20.109 RNA-seq Analysis Key

Amanda Kedaigle, Ernest Fraenkel

3/3/2017

Day 6 Part 2 - Clustering

An important technique in many bioinformatic analyses is clustering, a way to assign similar samples to groups. We'll explore two types of clustering by hand before returning to R.

1. **Hierarchical Clustering** builds a hierarchy of clusters. In the agglomerative, or bottom-up, method of hierarchical clustering, the two most similar samples or clusters are joined into one cluster repetitively. Work out the hierarchical clustering of the following 2D data points, using Euclidean distance (the length of a straight line between two points) and complete linkage (once a cluster contains more than one point, calculate the distance between two clusters as the longest distance between any two points in the respective clusters). Show each step, and when you're done, draw a dendrogram to show the results.

   Point 1: (0,0)

   Point 2: (3,0)

   Point 3: (0,6)

   Point 4: (21,2)

   Point 5: (23,2)

Round 1: Dist 1 to 2: 3.000
Dist 1 to 3: 6.000
Dist 1 to 4: 21.095
Dist 1 to 5: 23.087
Dist 2 to 3: 6.708
Dist 2 to 4: 18.111
Dist 2 to 5: 20.100
Dist 3 to 4: 21.378
Dist 3 to 5: 23.345
Dist 4 to 5: 2.000

New clusters: (1), (2), (3), (4,5)

Round 2:
Linkage cluster 1 to 2: 3.000
Linkage cluster 1 to 3: 6.000
Linkage cluster 1 to 4: 23.087
Linkage cluster 2 to 3: 6.708
Linkage cluster 2 to 4: 20.100
Linkage cluster 3 to 4: 23.34

New clusters: (1,2), (3), (4,5)

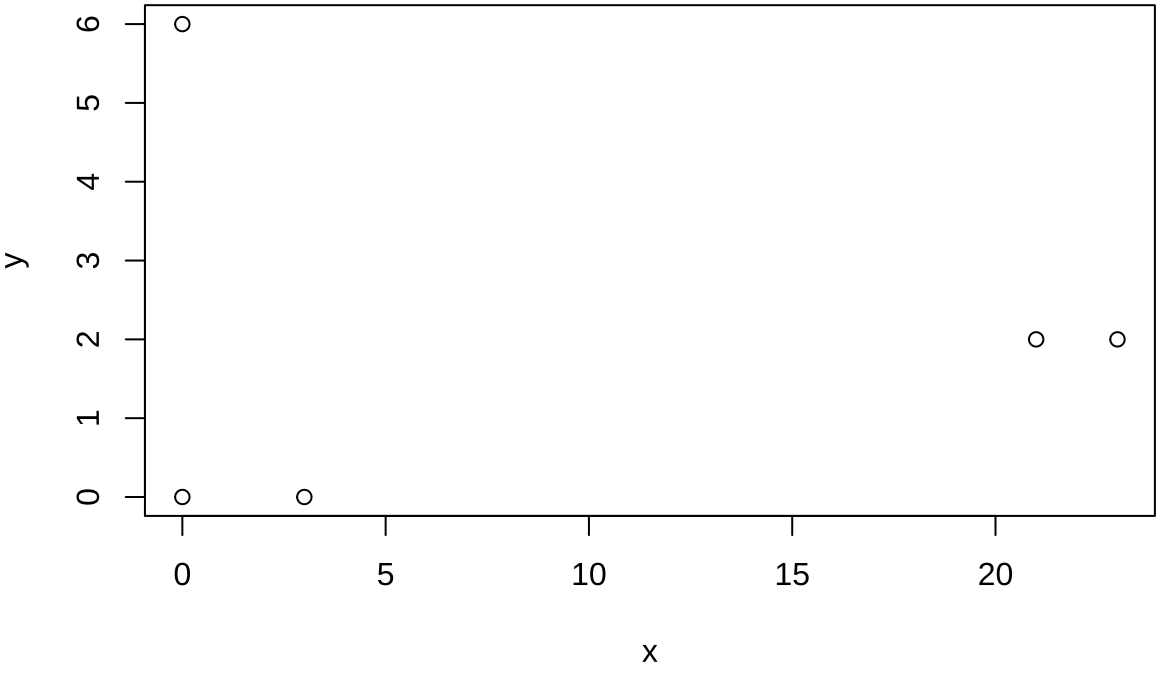
Round 3:
Linkage cluster 1 to 2: 6.708
Linkage cluster 1 to 3: 23.087
Linkage cluster 2 to 3: 23.34

New clusters: (1,2,3), (4,5)

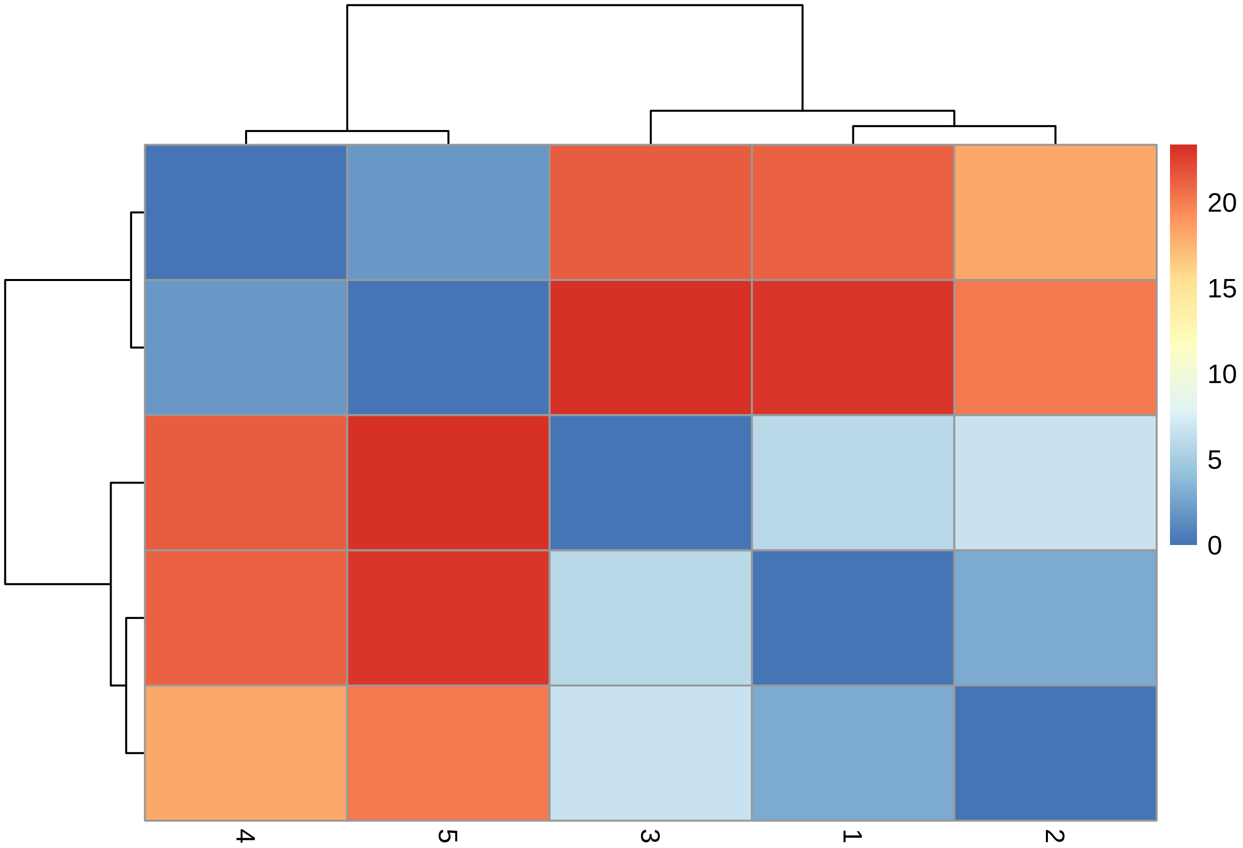
Round 4:
Linkage cluster 1 to 2: 23.345

Now, open the script "intro_clustering.R" and run the code for Problem 1 to see if you get the same dendrogram as R. Save your heatmap to a jpeg image file by clicking Export in the Plots window.

```
#Plot the given points
x = c(0,3,0,21,23)
y = c(0,0,6,2,2)
plot(x,y)

#Create a matrix to hold all the points, with x and y as rows
m = matrix(nrow=2, ncol=5)
m[1,] = x
m[2,] = y

#We've created our matrix with the x and y coordinates ("genes") as rows
#and the different points ("samples") as columns. This is how most biological
#matrices are organized, but many functions expect it to be the other way
#around, so let's also get the transverse of our matrix
trm = t(m)

#Heatmap and dendrogram. If you haven't already, load the package.
library("pheatmap")
```



```
#calculate Euclidean distance between the points
pointsDist = dist(trm, method = "euclidean")
#Draw a heatmap of the distances, with dendrograms from heirarchical clustering
pheatmap(pointsDist, labels_col=seq(from=1, to=5))
```



2. **K-means Clustering** is another clustering technique. In this algorithm, you decide ahead of time on a value for k, which is the number of clusters you'll get. You randomly choose k points to be the center or "centroid" of each cluster, and then iteratively assign nearby points to those clusters. Once a cluster has more than one point, the "centroid" is the average of the points in that cluster. Cluster the above points by k-means clustering, with k=3 by hand. Show your work.

- Use points 3, 4, and 5 as your initial centroids.
- Use points 1, 3, and 4 as your initial centroids.

Starting with points 3, 4,and 5:

- **Assign point 1 to the cluster with point 3. The centroid is now (0,3)**
- **Assign point 2 to the cluster with centroid (0,3). The centroid is now (1,2)**
- **Point 4 remains its own cluster**
- **Point 5 remains its own cluster**
- **Repeating the assignments does not cause the centroids to change any more, so the algorithm is done**

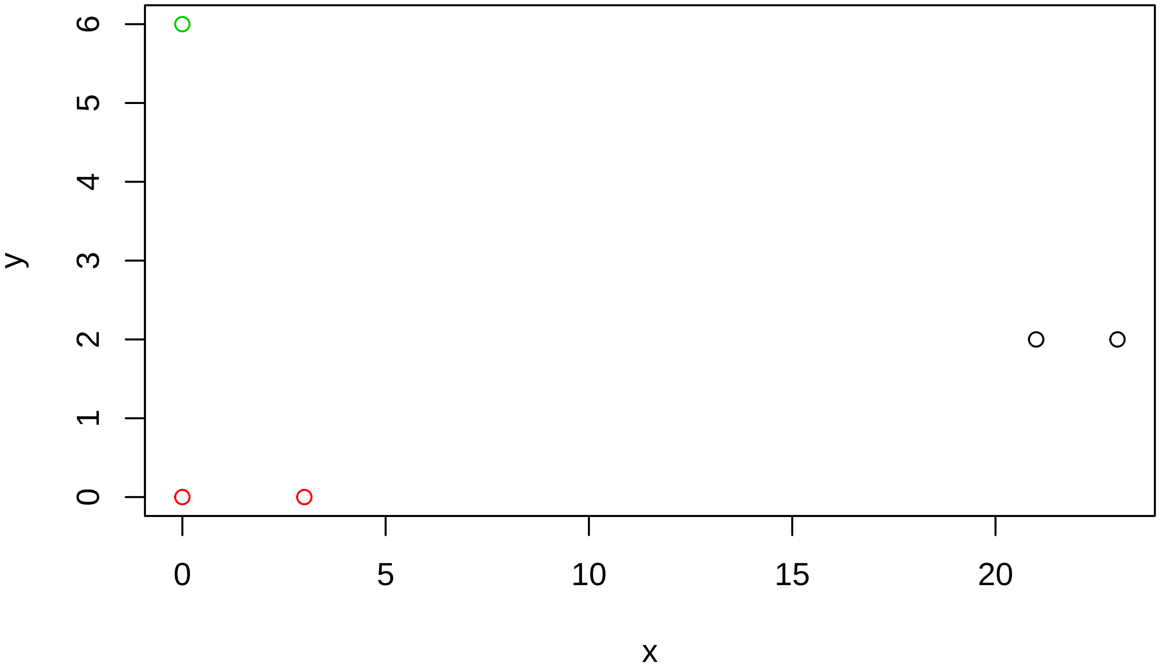
Starting with points 1, 3, and 4:

- **Point 1 remains its own cluster**
- **Assign point 2 to the cluster with point 1. The centroid is now (1.5,0)**
- **Point 3 remains its own cluster**
- **Point 4 remains its own cluster**
- **Assign point 5 to the cluster with point 4. The centroid is now (22,2)**

- **Repeating the assignments does not cause the centroids to change any more, so the algorithm is done**

Finally, try out the code in intro_clustering.R and save your colored plot as a jpeg image.

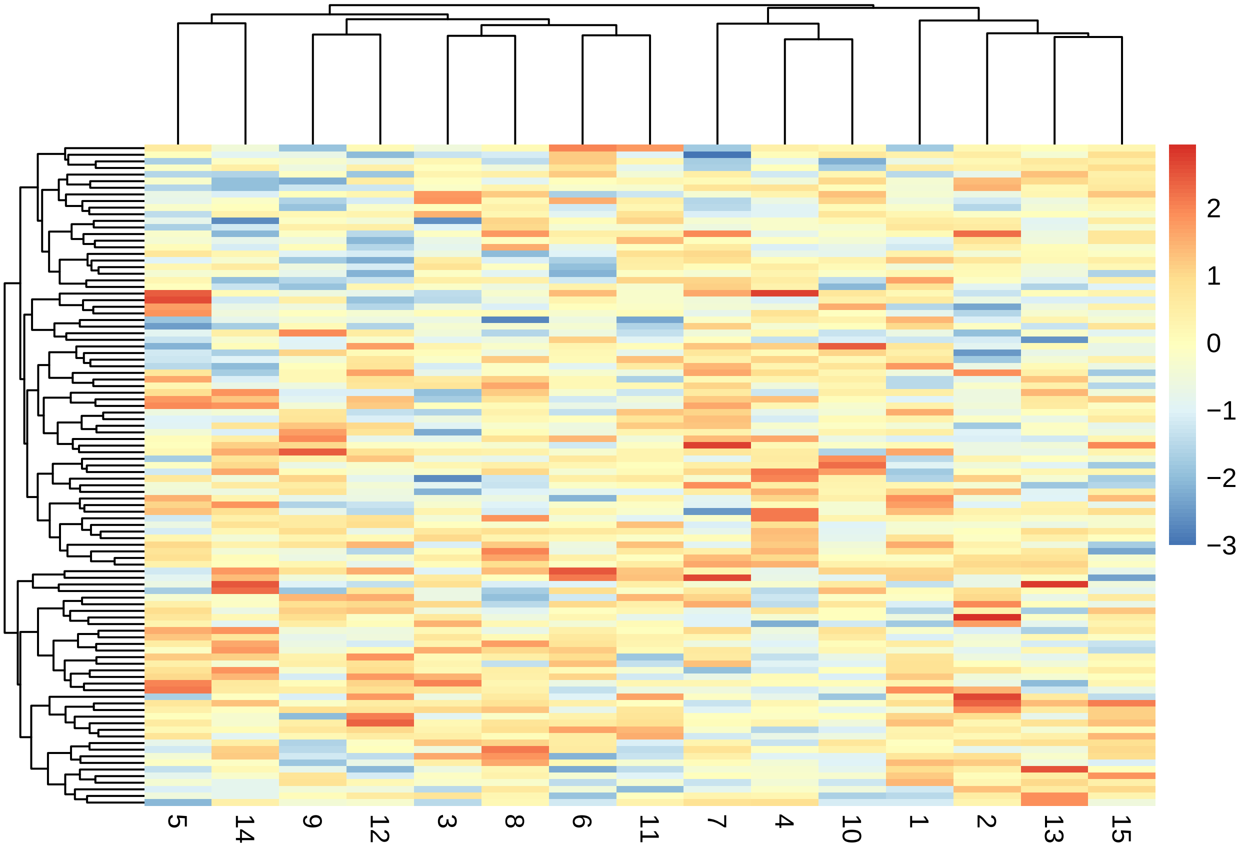
```
#Run kmeans with k=3. We'll tell the algorithm to choose 5 sets of random
#starting points and give us the most common answer
kclusters = kmeans(trm, centers=3, nstart=5)
#Plot the data, with different colors for the 3 clusters
clus = kclusters$cluster
plot(x, y, col=clus)
```



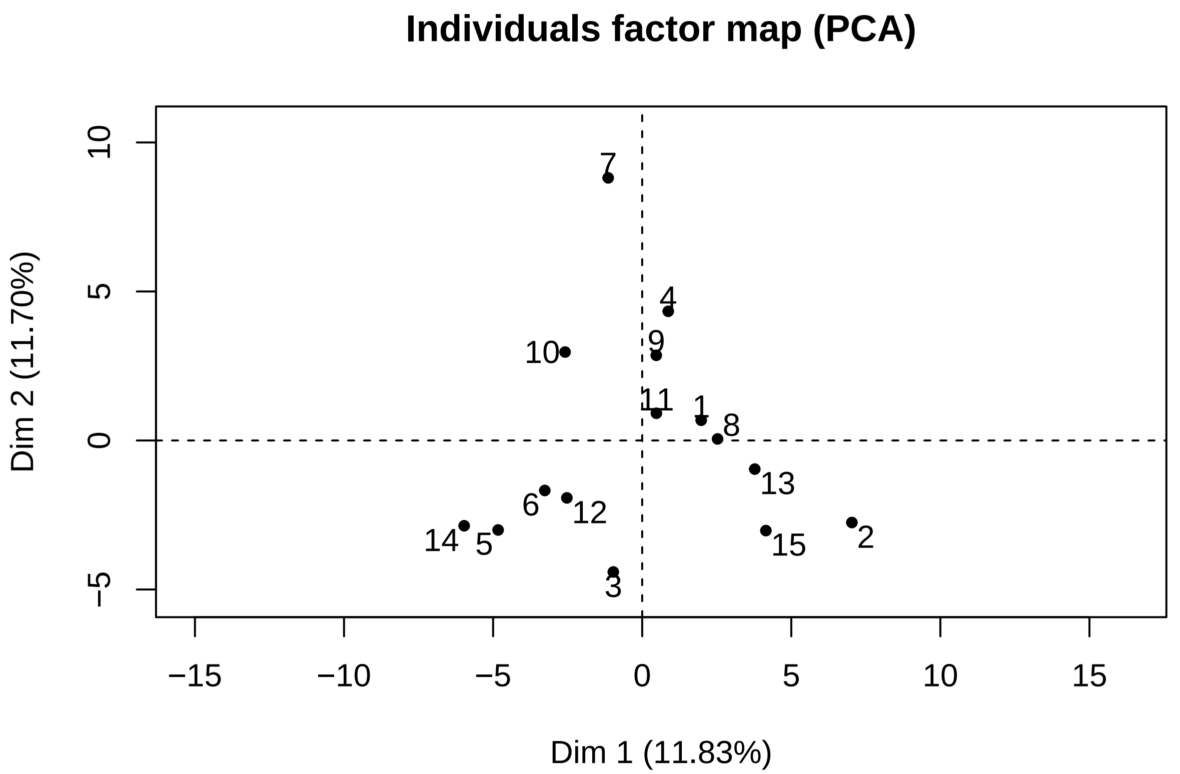
3. **Principal Components Analysis** is another technique, in which the data points (which often have more dimentions than our 2D points) are projected onto the 2D plane such that they spread out in the two directions that explain most of the difference between them. The x-axis (PC1) is the direction that separates the data points the most. The y-axis (PC2) is a direction (it must be orthogonal to the first direction) that separates the data the second most. The percent of the total variance that is contained in the direction is printed in the axis label. Here we'll use R to cluster a larger high-dimensional dataset. Use the code in intro_clustering.R to generate a random dataset for 15 samples, with information on 100 genes each. Create a heatmap and run principle components analysis on these data.

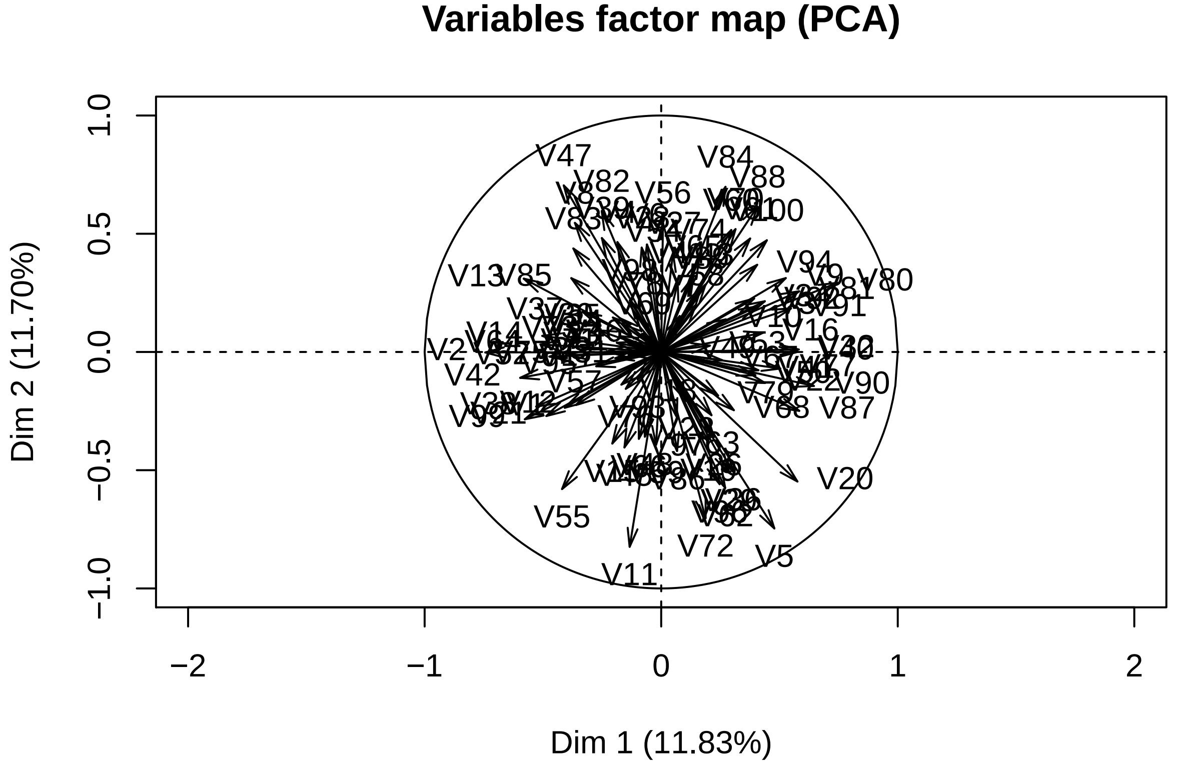
- After hierarchical clustering, save your heatmap as a jpg file. What are the two most similar samples to sample 7?
- Now run principle components analysis on this data. Save the individuals factors graph to a jpeg image. Are the same samples the most similar to sample 7 with this technique? Notice how much of the variation is explained in this 2-D chart. Do you think this is a good representation of the data?

```
#Generate a matrix of random data for 15 samples, where we've measured the
#differential RNA levels of 100 genes in every sample. We'll set a seed so
#that everyone's data looks the same.
set.seed(20)
exSample = rnorm(100)
exData = replicate(15, rnorm(100))
#Hierarchical clustering & heatmap
pheatmap(exData, labels_col=seq(from=1, to=15))
```



```
#PCA - we'll use a package called FactoMineR for this
library(FactoMineR)
result = PCA(t(exData))
```


Individuals factor map (PCA)


Variables factor map (PCA)

The closest samples to sample 7 are samples 4 and 10. Note that sample 11 is next to sample 7 in the heatmap, but is not very closely related according to the dendrogram. Samples 4 and 10 are also close to 7 in the PCA, along with sample 9. Only ~24% of the variance in this random data is explained by the two PCA axes, so this is not a great representation of the data.