Condition = c(Y = "blue",  
        N = "yellow")))  
  return(length(topgenes))  
}
```

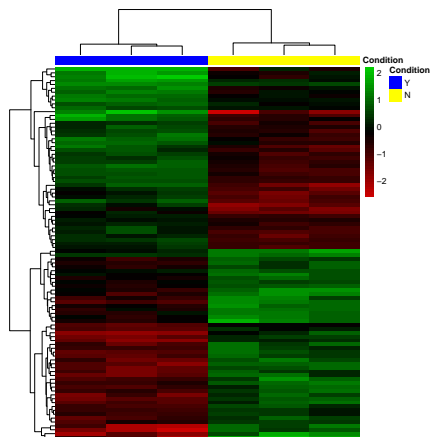
HEATMAP EXAMPLE: $m = 20,000, n = 20, \alpha = 0.005$



HEATMAP EXAMPLE: $m = 40,000, n = 20, \alpha = 0.0025$



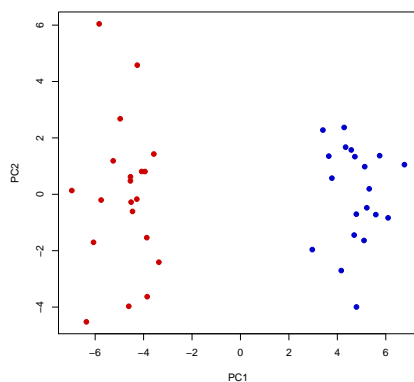
HEATMAP EXAMPLE: $m = 20,000, n = 3, \alpha = 0.005$



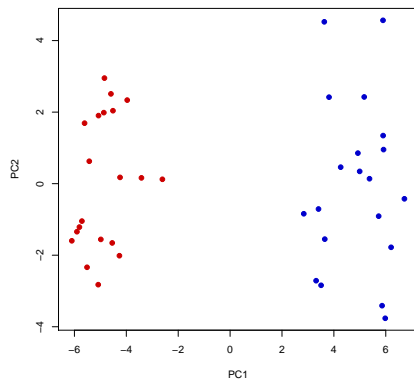
R CODE TO SIMULATE PC

```
simulate.noise.PC = function(n, m, alpha) {
  # Simulate Expression Matrix
  EXPRS = matrix(rnorm(2 * n * m), m, 2 * n)
  grp = factor(rep(0:1, c(n, n)))
  # Get the two sample t-statistics
  pvals = rowttests(EXPRS, grp)$p.value
  topgenes = which(pvals < alpha)
  EXPRS = EXPRS[topgenes, ]
  annodat = data.frame(Condition = ifelse(grp == 0, "N", "Y"), row.names = colnames(EXPRS))
  PC = cmdscale(dist(t(EXPRS)))
  plot(PC, xlab = "PC1", ylab = "PC2", col = ifelse(grp == 0, "red3", "blue3"),
        pch = 19)
  return(length(topgenes))
}
```

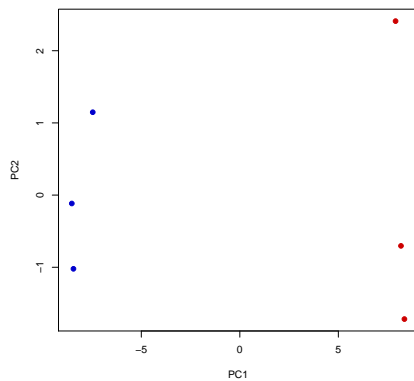
HEATMAP EXAMPLE: $K = 20000, n = 20, \alpha = 0.005$



HEATMAP EXAMPLE: $K = 40000, n = 20, \alpha = 0.0025$



HEATMAP EXAMPLE: $K = 20000, n = 3, \alpha = 0.005$



REMINDER: A SELF-FULFILLING PROPHECY

- ▶ Statistical methods for unsupervised learning guarantee one thing
- ▶ They will return a clustering of your data
- ▶ What they do not guarantee and are invariably unable to verify, is the biological relevance or reproducibility of the clustering
- ▶ In light of this Self-fulfilling Prophecy, these methods should be used with utmost care