Assignment 3: Task 1 - Cluster Analysis

Group 11

DATE

1. Problem Statement

The presented data contains students' enrolment activites for 72 universities (observations), which is further subdivided into 13 faculties (variables). The task is to perform several cluster methods in order to identify for which universites students have a similar enrolment behavior. The analysis starts with some descriptive statistics, followed by a short comment regarding the assumtions. Afterwards hierarchical methods as well as non-hierarchical methods and others are applied. Finally, a conclusion regarding the best performing clustering method is made.

2. Descriptive Statistics

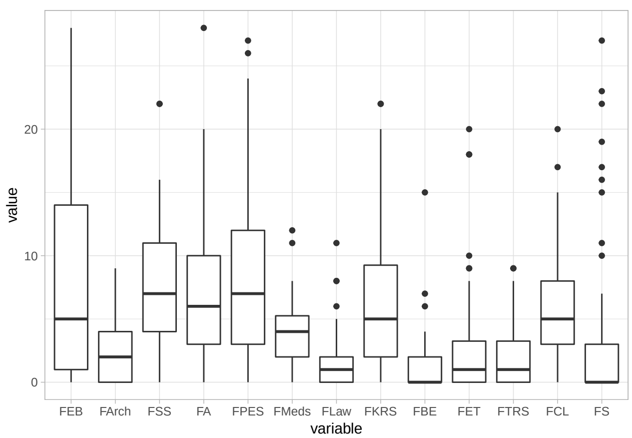
```
unistudis <- as.tibble(read.table("../data/unistudis.txt", header=T))

descr(unistudis[,-length(unistudis)], style = "rmarkdown",
      stats = c("mean", "sd", "min", "q1", "med", "q3", "max", "pct.valid"))
```

```
## ### Descriptive Statistics
## **Data Frame:** unistudis
## **N:** 72
##
## |          |    FEB |  FArch |    FSS |     FA |   FPES |  FMeds |   FLaw |   FKRS |
## |--------------:|-------:|-------:|-------:|-------:|-------:|-------:|-------:|-------:|
## |      **Mean** |   8.31 |   2.15 |   7.99 |   7.10 |   8.04 |   3.89 |   1.62 |   6.47 |
## |   **Std.Dev** |   8.60 |   2.09 |   4.65 |   5.73 |   6.18 |   2.60 |   2.15 |   6.13 |
## |       **Min** |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
## |        **Q1** |   1.00 |   0.00 |   4.00 |   3.00 |   3.00 |   2.00 |   0.00 |   2.00 |
## |    **Median** |   5.00 |   2.00 |   7.00 |   6.00 |   7.00 |   4.00 |   1.00 |   5.00 |
## |        **Q3** |  14.00 |   4.00 |  11.00 |  10.00 |  12.00 |   5.50 |   2.00 |   9.50 |
## |       **Max** |  28.00 |   9.00 |  22.00 |  28.00 |  27.00 |  12.00 |  11.00 |  22.00 |
## | **Pct.Valid** | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
##
## Table: Table continues below
##
##
##
## |          |    FBE |    FET |   FTRS |    FCL |     FS |
## |--------------:|-------:|-------:|-------:|-------:|-------:|
## |      **Mean** |   1.14 |   2.75 |   1.92 |   5.90 |   3.04 |
## |   **Std.Dev** |   2.23 |   4.20 |   2.52 |   4.24 |   6.10 |
## |       **Min** |   0.00 |   0.00 |   0.00 |   0.00 |   0.00 |
## |        **Q1** |   0.00 |   0.00 |   0.00 |   3.00 |   0.00 |
## |    **Median** |   0.00 |   1.00 |   1.00 |   5.00 |   0.00 |
## |        **Q3** |   2.00 |   3.50 |   3.50 |   8.00 |   3.00 |
## |       **Max** |  15.00 |  20.00 |   9.00 |  20.00 |  27.00 |
```

```
## | **Pct.Valid** | 100.00 | 100.00 | 100.00 | 100.00 | 100.00 |
```

First of all there are no missing values in the data, which is seen in the Pct.Valid row. Other than that the variables seem to differ in terms of mean and spread. FEB, FSS, FA, FPES, FKRS and FCL have relatively high mean values compared to the remaining ones. This means that students of the explicitly mentioned facultes on average enrl more frequently into language coureses. In terms of standard deviations FEB, FSS, FA, FPES, FKRS, FET, FCL and FS seem to have a high spread compared to the other faculties. So for these faculties students enrolment behavior differs strongly from university to university, whereas for other faculties it doesn't.

```r
# Detecting Outliers
melt(unistudis[,-length(unistudis)]) %>% ggplot(aes(x = variable, y = value)) +
  geom_boxplot() +
  theme_light()
```




```r
subset(unistudis, uniID == "16" | uniID == "46")
```

```
## # A tibble: 2 x 14
##     FEB FArch   FSS    FA  FPES FMeds  FLaw  FKRS   FBE   FET  FTRS   FCL
##   <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int> <int>
## 1     1     9    16     5    11     4     2     5    15     1     8     3
## 2    14     5     5    11    27     1     0     6     0     1     0     3
## # ... with 2 more variables: FS <int>, uniID <int>
```

In terms of outliers one can clearly see that some are present. Especially observation 46 and 16 seem to be problematic. Whereas observation 46 has outlier values at variable FS and FPES, observation 16 has outlier values at variable FArch, FSS, FBE and FTRS.

3. Assumptions

Even though there are no explicit assumptions when using cluster algorithms one has to consider the fact that variables with a higher spread will have a higher importance in hierarchical cluster algorithms. This argumentation goes along with outlier values, as they might be clustered in a cluster containing only the outlier value. Therefore, observation 16 and 46 are removed before centering and standardizing the data.

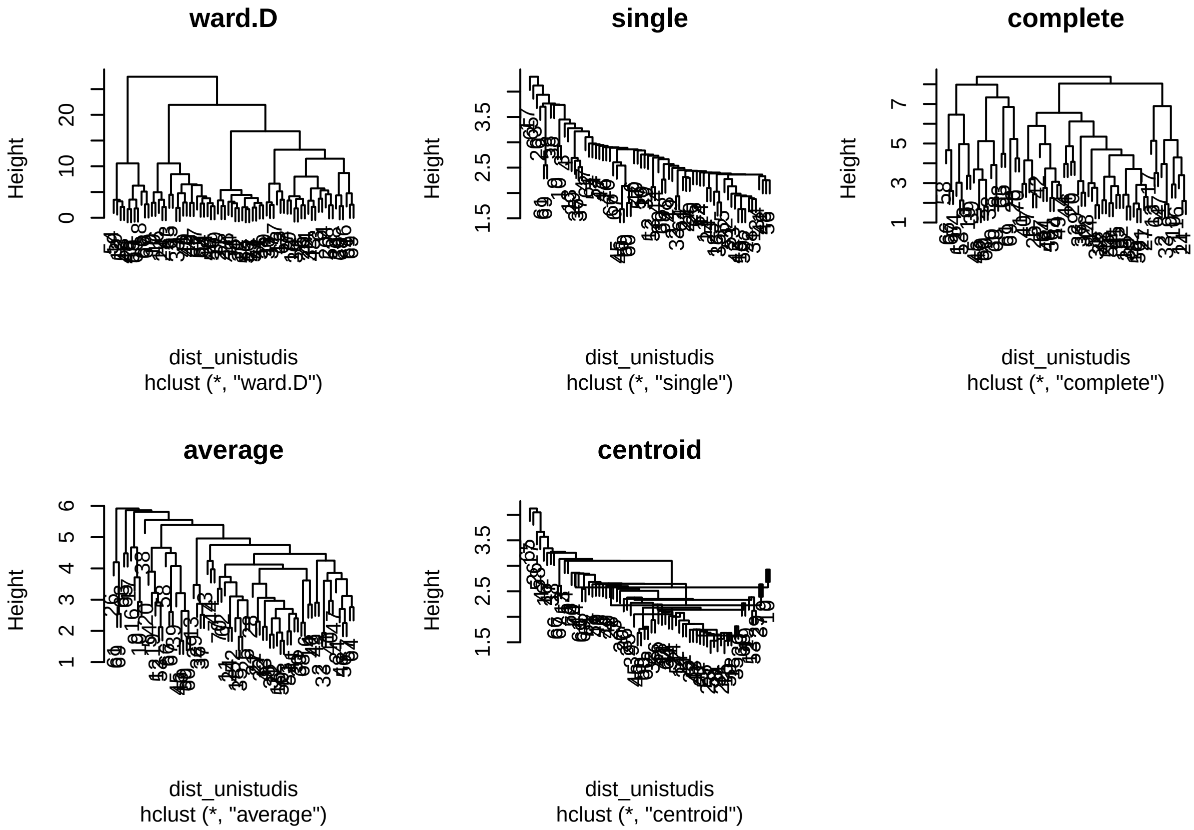
```r
# Remove observation 16 and 46
unistudis <- unistudis %>% filter(uniID != "16", uniID != "46")

# Standardize values
std_unistudis <- as.tibble(unistudis[,1:ncol(unistudis)-1] %>% scale(center = T, scale = T))
std_unistudis <- std_unistudis %>% mutate(uniID = unistudis$uniID) %>% dplyr::select(uniID, 1:13)
```

4. Method and Interpretation

4.1 Hierarchical Methods

The hierarchical methods applied in this section are single linkage, complete linkage, average linkage, centroid method and Ward's error sum of squares. The first step is to identify the correct amount of clusters based on either the visual analysis of the dendrograms or by investigating the drop in R^2. Analysing the dendrograms means examining the sizes of the changes in height in the dendrograms. A large change indicates the appropriate number of clusters. The authors decide to evaluate the dendrograms.

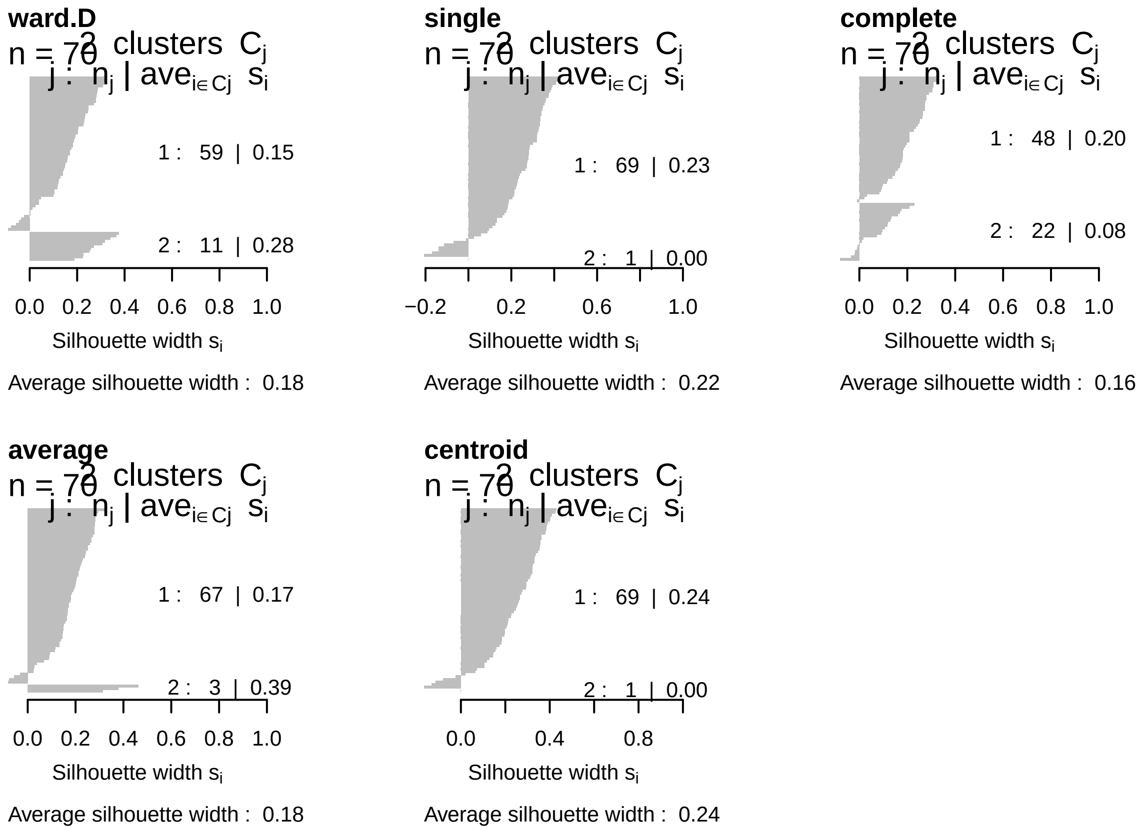
```r
# Calculate euclidean distance based on standardized values
dist_unistudis <- dist(std_unistudis[,-1])

# Apply hierarchical cluster methods
hclust_methods <-
  c("ward.D", "single", "complete", "average", "centroid")
hclust_results <- lapply(hclust_methods, function(m) hclust(dist_unistudis, m))
names(hclust_results) <- hclust_methods

# Plot dendrograms
par(mfrow = c(2,3))
hclust_dendro <- lapply(names(hclust_results), function(m) plot(hclust_results[[m]],
                                                               main = m))
names(hclust_dendro) <- hclust_methods
par(mfrow = c(1,1))
```



The dendrograms of Ward's method and complete indicate a two cluster solution, whereas based on the remaining dendrograms on could also decide for more clusters. The authors decide to continue with a two cluster solution.

Hence, the trees of each method are each cut into two clusters. To further evaluate whether the right amount of clusters is chosen the silhouette plots need to be analysed. On the other hand, silhouette plots not only provide information about the right amount of clusters chosen, but also give an indication about the degree of homogeneousity in each cluster. Silhouette values range from -1 to 1, whereas values close to -1 indicate observations poorly classified and values close to 1 vice versa. Observations with values close to 0 are intermediate cases, which can be assigned to one or another cluster equally likely.

```
# Assign observations to two clusters
noclust <- 2
hclust_cutree <- mapply(cutree, hclust_results, noclust)

# Create silhouette plots
par(mfrow = c(2,3))
sapply(colnames(hclust_cutree), function(m) plot(silhouette(hclust_cutree[,m],
                                                  dist_unistudis), main = m))
par(mfrow = c(1,1))
```



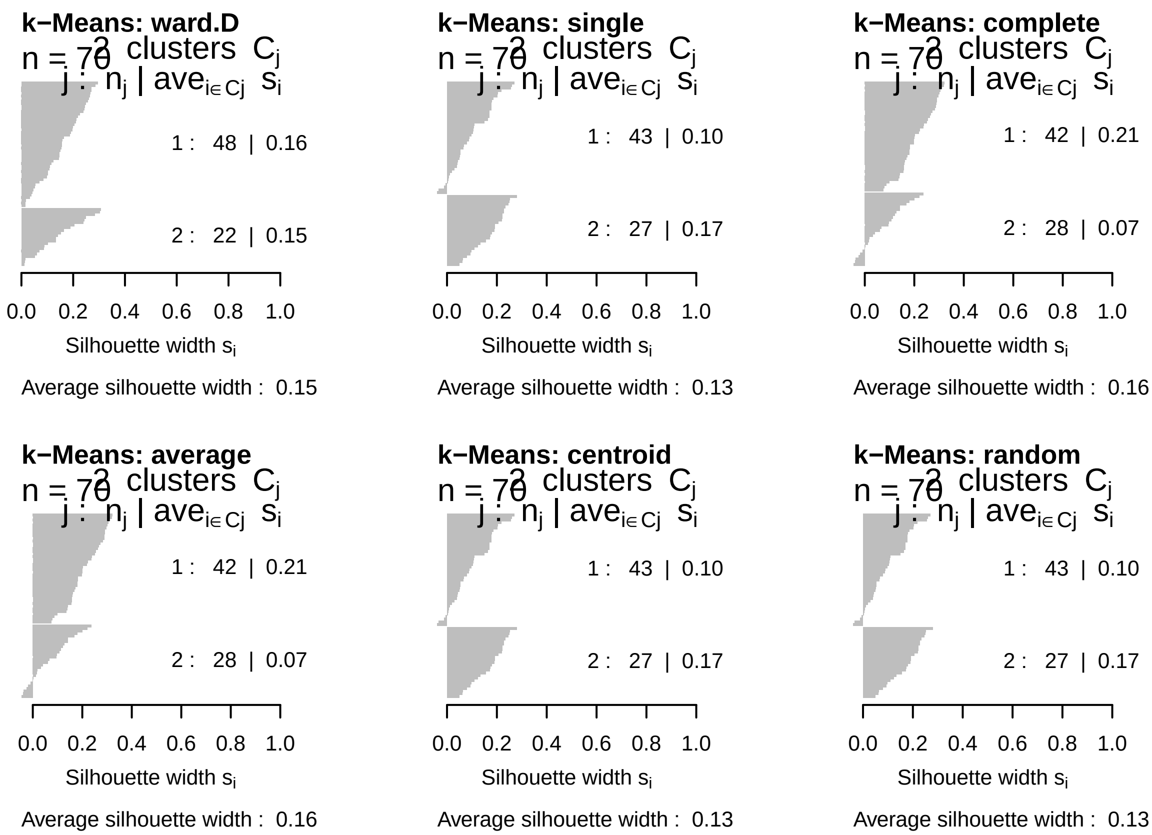
The silhouette plots show that the single linkage and centroid method cluster 69 in one cluster, which is a result of the chaining effect and gives basically no gain in information. The remaining models are better interpretalbe, as the relative frequency in the second cluster is higher. Nonetheless, silhouette scores of bigger than 0.4 are rare in each method, meaning that most of the observations are close to be intermediate cases. Therefore one might try to use non-hierarchical methods using input values from hierarchical methods to come to a better solution.

4.2 Non-Hierarchical and Model Based Methods

The non-hierarchical method used in this analysis is k-Means. As the result of k-Means is strongly dependent on the initial seeds, we are using random seeds as well as the results from each hierarchical cluster method as initial seeds. To evaluate the goodness of the methods the silhouette plots are again analysed.

```r
# Attach clusters to observations
std_unistudis_EXT <- as.tibble(cbind(std_unistudis[,-1], hclust_cutree))

# Calculate initials
initials <- lapply(hclust_methods,
                   function(m) aggregate(std_unistudis[,-1],
                                         list(as.vector(t(std_unistudis_EXT[,m]))), mean))
# k-Means with initials
clus_kmeans <- lapply(seq(1:length(hclust_methods)),
                      function(m) kmeans(std_unistudis[,-1], centers = initials[[m]][,-1]))

# k-means with random seeds
set.seed(1)
clus_kmeans[[length(clus_kmeans)+1]] <- kmeans(std_unistudis[,-1], centers = noclust)
```

```r
# Create silhouette plots
par(mfrow = c(2,3))
pnames <- paste("k-Means:", c(hclust_methods, "random"))
lapply(seq(1:length(clus_kmeans)),
       function(m) plot(silhouette(clus_kmeans[[m]]$cluster, dist_unistudis),
                        main = pnames[m]))
```

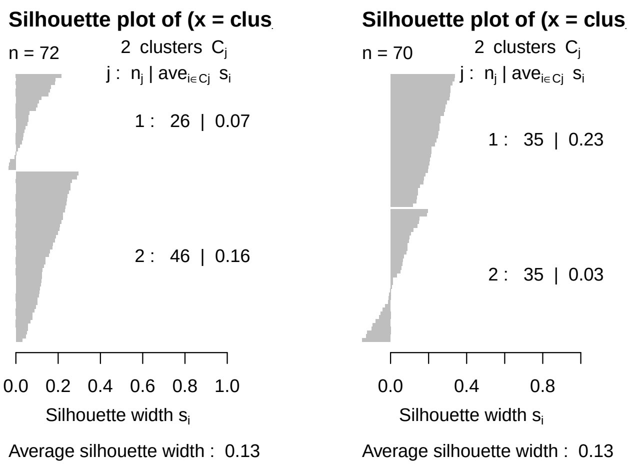


First of all each method clusters approximataly the same observations in each cluster. Other than that the silhouette plots of k-Means with Ward and average linkage seeding seem to be the best k-Means cluster solutions. This can also be validated when comparing the average silhouette width. K-Means with complete seeding performs bad in specifying the second cluster, whereas the remaining methods have difficulties in specifying the first cluster.

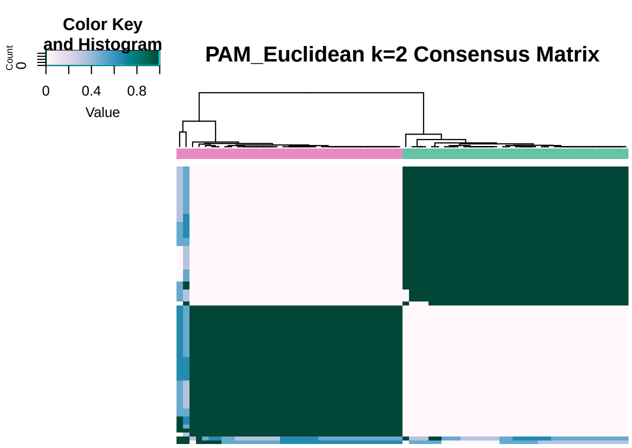
5. Alternative solutions

```r
# k-Mediods (Partitioning Around Mediods)
unistudis_all <- as.tibble(read.table("../data/unistudis.txt", header=T))
std_unistudis_all <- as.tibble(unistudis_all[,1:ncol(unistudis_all)-1] %>% scale(center = T, scale = T)
std_unistudis_all <- std_unistudis_all %>% mutate(uniID = unistudis_all$uniID) %>% dplyr::select(uniID,
dist_unistudis_all <- dist(std_unistudis_all[,-1])

clus_pam <- pam(std_unistudis_all[,-1], noclust)

# Model based clustering (selection based on BIC)
clus_mod <- Mclust(std_unistudis[,-1], G = 2:9)
```

```r
par(mfrow = c(1,2))
plot(silhouette(clus_pam$cluster, dist_unistudis_all))
plot(silhouette(clus_mod$classification, dist_unistudis))
```




```r
# Cluster ensemble
CC <- consensus_cluster(std_unistudis, nk = 2:4, p.item = 0.8, reps = 5,
                        algorithms = c( "pam", "diana", "hc", "gmm"))

pam.2 <- CC[, , "PAM_Euclidean", "2", drop = FALSE]
cm <- consensus_matrix(pam.2)
hm <- graph_heatmap(pam.2)
```




```r
ccomp <- consensus_evaluate(std_unistudis, CC, plot = FALSE)
```

Additionally, Partitioning Around Mediods (PAM) and model-based clustering was explored. PAM is supposed to be less sensitive to outliers compared to k-means, which is why we applied it to the full dataset. The silhouette plot did not show a homogenous within cluster structure. Another approach includes the model-based clustering which tries to estimate the distribution of cluster segments and maximize the probability that a sample comes from one distribution. This is also called soft partitioning, while the other approaches followed hard partitioning. We also explored ensemble methods as way of directly comparing different algorithms and cluster sizes. Specifically, we compared PAM and two hierarchical clustering methods, DIANA, HC and one model-based approach, GMM. DIANA is a divisive clustering algorithm while HC takes an agglomarative approach. In 5 rounds of applying each algorithm to bootstrapped subsamples of the data a consensus within the cluster assignment is reached. This can be visualized per algorithm and cluster size in a NxN Consensus Matrix, with N = # samples. The matrix values are within a $[0, 1]$ boundary with 1 indicating agreement across all iterations of sample assignment. The heatmap output is enclosed in the appendix. It showed high agreement within DIANA and PAM, especially for cluster size = 2. This means, that the clustering solution is unambigious. This takes a different approach in evaluation, as we are not considering the silhouette plot width but instead the agreement within an algorithm in regard of the cluster assignment.

The evaluation with respect to compactness and seperability (see appendix) shows that the algorithms produce equally compact clusters but DIANA and PAM generate solutions which are more separable. This analysis supports our previously chosen solution of two clusters. The model based approach seems to be not optimal for this task, as it generates clusters very different from the other algorithms. This might be caused by the strong assumptions of an underlying gaussian distribution of each cluster is not satisfied.

6. Conclusion

The performed cluster analysis consisted of the comparison of different hierarchical approaches and their dendogram and silhouette plots. The results were fed into k-means and k-mediods as one of several methods for initializing the cluster seeds. Additionally, model based and ensemble methods were explored. The best solution in terms of within cluster connectivity, consensus of the algorithm and silhouette plots was achieved by the k-means method using two clusters and initial seeds from the prior hierarchical solution.

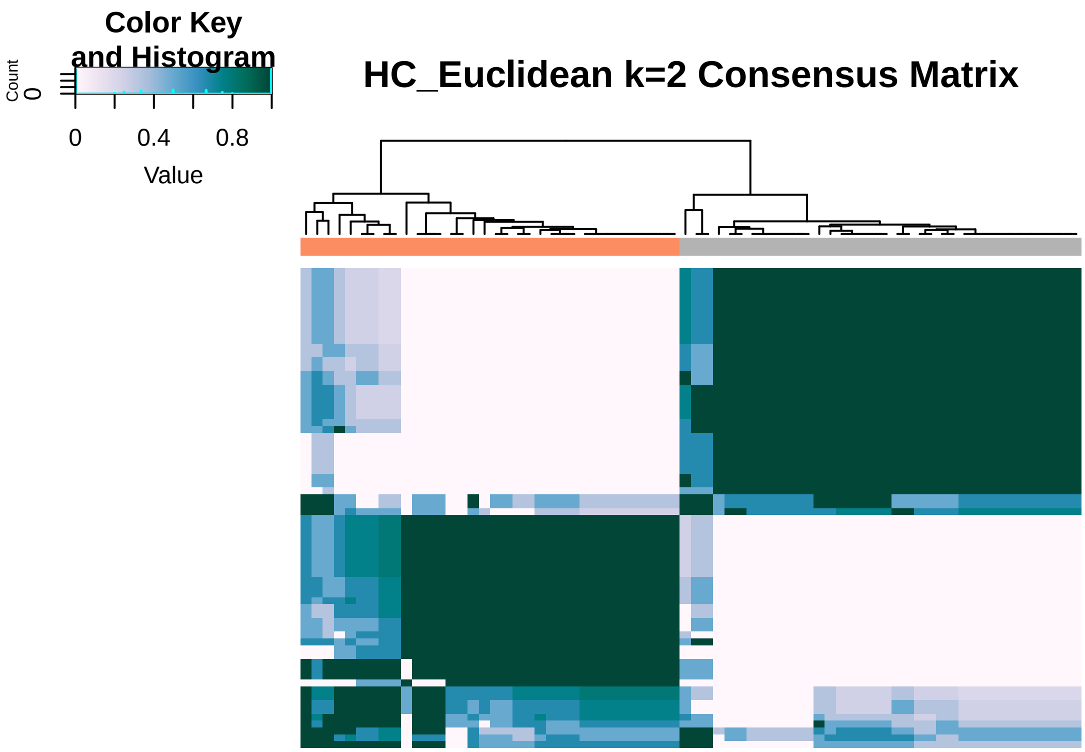
Appendix

```r
ccomp$ii
```

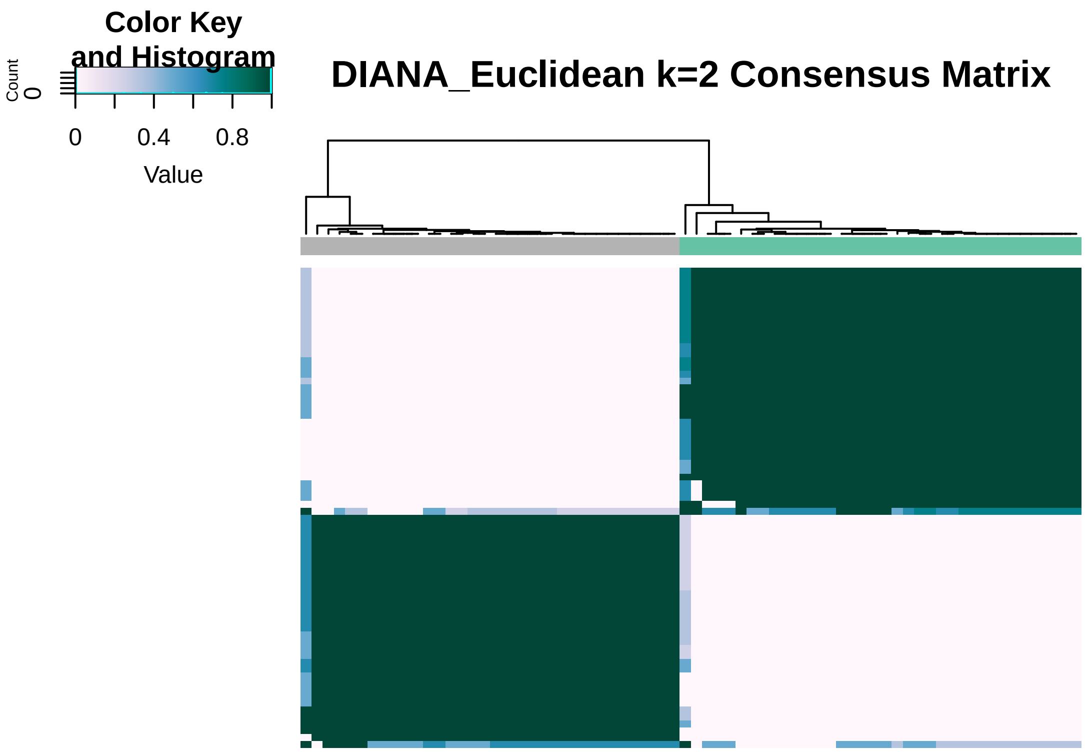
```
## $`2`
##        Algorithms calinski_harabasz       dunn       pbm         tau
## 1   PAM_Euclidean        181.987795 0.11538260 1131.8903 0.581166022
## 2 DIANA_Euclidean        181.883858 0.07434944 1134.1519 0.581628933
## 3    HC_Euclidean        177.031295 0.07092409 1108.0809 0.573450208
## 4             GMM          2.044554 0.03716404   13.4435 0.008195631
##        gamma    c_index davies_bouldin mcclain_rao     sd_dis   ray_turi
## 1 0.82163882 0.07706750      0.5545375   0.3796207 0.05526508 0.09341286
## 2 0.82228369 0.07698183      0.5535924   0.3795912 0.05524681 0.09338994
## 3 0.81072092 0.08171235      0.5601340   0.3844416 0.05545248 0.09594983
## 4 0.01158586 0.49597243      5.0400995   0.9932331 0.27497875 8.25368201
##       g_plus silhouette    s_dbw Compactness Connectivity
## 1 0.04459940 0.57903095 1.174606    13.89968     6.655952
## 2 0.04443918 0.57965458 1.334597    13.89678     7.405952
## 3 0.04733053 0.57112089 1.256852    14.02307    11.383730
## 4 0.24719365 0.01916799 1.921623    25.12392    85.213492
##
## $`3`
##        Algorithms calinski_harabasz       dunn       pbm        tau
## 1   PAM_Euclidean        214.520522 0.07463447 1814.5362 0.58582849
## 2 DIANA_Euclidean        163.355885 0.07882758 1334.0913 0.55243795
## 3    HC_Euclidean        180.831959 0.12248790 1457.8983 0.56279848
## 4             GMM          5.716517 0.03716404   58.6744 0.08090434
##       gamma    c_index davies_bouldin mcclain_rao     sd_dis  ray_turi
## 1 0.8825768 0.04275800      0.5868111   0.3117642 0.09972087 0.1111413
## 2 0.8139744 0.07341713      0.6072031   0.3563087 0.12370806 0.2146094
## 3 0.8380532 0.06154874      0.6210095   0.3387230 0.11520967 0.1714573
## 4 0.1216506 0.40044144      3.6913238   0.8815016 0.30590135 6.7648794
##       g_plus  silhouette    s_dbw Compactness Connectivity
## 1 0.02585710  0.51815128 1.710380    10.17542     14.14802
## 2 0.04282610  0.48693630 1.588319    10.92909     13.69960
## 3 0.03650273  0.49007509 1.920002    10.70475     11.09405
## 4 0.19416585 -0.02183993 3.607987    23.24932    114.09881
##
## $`4`
##        Algorithms calinski_harabasz       dunn        pbm        tau
## 1   PAM_Euclidean        227.863532 0.13673086 1781.66977 0.54513109
## 2 DIANA_Euclidean        232.949154 0.14669246 1843.78724 0.54971114
## 3    HC_Euclidean        231.009375 0.13673086 1921.00392 0.54884390
## 4             GMM          5.092078 0.04084787   37.65474 0.05906517
##        gamma    c_index davies_bouldin mcclain_rao    sd_dis   ray_turi
## 1 0.90250075 0.03161506      0.6570925   0.2771316 0.1445638  0.1313882
## 2 0.90954620 0.02934120      0.6546807   0.2738412 0.1416994  0.1247202
## 3 0.90704399 0.03035006      0.6426126   0.2755111 0.1420995  0.1274184
## 4 0.09618843 0.38473667      7.9241677   0.8896470 0.8829723 56.2782054
##       g_plus silhouette    s_dbw Compactness Connectivity
## 1 0.01777862  0.4473270      NaN    8.498332     22.13214
## 2 0.01651340  0.4567194      NaN    8.428514     18.52976
## 3 0.01701016  0.4638246      NaN    8.442707     17.76429
```

```
## 4 0.17032768 -0.1071349 6.815466   23.194739    127.69008
```

```r
diana.2 <- CC[, , "DIANA_Euclidean", "2", drop = FALSE]
hc.2 <- CC[, , "HC_Euclidean", "2", drop = FALSE]
gmm.2 <- CC[, , "GMM", "2", drop = FALSE]
hm.1 <- graph_heatmap(hc.2)
```




```r
hm.2 <- graph_heatmap(diana.2)
```




```
gmm.2 <- graph_heatmap(gmm.2)
```