Introduction to *CLUES* Package

Fang Chang
York University

Vincent Carey
Harvard Medical School

Weiliang Qiu
Harvard Medical School

Ruben H. Zamar
University of British Columbia

Ross Lazarus
Harvard Medical School

Xiaogang Wang
York University

November 14, 2009

1 Introduction

We introduce a novel R package *clues* which provides a clustering methodology with no prior information on number of clusters required. Shrinking procedure, partition procedure and determination of the optimal number of groups are three mainstreams of the algorithm. The functions in *clues* have the capability of locating optimal number of partitions according to the strength measures, either CH or Silhouette index. Additionally, in order to assess the performance of clustering methods, functions that are computing agreement indices (Rand index, Hubert and Arabie's adjusted Rand index, Morey and Agresti's adjusted Rand index, Fowlkes and Mallows index and Jaccard index) for any two partitions are also provided.

2 An Illustrative Application to Iris Data

In this section, for the purpose of illustrating the usage of *clues*, we borrow the Fisher/Anderson `iris` dataset included in base R. This data set is of dimension 150×5 with 4 variables being sepal length and width and petal length and width. The data present results for 3 balanced species in order, "setosa", "versicolor" and "virginica". First we invoke this package by inputting the following command at R Console.

```
> library("clues")
```

We take a glance of the data and examine its pair-wise scatter plot for the original data which is given in Figure 1.

```
> data("iris")
> head(iris)

  Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1          5.1         3.5          1.4         0.2  setosa
2          4.9         3.0          1.4         0.2  setosa
3          4.7         3.2          1.3         0.2  setosa
4          4.6         3.1          1.5         0.2  setosa
5          5.0         3.6          1.4         0.2  setosa
6          5.4         3.9          1.7         0.4  setosa

> dat.s <- as.matrix(iris[, -5])
> colnames(dat.s) <- c("SL", "SW", "PL", "PW")
```

There are two clear groups in the scatter plot, one group consists of "setosa", and the other is formed by "versicolor" and "virginica", and the data for these two species are sort of messing up.

2.1 Manipulating Data with `clues`

In this section, we illustrate the usage of the function `clues`. By applying this function on a targeted data set, we could obtain the final partition result. An advantage of this function that makes it stand out from the others is that it does not require the input of the number of clusters. For `iris` data, when we apply function `clues` directly with `strengthMethod`, which reflects the compactness of the clusters, being `sil`, we will get 2 clusters. The first 50 observations form one group and the rests form the second. Alternatively when `CH` is selected as `strengthMethod`, we will get 3 groups instead. Still the first 50 observations form the first group, group 2 and 3 are sort of mixing together. More details are provided as follows.

```
> res.sil <- clues(dat.s, strengthMethod = "sil", disMethod = "Euclidean")
> res.sil

Number of data points:
[1] 150

Number of variables:
[1] 4

Number of clusters:
[1] 2

Cluster sizes:
```

```
> pairs(dat.s)
```

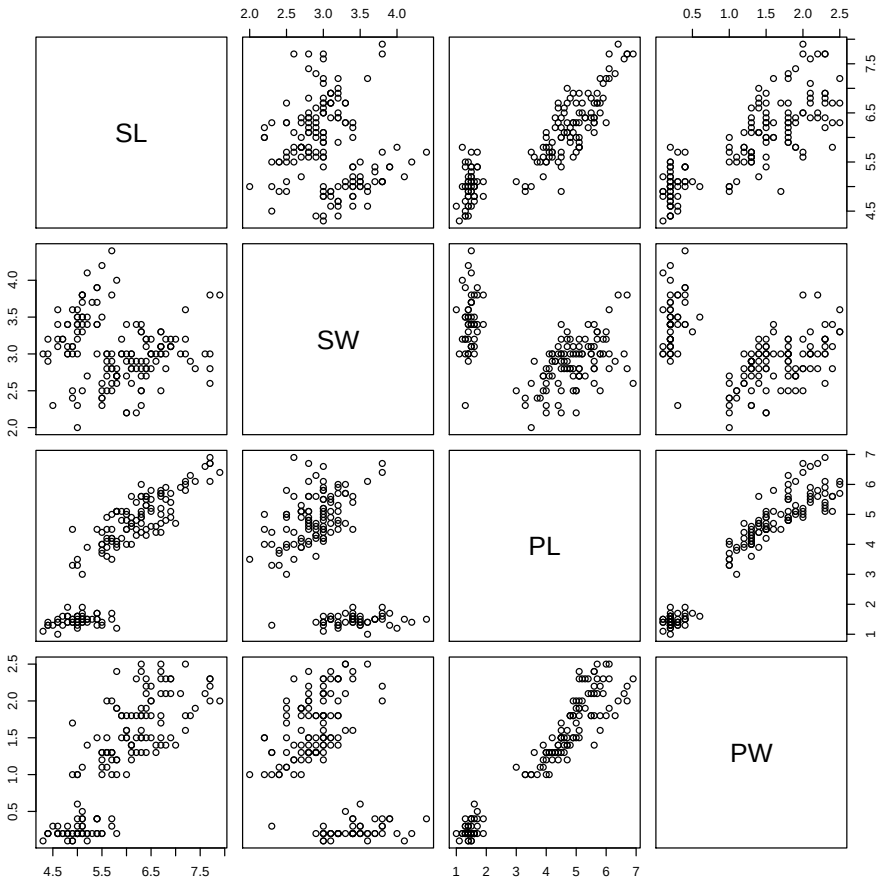

Figure 1: Pairwise scatter plots for original iris data

```
[1]  50 100

Strength method:
[1] "sil"

avg Silhouette:
[1] 0.6867351

dissimilarity measurement:
[1] "Euclidean"
```

```
Available components:
 [1] "K"              "size"           "mem"            "g"
 [5] "avg.s"          "s"              "K.vec"          "g.vec"
 [9] "myupdate"       "y.old1"         "y.old2"         "y"
[13] "strengthMethod" "disMethod"

> res.CH <- clues(dat.s, strengthMethod = "CH", disMethod = "Euclidean")
> res.CH

Number of data points:
[1] 150

Number of variables:
[1] 4

Number of clusters:
[1] 3

Cluster sizes:
[1] 50 65 35

Strength method:
[1] "CH"

CH:
[1] 556.1177

dissimilarity measurement:
[1] "Euclidean"


Available components:
 [1] "K"              "size"           "mem"            "g"
 [5] "CH"             "K.vec"          "g.vec"          "myupdate"
 [9] "y.old1"         "y.old2"         "y"              "strengthMethod"
[13] "disMethod"
```

The clustering result from `res.sil` gives two reasonable groups shown in Figure 2. The average silhouette index appears to be 0.6867351 which means that the partition is well done since it is not far from its upper limit 1. Since groups 2 and 3 are partially overlapped, this leads to 15 data points that belong to group 3 being dragged to group 2. The CH index appears to be 556.1177 indicating that between group variation is 556.1177 times of within group variation. Additionally, besides `Euclidean`, `1-corr` is

also available as a `disMethod`. However, for this type of data set, it turns out that Euclidean distance works much better than 1− correlation.

2.2 Visualizing Partition Result

Now that we have obtained the clustering result, *clues* provides functions that can be applied to visualize it. Figures 2 and 3 are obtained by function `plotClusters` and show graphical result with clusters distinguished by different plot symbols and colors. The partition shown in Figure 2 is obtained when `strengthMethod` is `sil`, while the partition exhibited in Figure 3 is obtained when `strengthMethod` is chosen to be `CH`.

```
> plotClusters(dat.s, res.sil$mem, plot.dim = 1:4)
```

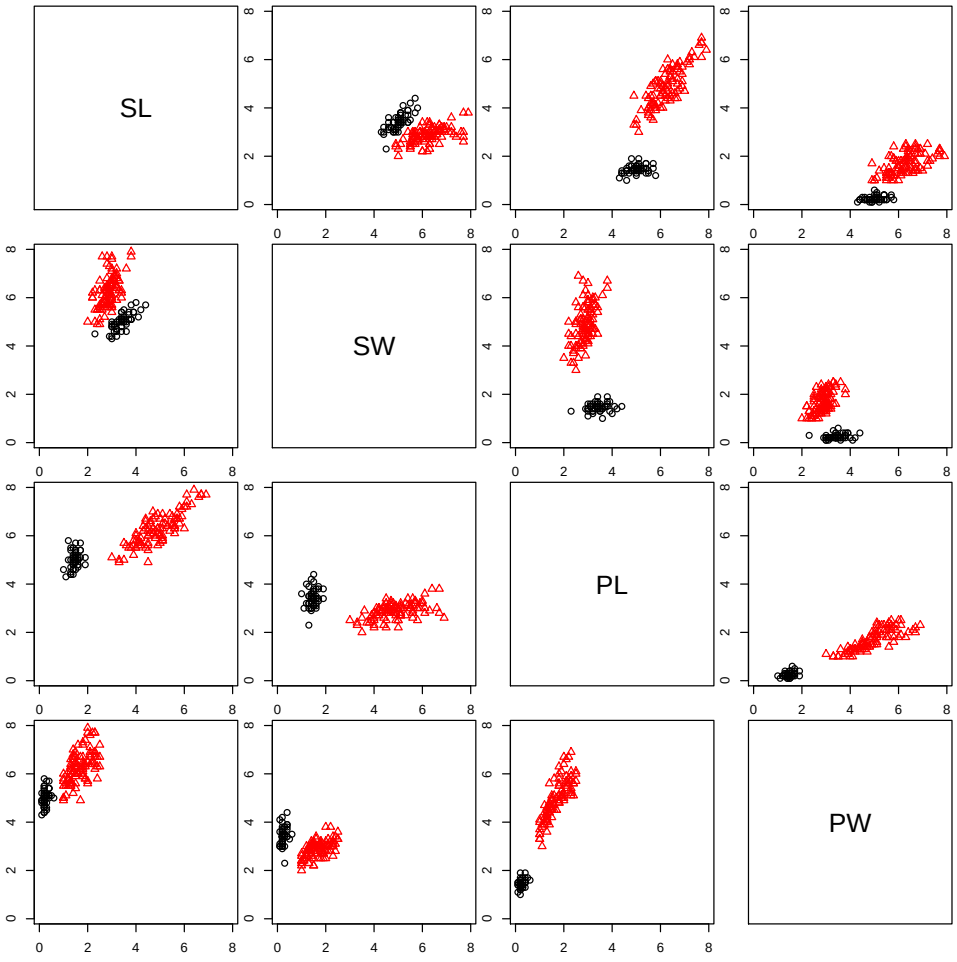

Figure 2: Scatter plots for iris data after clustering using Silhouette index

Also function `plotAvgCurves` provides us a way to plot the average trajectories for each existing cluster, shown in Figure 4 and Figure 5.

```
> plotClusters(dat.s, res.CH$mem, plot.dim = 1:4)
```

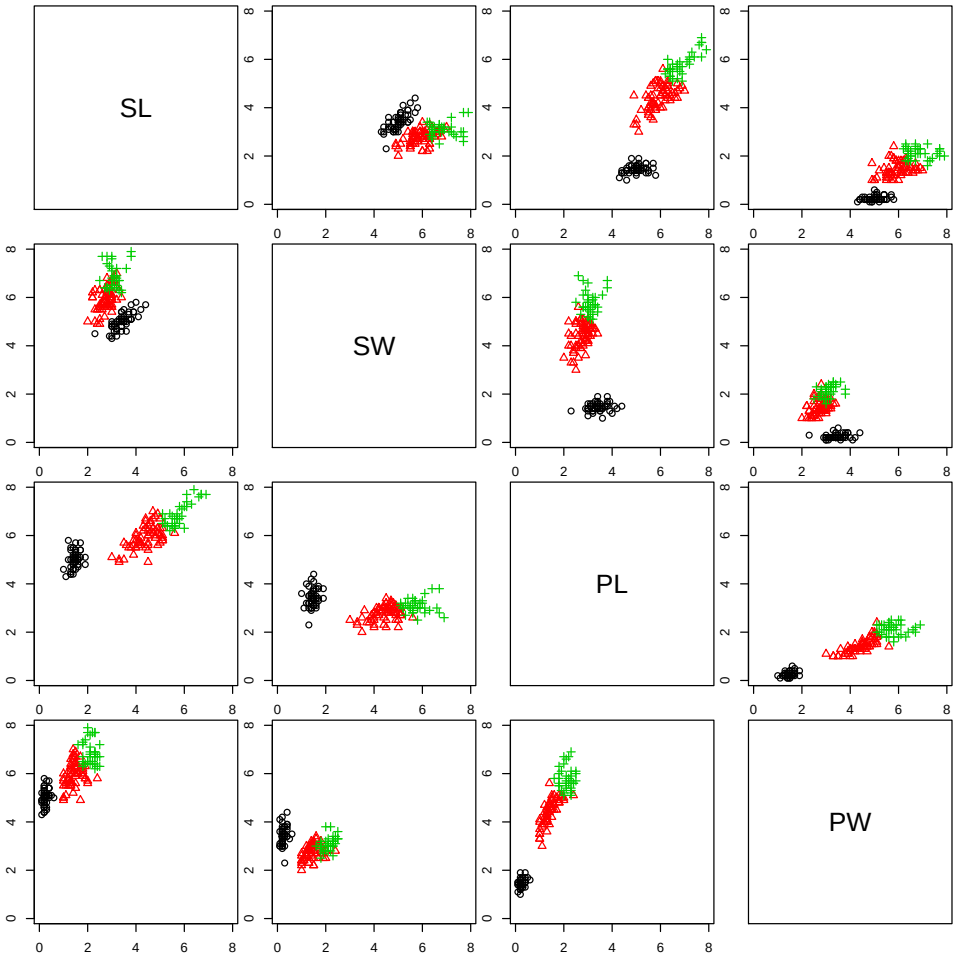

Figure 3: Scatter plots for iris data after clustering using CH index

2.3 Comparison Between Different Partition Methods

To view similarities among the partitions derived from different clustering methods, function `compClust`, which calculates mutual agreement indices for any pair of the methods considered, gives decision makers useful insight, especially in the case when the true membership is unknown. If different partitions are quite similar in terms of agreement indices, the true cluster structure is at least separated and the resulting clustering is fairly reliable. Numeric output of this function includes strength indices which consist of average Silhouette index and CH index, agreement indices.

We compare `clues` with `kmeans` borrowed from package *stats* and `pam` borrowed from package *cluster*. Due to the blurred boundary, the true clusters for `iris` can be treated either 2 or 3. We test the performances of clustering methods for both scenarios.

```
> library(cluster)
```

```
> plotAvgCurves(dat.s, res.sil$mem)
```

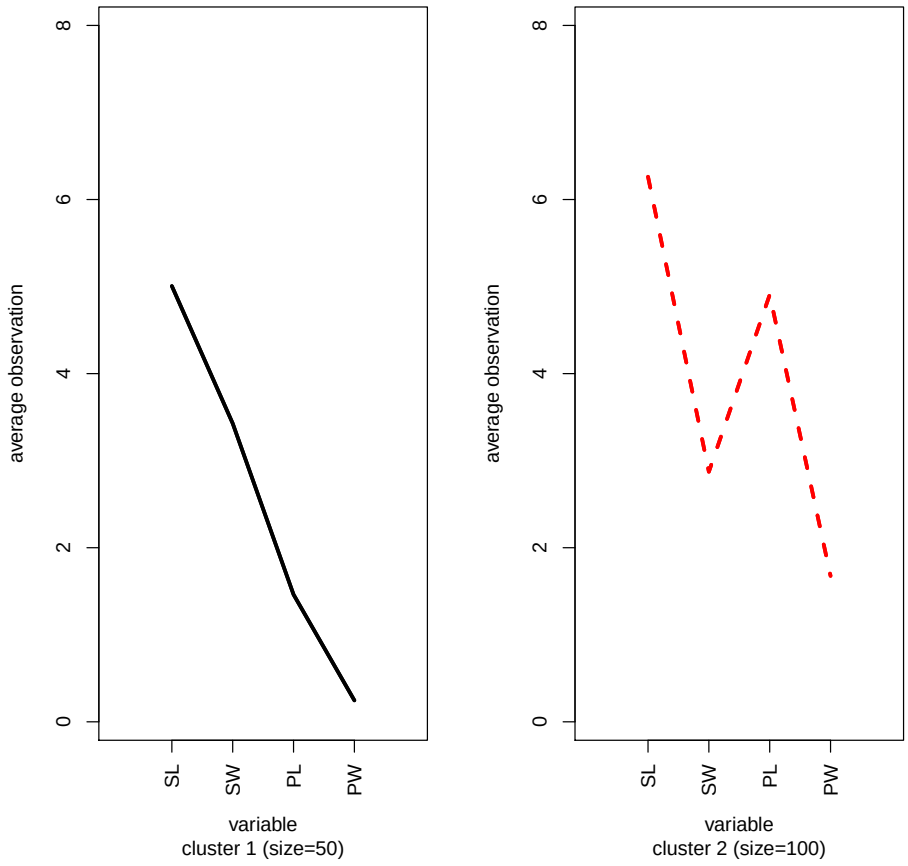

Figure 4: Average trajectory plots for iris data after clustering using Silhouette index

```
> library(stats)
> iris.mem <- rep(1, dim(iris)[1])
> iris.mem[iris$Species == "versicolor"] <- 2
> iris.mem[iris$Species == "virginica"] <- 3
> res.km <- kmeans(dat.s, 3, algorithm = "MacQueen")
> res.pam <- pam(dat.s, 3)
> memMat <- cbind(iris.mem, res.CH$mem, res.sil$mem, res.km$cluster,
+     res.pam$clustering)
> colnames(memMat) <- c("true", "clues.CH", "clues.sil", "km",
+     "pam")
> tt <- compClust(dat.s, memMat)
> print(sapply(tt, function(x) {
```

```
> plotAvgCurves(dat.s, res.CH$mem)
```

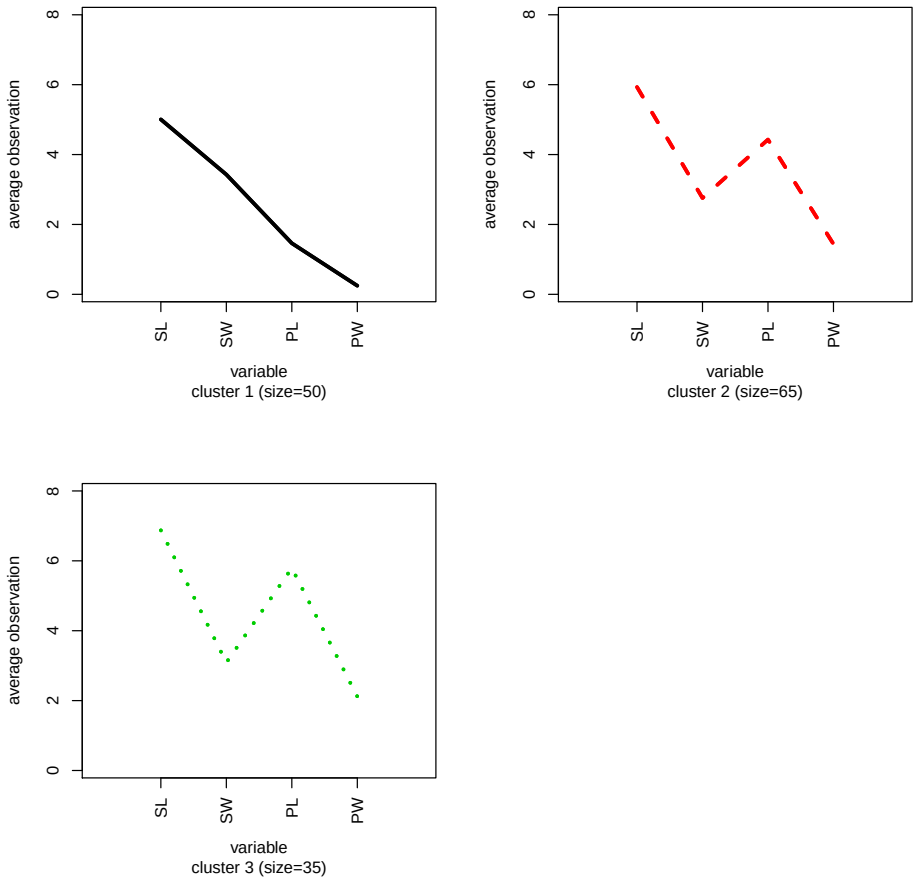

Figure 5: Average trajectory plots for iris data after clustering using CH index

```
+     round(x, 2)
+ }))

$avg.s
     true  clues.CH clues.sil        km       pam
     0.50      0.56      0.69      0.55      0.55

$CH
     true  clues.CH clues.sil        km       pam
   487.33    556.12    502.82    561.59    561.63

$Rand
          true clues.CH clues.sil   km  pam
```

```
true      1.00     0.89      0.78 0.87 0.88
clues.CH  0.89     1.00      0.80 0.97 0.97
clues.sil 0.78     0.80      1.00 0.79 0.79
km        0.87     0.97      0.79 1.00 0.99
pam       0.88     0.97      0.79 0.99 1.00

$HA
          true clues.CH clues.sil   km  pam
true      1.00     0.75      0.57 0.72 0.73
clues.CH  0.75     1.00      0.61 0.92 0.94
clues.sil 0.57     0.61      1.00 0.59 0.59
km        0.72     0.92      0.59 1.00 0.98
pam       0.73     0.94      0.59 0.98 1.00

$MA
          true clues.CH clues.sil   km  pam
true      1.00     0.75      0.57 0.72 0.73
clues.CH  0.75     1.00      0.61 0.92 0.94
clues.sil 0.57     0.61      1.00 0.59 0.60
km        0.72     0.92      0.59 1.00 0.98
pam       0.73     0.94      0.60 0.98 1.00

$FM
          true clues.CH clues.sil   km  pam
true      1.00     0.83      0.77 0.81 0.82
clues.CH  0.83     1.00      0.79 0.95 0.96
clues.sil 0.77     0.79      1.00 0.78 0.79
km        0.81     0.95      0.78 1.00 0.99
pam       0.82     0.96      0.79 0.99 1.00

$Jaccard
          true clues.CH clues.sil   km  pam
true      1.00     0.71      0.60 0.68 0.70
clues.CH  0.71     1.00      0.63 0.90 0.93
clues.sil 0.60     0.63      1.00 0.61 0.62
km        0.68     0.90      0.61 1.00 0.97
pam       0.70     0.93      0.62 0.97 1.00
```

It is easy to see `clues` with `CH` being the strength index has outstanding performance since it matches the true partition best in terms of any of these agreement indices given that the true number of clusters is 3 where each species forms its own group.

```
> iris.mem1 <- rep(2, dim(iris)[1])
```

```
> iris.mem1[iris$Species == "setosa"] <- 1
> res.km1 <- kmeans(dat.s, 2, algorithm = "MacQueen")
> res.pam1 <- pam(dat.s, 2)
> memMat1 <- cbind(iris.mem1, res.CH$mem, res.sil$mem, res.km1$cluster,
+     res.pam1$clustering)
> colnames(memMat1) <- c("true", "clues.CH", "clues.sil", "km",
+     "pam")
> tt <- compClust(dat.s, memMat1)
> print(sapply(tt, function(x) {
+     round(x, 2)
+ }))
$avg.s
     true  clues.CH clues.sil        km       pam
     0.69      0.56      0.69      0.68      0.69

$CH
     true  clues.CH clues.sil        km       pam
   502.82    556.12    502.82    513.92    509.70

$Rand
          true clues.CH clues.sil   km  pam
true      1.00     0.80      1.00 0.96 0.99
clues.CH  0.80     1.00      0.80 0.78 0.79
clues.sil 1.00     0.80      1.00 0.96 0.99
km        0.96     0.78      0.96 1.00 0.97
pam       0.99     0.79      0.99 0.97 1.00

$HA
          true clues.CH clues.sil   km  pam
true      1.00     0.61      1.00 0.92 0.97
clues.CH  0.61     1.00      0.61 0.56 0.59
clues.sil 1.00     0.61      1.00 0.92 0.97
km        0.92     0.56      0.92 1.00 0.95
pam       0.97     0.59      0.97 0.95 1.00

$MA
          true clues.CH clues.sil   km  pam
true      1.00     0.61      1.00 0.92 0.97
clues.CH  0.61     1.00      0.61 0.57 0.59
clues.sil 1.00     0.61      1.00 0.92 0.97
km        0.92     0.57      0.92 1.00 0.95
pam       0.97     0.59      0.97 0.95 1.00
```

```
$FM
          true clues.CH clues.sil   km  pam
true      1.00     0.79      1.00 0.96 0.99
clues.CH  0.79     1.00      0.79 0.77 0.78
clues.sil 1.00     0.79      1.00 0.96 0.99
km        0.96     0.77      0.96 1.00 0.98
pam       0.99     0.78      0.99 0.98 1.00

$Jaccard
          true clues.CH clues.sil   km  pam
true      1.00     0.63      1.00 0.93 0.98
clues.CH  0.63     1.00      0.63 0.60 0.62
clues.sil 1.00     0.63      1.00 0.93 0.98
km        0.93     0.60      0.93 1.00 0.95
pam       0.98     0.62      0.98 0.95 1.00
```

When we assume the true number of groups to be 2 where "setosa" forms the first group and the rests are left as the second, method `clues` with strength measure being `sil` gives a perfect match to the true partition. Methods `kmeans` and `pam` give slightly different results.

2.4 Data Sharpening

The function `shrinking` serves as a data sharpener.

```
> shrinkres <- shrinking(dat.s, K = 60, disMethod = "Euclidean")
> dimnames(shrinkres) <- dimnames(dat.s)
```

After data sharpening, pairwise scatter plot for the sharpened data is presented in Figure 6.

In the pairwise scatter plot for sharpened iris data set, three distinct data points are clearly seen which indicates there are 3 distinct groups for the original data when we fix `K` to be 60.

3 Discussion

In this vignette, we provide a rough guidance for a recently developed clustering package *clues*. It is superior to commonly used partition algorithms by getting rid of necessity of prior information about the number of clusters. Some functions, such as `get_CH`, `get_Silhouette` and `CompClust` can be used as tools for evaluating new clustering methods in simulation study.

```
> pairs(shrinkres)
```

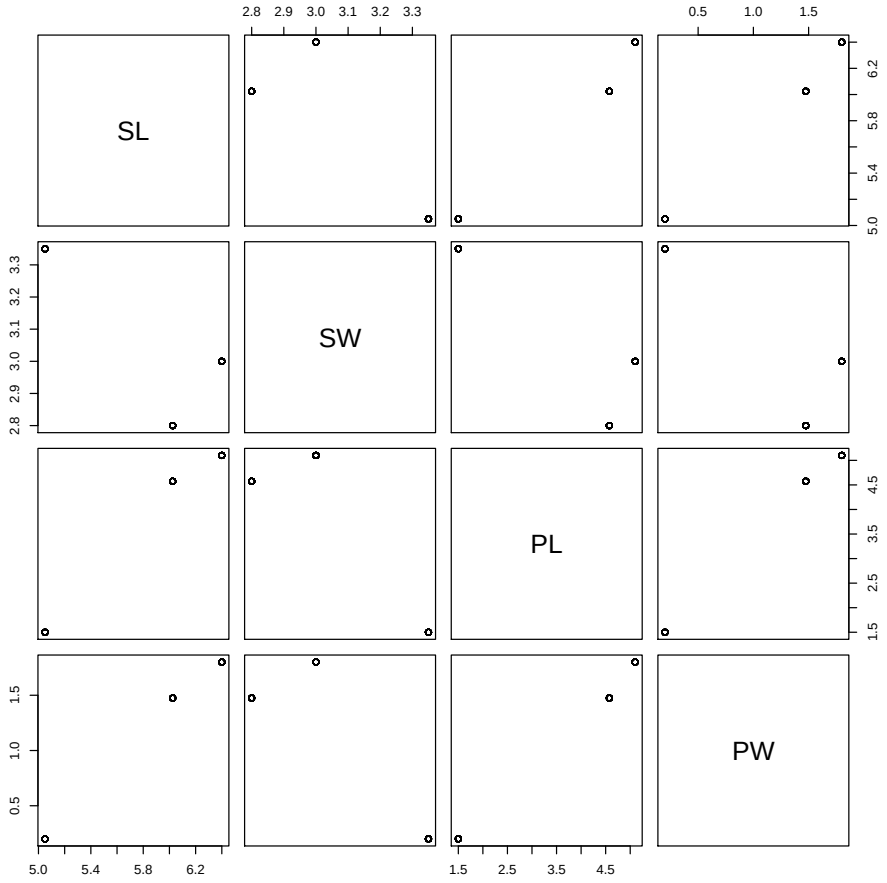

Figure 6: Scatter plots for sharpened iris data

4 References

Fisher, R.A., 1936. The Use of Multiple Measurements in Taxonomic Problems, *Annals of Eugenics*, 7(2), 179-188.

Wang, X.G., Qiu, W.L., and Zamar, R.H., 2007. CLUES: A Non-parametric Clustering Method Based on Local Shrinking, *Computational Statistics & Data Analysis*, 52(1), 286-298.