Program *K*-means
User's Guide

Pierre Legendre
Département de sciences biologiques
Université de Montréal
C.P. 6128, succursale Centre-ville
Montréal, Québec H3C 3J7, Canada

July 2001

Internet: **Pierre.Legendre@umontreal.ca**

What does program *K*-means do?

K-means is a least-squares partitioning method allowing users to divide a collection of objects into *K* groups. The theory is presented in all textbooks of numerical classification methods, including section 8.8 of Legendre and Legendre (1998). The program implements a simple alternating least-squares algorithm, the same one as used in the SAS FASTCLUS procedure. The algorithm iterates between two simple steps:

• Compute cluster centroids and use them as new cluster seeds.

• Assign each object to the nearest seed.

Among the 100 or so algorithms that have been described for *K*-means partitioning, this one is known to be the fastest; it also has good convergence properties.

In the course of the iterations, the program tries to minimise the sum, over all groups, of the squared within-group residuals, which are the distances of the objects to the respective group centroids. Convergence is reached when the objective function (i.e., the residual sum-of-squares) cannot be lowered any more. The groups obtained are such that they are geometrically as compact as possible around their respective centroids.

Since *K*-means partitioning is one of the so-called "NP-hard problems", no algorithm can guarantee that the absolute minimum of the objective function has been reached; any given run of *K*-means partitioning may end up in what is called a "local minimum" of the objective function. A strategy, described below, helps users find a value of the objective function which is likely to represent the overall minimum.

Users must be aware of the fact that *K*-means tends to partition the data into groups of about equal sizes and spherical shape, each group being assumed to have a unit variance-covariance matrix.

Strategies towards an optimal solution

The number of groups, *K*, into which the observations are partitioned, is determined by the user of the program. When *K* is known from theory, there is no problem. In most cases, however, the true number of groups in the data set is unknown and the user must search through several values of *K*. Programs K-means2 allows users to search through different values of *K* in a cascade, starting with k1 groups and ending with k2 groups, with k1 ≥ k2. In the cascade from a larger to the next smaller number of groups, the two closest groups are identified and fused. Then the two-step alternating least-squares algorithm described above is run until convergence, reallocating objects to the groups.

For each number of groups (*K*), the Calinski-Harabasz (1974) pseudo-*F*-statistic (C-H) is computed:

$$\text{C-H} = [R^2/(K-1)]/[(1-R^2)/(n-K)] \text{ where } R^2 = (\text{SST} - \text{SSE})/\text{SST}.$$

SST is the total sum of squared distances to the overall centroid and SSE is the sum of squared distances of the objects to their group's own centroids.

One is interested to find the number of groups, *K*, for which the Calinski-Harabasz criterion is maximum; this corresponds to the most compact set of groups. In a simulation study involving 30 stopping rules for cluster analysis, Milligan and Cooper (1985) found that the Calinski-Harabasz criterion was the one recovering the correct number of groups the most often.

To start the procedure, the observations (objects) have to be initially assigned to the groups. This can be done in three ways, described by the following menu:

```
Initial assignment of objects to groups:
(1) Equal groups along the series
(2) At random
(3) From file of group assignments
(4) Read estimated centroids from file
```

Option 1 is intended for time series or spatial transects displaying autocorrelation.

With option 2, the strategy used in the program for finding the optimal number of groups is to repeat the *K*-means partitioning cascade using different random assignments of the objects to the groups. In the example provided below, the partitioning cascade has been repeated 20 times with different initial assignments of objects to the groups. The program retains, and writes to the output file, the solution for each number of groups *K* for which the Calinski-Harabasz criterion is the highest; this is equivalent to minimizing the residual sum-of-squares statistic over all run results having that number of groups.

If option 3 is chosen, a file has to be provided giving a list of numbers which represent the groups to which the objects are assigned. The file is structured as follows:

Line 1: how many objects are members of each of the *k* groups.

Line 2: list the numbers of the objects that are members of group 1.

Line 3: list the numbers of the objects that are members of group 2.

[**...**]

Line *k*+1: list the numbers of the objects that are members of group *k*.

The list of members of any one group may span on several successive lines. The members of any one group may be in any order in the list. A new group must start on a new line. For example, 10 objects may be assigned to $k = 3$ groups using the following file:

```
2 5 3
4 7
1 8 9 5 6
2 10 3
```

The program checks that the values in the first line of the file sum to *n*, which is the total number of objects in the study. A *Premature end-of-file* message is sent if the file does not contain *n* values. The program also checks for objects unassigned to groups, sending a message that some objects are likely to have been assigned to multiple groups by mistake.

When some of the groups are assumed to be very different from the others in size or shape, a meaningful partition is more likely to be obtained by providing initial estimates for the group centroids. Using option 4, estimated centroids can be read from a text file in which the rows are the centroids and the columns are the variables. If the variables are to be transformed by the program (species transformations,

standardization, or ranging), make sure that the centroids are in the same units as the transformed data. In any case, the program checks that the centroids are within the range of the data and stops if they are not.

Input files

The input data file is an ASCII text file without row or column identifiers. Three options are available:

```
Structure of the input file:
(1) Rows are objects and columns are variables
(2) Columns are objects and rows are variables
(3) Sonar data: position and QTC variables
```

Option (3) was added because sonar data files may contain position variables before the variables to be used for partitioning. The first set of variables, i.e. the position variables, are read and discarded since they are of no use for the *K*-means partitioning phase of the analysis. The use of *K*-means partitioning for acoustic seabed classification and seafloor mapping is described in Legendre et al. (2002) and Legendre (2003).

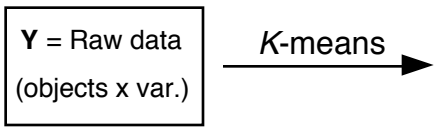
Data transformations

Different types of transformations can be applied to the data prior to *K*-means partitioning. Transformations have to be used parsimoniously. The main strategies are described in Fig. 1. The theoretical aspects to consider follow from the fact that *K*-means is a least-squares partitioning procedure that creates groups in Euclidean space. So, the data should be transformed only if the distance relationships among the objects makes more sense in Euclidean space after the transformation.

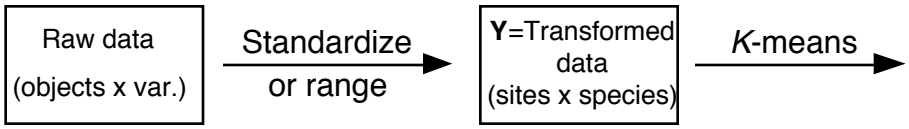
(a) For physical variables that are dimensionally homogeneous, or from which the physical dimensions have already been removed (this is the case of ranged spectral decompositions of sonar backscatter), no transformation should be used.

Table of physical variables

(a) Variables are dimensionally homogeneous

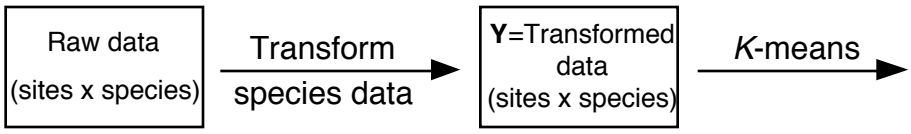


(b) Variables are not dimensionally homogeneous



Site-by-species data table

(c) Species presence/absence or abundance data: Transformation approach



(d) Species presence/absence or abundance data: Distance-based approach

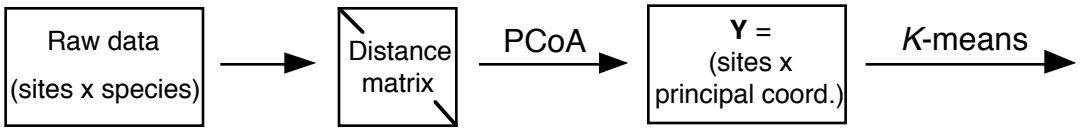


Figure 1. Data transformations have to be used only in specific cases, outlined above.

(b) For physical variables that are not dimensionally homogeneous, the relative positions of the objects in Euclidean space depends to a large extent on the arbitrary choice of physical units. Standardization can and should be applied to the variables to remove this arbitrariness.

Standardization (z-score transformation):	$y(i,j)'=(y(i,j) - \text{Mean}(y)) / \text{StDev}(y)$
Ranging for variables with true zero:	$y' = y(i,j)/yMax$
Ranging for variables with arbitrary zero:	$y' = (y(i,j) - yMin)/(yMax - yMin)$

Milligan and Cooper (1988) report simulation results showing that, for clustering purposes, if a transformation is needed, the ranging transformations (above) are in general far superior to standardization.

(c) For species abundance data tables, a series of simple transformations have recently been proposed by Legendre and Gallagher (submitted). These transformations have been incorporated into the *K*-means program. The menu proposed by the program is the following:

```
Transform species abundance data, in order to base
the analysis on a different distance measure?
(see Legendre & Gallagher, manuscript)

(0) No transformation (Euclidean distance preserved)
(1) Chord distance
(2) Chi-square metric
(3) Chi-square distance
(4) Distance between species profiles
(5) Hellinger distance
```

These transformations are such that Euclidean distances computed on the transformed data are equal to the chosen distances if they were computed on the original, untransformed data. Since *K*-means preserves Euclidean distances among the objects, the end result of these transformations is to effectively preserve the selected distances during partitioning.

(d) For species abundance data tables, transformations are available only for some of the distances that are appropriate; if one wants to preserve some other distance, for instance the Steinhaus/Bray-Curtis distance, another strategy has to be used. One must select a resemblance function (e.g. Jaccard similarity for presence-absence data, or Steinhaus/Bray-Curtis distance for species abundances), compute the said similarity or distance matrix, and use a program of principal coordinate analysis (PCoA) to obtain a series of coordinates in Euclidean space, prior to *K*-means partitioning.

A program to carry out principal coordinate analysis with correction for negative eigenvalues is available on the WWWeb site <http://www.fas.umontreal.ca/biol/legendre/>. Select "Computer programs and datasets"from the left-hand column of the introduction page, then "DistPCoA". The output of this program can be used as input for *K*-means partitioning.

Weighting the variables

Should all variables receive the same weight in the assessment of the distance among objects and, thus, in *K*-means partitioning? Probably not, but in most situations the weights to be attributed to variables are unknown.

• When weights are already known from theory, they can be written to a file, which is then read and used by the *K*-means program.

• Weights that are optimal for *K*-means partitioning can be computed beforehand. A program for optimal variable weighting is available on the WWWeb site <http://www.fas.umontreal.ca/biol/legendre/>. Select "Computer programs and datasets"from the left-hand column of the introduction page, then "Optimal Variable Weighting (OVW)". This program can find optimal weights for the variables in view of ultrametric clustering (i.e., standard agglomerative clustering methods), additive tree clustering (i.e., phylogenetic trees), and *K*-means partitioning. The optimal weights computed by this program can be incorporated into the *K*-means run.

The prompt of the program allowing to read in a vector of weights is the following:

```
Is there an input file of variable weights?
(0) No,  (1) Yes
```

Output file

The output file contains the results of the run. A first table gives the value of the Calinski-Harabasz pseudo-*F*-tatistic for each number of groups *K*. This is followed by the actual best solutions for each number of groups K:

• The residual sum-of-squares statistic (SSE), the Calinski-Harabasz pseudo-*F*-statistic (C-H), and the group membership of each group (i.e., the number of objects in each group).

• The vector of group assignments for all objects. This vector can be copied-and-pasted into the data base. For geographic data with known geographic positions, the vector can be used to map the groups obtained by *K*-means partitioning.

Disclaimer

This program is provided without any explicit or implicit warranty of correct functioning. It has been developed as part of a university-based research program. If, however, you should encounter problems with this program, the author will be happy to help solve them. Researchers may use this program for scientific purposes, but the source code remains the property of Pierre Legendre. Users of the program may refer to the present user's manual as follows:

Legendre, P. 2001. Program *K*-means user's guide. Département de sciences biologiques, Université de Montréal. 11 pages.

Technical notes

The program is available in a variety of forms:

• FORTRAN source code for Macintosh (file K-means2.f), which can be compiled using a FORTRAN compiler. The user may modify the Parameter statement at the beginning of the program, which fixes the size of the largest data matrix that can be analysed analysed (nmax = maximum number of observations, pmax = maximum number of variables, kmax = maximum number of groups). [Source code not distributed at the moment.]

• FORTRAN source code for DOS or Windows (file K-MEANS2.FOR), which can be compiled using a FORTRAN compiler. The user may modify the Parameter statement at the beginning of the program, which fixes the size of the largest data matrix that can be analysed (nmax = maximum number of observations, pmax = maximum number of variables, kmax = maximum number of groups). [Source code not distributed at the moment.]

• Compiled version for PowerPC processors for Macintosh (file K-MEANS2/PPC). The maximum size of the data matrix is 10000 objects, 250 variables and 50 groups. The program requires 24.5 Mb RAM for running.

• Compiled version for IBM compatible PC (file K-MEANS2.EXE). The maximum size of the data matrix is 10000 objects, 250 variables and 30 groups. The program has been compiled for 32-bit operating systems (i.e. Windows95 or WindowsNT) and requires ??? Mb RAM for most situations. It is preferable to have 24 Mb RAM available for calculations on very large matrices.

References

Calinski, T. and J. Harabasz. 1974. A dendrite method for cluster analysis. *Communications in Statistics* 3: 1-27.

Legendre, P. 2003. Reply to the comment by Preston and Kirlin on "Acoustic seabed classification: improved statistical method". *Canadian Journal of Fisheries and Aquatic Sciences* 60: 1301-1305.

Legendre, P., K. E. Ellingsen, E. Bjørnbom and P. Casgrain. 2002. Acoustic seabed classification: improved statistical method. *Canadian Journal of Fisheries and Aquatic Sciences* 59: 1085-1089.

Legendre, P. and E. D. Gallagher. 2001. Ecologically meaningful transformations for ordination of species data. *Oecologia* 129: 271-280.

Legendre, P. and L. Legendre. 1998. *Numerical ecology. 2nd English edition*. Elsevier Science BV, Amsterdam.

Milligan, G. W. and M. C. Cooper. 1985. An examination of procedures for determining the number of clusters in a data set. *Psychometrika* 50: 159-179.

Milligan, G. W. and M. C. Cooper. 1988. A study of standardization of variables in cluster analysis. *Journal of Classification* 5: 181-204.

Appendix: Test runs

A data table containing 495 rows (objects) and 163 variables (columns) was run as an example. The output in the dialogue window was the following.

```
K-means partitioning program

Pierre Legendre
Departement de sciences biologiques
Universite de Montreal.
            © Pierre Legendre, 1999

Name of input file?
Input file: Rose4/Sonar, 495x163

Structure of the input file:
(1) Rows are objects and columns are variables
(2) Columns are objects and rows are variables
(3) Sonar data: position and QTC variables
1

How many objects (rows)?
495

How many variables (columns)?
163

Print how many lines of data? (None: type 0)
0

Transform species abundance data, in order to base
the analysis on a different distance measure?
(see Legendre & Gallagher, manuscript)

(0) No transformation (Euclidean distance preserved)
(1) Chord distance
(2) Chi-square metric
(3) Chi-square distance
(4) Distance between species profiles
(5) Hellinger distance
0

Is there an input file of variable weights?
(0) No,  (1) Yes
0

Name of output file
Output file: Rose Bay.out
```

```
Initial assignment of objects to groups:
(1) Equal groups along the series
(2) At random
(3) From file of group assignments
(4) Read estimated centroids from file
2

How many random starts would you like?
20

Run K-means from how many (k1) to how many groups (k2)? (k1 >= k2)
10 2

Standardize or range the variables?
(Use ONLY if the variables are not dimensionally homogeneous)
(0) Do not standardize or range
(1) Standardize: y(i,j)'=(y(i,j)-Mean(y))/StDev(y)
(2) Range: y' =  y(i,j)/yMax
(3) Range: y' = (y(i,j)-yMin)/(yMax-yMin)
0

Type a small integer (1-20) to initialize the random number generator:
5

Currently working on random start no.           1
Currently working on random start no.           2
Currently working on random start no.           3
Currently working on random start no.           4
Currently working on random start no.           5
Currently working on random start no.           6
Currently working on random start no.           7
Currently working on random start no.           8
Currently working on random start no.           9
Currently working on random start no.          10
Currently working on random start no.          11
Currently working on random start no.          12
Currently working on random start no.          13
Currently working on random start no.          14
Currently working on random start no.          15
Currently working on random start no.          16
Currently working on random start no.          17
Currently working on random start no.          18
Currently working on random start no.          19
Currently working on random start no.          20

Real time spent:    858.03 seconds

End of the program.
```

Output file

The maximum value of the Calinski-Harabasz (1974) statistic (C-H) is found at $K = 4$, indicating that the within-group density is maximum for 4 groups. This is the most interesting partition of the 495 objects. Details about group assignment are given for all values of K. Only two of these tables ($K = 10$ and $K = 4$) are reproduced here to save space.

```
K-means partitioning program

 Pierre Legendre
 Departement de sciences biologiques
 Universite de Montreal.
             © Pierre Legendre, 1999

 Best result for each number of groups (K)
 out of  20 random starts for file              Rose4/Sonar, 495x163

 No. groups (K)    C-H pseudo-F-statistic
      2                 1039.39661
      3                 1143.07434
      4                 1445.60046
      5                 1320.63013
      6                 1279.82825
      7                 1247.02515
      8                 1203.20239
      9                 1208.37476
     10                 1167.59668

 C-H = [R-square/(K-1)]/[(1-R-square)/(n-K)] is the Calinski-Harabasz (1974)
 pseudo-F-statistic. Use the partition for which C-H is maximum.

 Detailed results of group membership:

K =10 groups: SSE =     12.93778        C-H =  1167.59668 ( 42 iterations) (Random start  7)
Group membership:   40   43   45   45   22   99   59   33   41   68

K =10: Vector of group assignments for the   495 objects:

   2    8    9    2    2    2    2    2    2    2    2    2    2    8    2    2    8    2    2    5
   5    5    9    5    9    9    9    7    2    7    9    9    9    9    9    9    7    9    9    9
   7    7    9    7    9    7    4    7    9    7    9    9    7    9    7    4    7    7    9    7
   7    4    4    7    7    7    4    7    4    4    4    4    7    7    7    7    4    4    7    4
   4    4    4    4    7    7    7    4    7    7    4    7    7    7    7    4    7    7    4    7
   7    7    7    7    7    4    7    7    7    7    9    7    4    4    4    4    4    4    4    7
   4    4    4    4    4    4    4    4    4    4    4    7    4    5    7    4    4    4    7    7
   4    7    7    9    7    4    7    7    7    7    9    9    7    2    9    9    9    9    9    2
   9    2    9    9    2    2    5    9    2    5    5    8    9    9    2    2    2    2    9    2
   2    2    9    2    2    2    5    2    2    9    5    5    2    2    2    2    5    8    5    5
   5    8    5    2    5    5    8    5    8    5    5    2    8    8    8    2    8    8    8    8
   8    5    8    8    8    8    8    8    8    8    3    3    8    8    8    8    8    3    3    8
   3    8    8    3    3    8    3    3    3    3    3    9    3    3    3    3    3    3    3    3
  10    3    3    3    3    3   10    3   10   10    3   10    3   10   10   10    3    3   10    3
  10   10   10   10    3   10   10   10   10   10   10    3   10   10   10   10   10   10   10   10
  10   10    3    3   10   10   10   10    1   10    1    1   10    1    1   10   10    1   10   10
  10    1   10    1    1    1   10    1    1    1    1    1    1    1    6    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1   10   10   10   10    3    3
  10    9    3    2    3    3    3   10   10   10    3    3    3    1   10   10    3   10    2    1
  10   10   10   10   10    1   10    1   10   10   10   10   10   10   10   10   10    6    6    6
   6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6
   6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6
   6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6
   6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6    6
   6    6    6    6    6    6    6    6    6    6    6    6    6    6    6

(same for k = 9 to 5 groups)
```

```
K = 4 groups: SSE =     29.82503        C-H =   1445.60046 (   6 iterations) (Random start 12)
Group membership:  116  124  129  126

K = 4: Vector of group assignments for the   495 objects:

   3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
   3    3    3    3    3    3    1    1    3    1    3    3    3    3    3    3    1    1    3    3
   1    1    1    1    3    1    1    1    3    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    3    1    1    1    1    1    1
   1    1    1    1    1    1    1    1    1    1    1    1    1    3    3    3    3    3    3    3
   3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
   3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
   3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3    3
   3    3    3    3    3    3    3    3    3    3    4    3    3    3    3    3    3    4    4    3
   4    3    3    4    4    3    4    4    4    4    4    1    4    4    4    4    4    4    4    4
   4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4
   4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4    4
   4    4    4    4    4    4    4    4    4    4    4    2    4    4    2    4    4    4    4    4
   4    4    4    2    2    2    4    2    2    2    2    2    4    2    2    2    2    2    2    2
   2    2    2    2    4    4    2    4    2    2    2    4    4    4    4    4    4    4    4    4
   4    3    4    3    3    4    4    4    4    4    4    4    4    4    4    4    4    4    3    4
   4    4    4    4    4    4    4    2    4    4    4    4    4    4    4    4    4    2    2    2
   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
   2    2    2    2    2    2    2    2    2    2    2    2    2    2    2
```

(same for k = 3 and 2 groups)