Computers & Geosciences Vol. 13, No. 1, pp. 13–35, 1987
Printed in Great Britain.
0098-3004/87 $3.00 + 0.00

CONISS: A FORTRAN 77 PROGRAM FOR STRATIGRAPHICALLY CONSTRAINED CLUSTER ANALYSIS BY THE METHOD OF INCREMENTAL SUM OF SQUARES*

ERIC C. GRIMM
Limnological Research Center, University of Minnesota, 220 Pillsbury Hall, 310 Pillsbury Dr. S.E., Minneapolis, MN 55455, U.S.A.

(*Received* 13 *January* 1986; *accepted* 29 *May* 1986)

Abstract—Stratigraphically constrained cluster analysis is a multivariate method for quantitative definition of stratigraphic zones. As opposed to ordinary, unconstrained cluster analysis, only stratigraphically adjacent clusters are considered for merging. The method of incremental sum of squares has been used widely for unconstrained analyses and has proved particularly satisfactory for pollen frequency data. CONISS is a FORTRAN 77 program for stratigraphically constrained cluster analysis by this method. Several data transformations lead to different implicit dissimilarity coefficients. As an option, the program also will perform an unconstrained analysis, which can be useful for comparison with the constrained analysis.

Key Words: Biostratigraphy, Cluster analysis, Stratigraphically constrained cluster analysis, Stratigraphic zones.

INTRODUCTION

Biostratigraphic sequences may be divided into zones to facilitate description and correlation. Zones can be defined by the presence or absence of fossil taxa, by the abundance of selected taxa, or by the entire fossil assemblage. Assemblage zones are based on the representation of all taxa present, or on taxa of a certain type (Hedberg, 1976). Assemblage zones can be defined for a single stratigraphic section or for a large number of sections (American Commission on Stratigraphic Nomenclature, 1970). Quaternary palynologists typically define local pollen assemblage-zones for individual sections. Many investigators simply have demarcated zones by visual inspection of a stratigraphic diagram. Others have used quantitative methods to delineate zones, which are objective, although as might be expected different methods typically give somewhat different results. Nevertheless, they are useful for analysis and presentation of data and provide consistent criteria for defining zones (Birks and Gordon, 1985; Birks, 1986).

Most methods of numerical zonation are cluster analyses, constrained so that clusters contain only stratigraphically adjacent samples. Several techniques have been used, including divisive and agglomerative hierarchical methods and nonhierarchical methods (Birks and Gordon, 1985; Gordon, 1980, 1981).

One of the earliest programs for constrained cluster analysis was OPTAGG by E. J. Cushing, a FORTRAN adaptation of L. Orloci's ALGOL program of the same name, which was unconstrained (E. J. Cushing, 1986, pers. comm.). OPTAGG performs a group-average cluster analysis, which is hierarchical agglomerative, with either Euclidian distance or Orloci's (1967) standard or chord distance as the dissimilarity coefficient. (OPTAGG does not perform the optimal agglomeration described by Orloci, 1967.) OPTAGG has had several applications, including unconstrained cluster analysis of Cretaceous and Tertiary palynomorphs (Oltz, 1969, 1971) for characterizing zones and constrained analysis for zonation of Quaternary fossil-pollen sequences (e.g. Mehringer, Arno, and Peterson, 1977).

The most widely used methods of constrained cluster analysis for palynological data have been those of Gordon and Birks (1972). They presented one agglomerative method (CONSLINK), which uses a single linkage criterion for merging clusters, and two divisive methods, SPLITINF, which uses an information statistic, and SPLITLSQ, which uses a sum of squares measure of within-cluster variability. Gordon and Birks used two different dissimilarity coefficients with CONSLINK, the Manhattan or city-block metric and the Canberra metric. The former has been used more frequently. An excellent example of the application of these methods is Birks and Berglund (1979). Birks and Gordon (1985) recently published a listing of the program ZONATION, which carrys out all of these zonation procedures.

One criterion for numerical zonation has been to minimize within-zone dispersion (sum of squares) or mean within-zone dispersion (variance). Gill (1970) and Gordon and Birks (1972) published hierarchical divisive methods to approximate this criterion. Hawkins and Merriam (1973, 1974) and Hawkins (1976) presented nonhierarchical methods that use a dynam-

*Contribution No. 323, Limnological Research Center, University of Minnesota.

ic programming algorithm for determining the overall minimum within-zone dispersion for g zones for univariate or multivariate data. Birks and Gordon (1985) fully describe the dynamic algorithm, and their program ZONATION will carry it out.

These methods assume homogeneity of the fossil assemblage to be the primary criterion of zone definition. Some stratigraphic sequences, however, may be transitional in nature. The BARRIER algorithm of Gordon (1973) attempts to separate transitional levels or zones from more homogeneous ones. The algorithm minimizes within-cluster variation around "local centroids" and places "barriers" between adjacent samples, the heights of which indicate the dissimilarities between adjacent groups based on a sum of squares criterion. High barriers indicate transitional samples. The algorithm is iterative and depends on an arbitrary value for the sum of barrier heights. The method also can be approached as a form of single-linkage cluster analysis with a particular dissimilarity coefficient, and a dendrogram can be constructed. Examples of its use with fossil-pollen stratigraphies are Gordon and Birks (1974), Birks (1981), and Jacobson and Grimm (1986).

An advantage of hierarchical methods is that relationships among zones are examined easily. Agglomerative methods are more satisfactory for zonation, because clusters are built up locally. Stratigraphically constrained hierarchical divisive methods, although computationally feasible, are dependent on the exact and entire stratigraphic sequence. The position of the first split depends on the composition of all the samples above and below it, and the positions of splits may change if the sequence is truncated at the top or bottom, especially if any stratigraphic revertence occurs. OPTAGG and CONSLINK are both hierarchical and agglomerative. The single linkage criterion of CONSLINK can lead to severe chaining, especially in pollen sequences where analysts usually place samples closer together in zones of rapid change.

Recent studies have indicated that with frequency pollen data the incremental sum of squares method of cluster analysis performs satisfactorily (e.g. Birks, Webb, and Berti, 1975). The method is agglomerative and hierarchical. This paper describes the program CONISS, which carries out stratigraphically constrained cluster analysis by the method of incremental sum of squares. As an option the program will perform an unconstrained analysis. Although intended for stratigraphic data, the method is useful for other types of linearly ordered or transect data, for example vegetation zones along a natural gradient. Examples are pollen frequency data, but any type of appropriate fossil or even nonfossil stratigraphic data may be zoned.

THE METHOD AND ALGORITHM

Ward (1963) first published the incremental sum of squares method, but several others discovered it independently, and it is known by several other names, including Ward's method, minimum variance, sum of squares, error sum of squares, and optimal agglomeration (Ward, 1963; Orloci, 1967, 1978; Wishart, 1969; Anderson, 1971; Anderberg, 1973; Gordon, 1981; Pielou, 1984). The name incremental sum of squares (Burr, 1970; Clifford and Stephenson, 1975) most accurately describes the technique, which is one of the most widely used methods of cluster analysis.

A general goal of sum of squares cluster analysis is to minimize the total within-cluster dispersion for g groups around g centroids. The incremental sum of squares method approximates the overall optimal g groups. It is an agglomerative algorithm that places clusters in a hierarchy. Truly optimal clusters for different values of g would not be hierarchical necessarily (Gordon, 1981; Birks and Gordon, 1985). The optimality of the hierarchical zones can be determined with the dynamic algorithm in Birks and Gordon's (1985) program ZONATION. In many situations clusters formed by the incremental sum of squares method will be those with the minimum total within-cluster dispersion. In situations in which the clusters do not have the minimum total dispersion, the hierarchical nature of the zonation may be more desirable, especially if the sample to sample variability differs along the section. Zones may change in heterogeneity, and the g most geologically reasonable zones may not be the g zones with minimum total within-cluster dispersion. The hierarchical cluster analysis allows for the local determination of zone boundaries along the section. The name "minimum variance" for the incremental sum of squares method is misleading somewhat, as the hierarchical agglomeration does not give necessarily the clusters with minimum variance possible.

Within-cluster dispersion or sum of squares for the pth cluster is defined as:

$$D_p = \sum_{i=1}^{n_p} \sum_{j=1}^{m} (x_{pij} - \bar{x}_{pj})^2$$

where n_p = number of samples in cluster p, m = number of variables, x_{pij} = value of jth variable of sample i in cluster p, and $\bar{x}_{pj}$ = mean value of variable j in cluster p. Mean dispersion or variance of cluster p is D_p/n_p. Total dispersion for g clusters is:

$$D = \sum_{p=1}^{g} D_p.$$

If clusters p and q are merged together to form cluster pq, the increase in dispersion I_{pq} is:

$$I_{pq} = D_{pq} - D_p - D_q.$$

At each stage of the clustering the two clusters p and q that give the least increase in dispersion I_{pq} are merged.

These equations are sufficient for a computer algorithm to perform the cluster analysis, but a more

efficient algorithm exists (Wishart, 1969; Burr, 1970; Anderson, 1971). First, a dissimilarity matrix of squared Euclidian distances between all samples is generated. The data matrix then is no longer required. At each successive stage the dissimilarity matrix is searched, and the pair of clusters p and q with the smallest dissimilarity value (d_{pq}) are merged. The increase in dispersion $I_{pq} = \frac{1}{2}d_{pq}$, and total dispersion is incremented by this amount after each stage. The number of clusters decreases by one, and all dissimilarities in the matrix with either cluster p or q must be updated. The update equation for the new dissimilarity value $d_{r(pq)}$ between cluster r and new cluster pq formed by the merger of p and q is:

$$d_{r(pq)} = \frac{(n_r + n_p)d_{rp} + (n_r + n_q)d_{rq} - n_r d_{pq}}{n_r + n_p + n_q}.$$

In the usual unconstrained analysis, the entire dissimilarity matrix is searched at each stage for the minimum value. In the constrained analysis, only stratigraphically adjacent clusters are considered.

DISSIMILARITY COEFFICIENTS

The described algorithm operates on a dissimilarity matrix of squared Euclidian distances. Although in practice any dissimilarity matrix can be used, the geometric properties of non-Euclidian dissimilarity coefficients have not been investigated. However, transformations of the data can be made before calculation of the matrix of squared Euclidian distances. Transformations can weight variables or samples, and certain transformations produce dissimilarity coefficients other than simple Euclidian distance. The program will perform three transformations: standardization of variables to mean zero and unit standard deviation, normalization of sample vectors to unit length, and square-root transformation. Standardization equally weights the variables, and the associated dissimilarity coefficient may be referred to as standardized Euclidian distance. The implicit dissimilarity coefficient resulting from normalization of sample vectors is Orloci's (1967, 1978) standardized or chord distance (termed cosine theta distance by Prentice (1980); the well-known cosine theta similarity coefficient of Imbrie and Purdy (1962) is the cosine of the angle subtending the chord). A square-root transformation of frequency data produces the chord distance of Edwards and Cavalli-Sforza (1964). Both normalization of sample vectors and square-root transformation place samples on the surface of a unit hypersphere, thus making the Euclidian distance between transformed samples a chord. Orloci's chord distance alters the lengths but not the directions of the sample vectors. Edwards and Cavalli-Sforza's chord distance weights variables and alters the orientations of the sample vectors. For frequency data, this coefficient up-weights rare variables relative to abundant ones. It has proved particularly satisfactory for pollen percentage data (Overpeck, Webb, and Prentice, 1985). (See Prentice (1980) and Overpeck, Webb, and Prentice (1985) for further discussion of dissimilarity coefficients for relative and absolute pollen data.)

DENDROGRAMS

A dendrogram illustrates the hierarchical relationships of the clusters defined by the analysis. Various criteria have been used to define the heights of nodes connecting clusters in dendrograms for incremental sum of squares cluster analysis, including (1) increase in dispersion at each stage (Gordon, 1981), (2) total dispersion at each stage (Ward, 1983; Anderberg, 1971), (3) within-cluster dispersion of individual clusters (Pielou, 1984), and (4) mean within-cluster dispersion of individual clusters (Orloci, 1967; Birks, Webb, and Berti, 1975). Each of these scales provides somewhat different information about the analysis. Dendrograms with scales (1) and (4) are subject to reversals. Dendrograms (3) and (4) provide information about individual clusters. Within-cluster dispersion (3) is highly dependent on cluster size. Total within-cluster dispersion (2) is not subject to reversals and illustrates the progressive formation of the clusters—the node of each merger is above the nodes of all previous mergers. In general, dendrograms (2) and (4) provide the most useful information.

Clusters may be defined by simply cutting the dendrogram at a given height. This practice may be useful for defining zones, depending on the investigator's purpose, but it can be arbitrary, as the same zones may not be duplicable by a straight cut on the different dendrograms. A straight cut across dendrogram (4) will produce clusters all having mean within-cluster dispersion less than a certain value. If homogeneity (little within-zone variability or dispersion) is a criterion for zone definition (e.g. Gordon and Birks, 1972), this strategy might be the best. In a stratigraphic sequence, however, some clearly recognizable zones may be less homogeneous than others, either because of sample to sample variability or strong gradation from bottom to top. A straight cut across dendrogram (2) will in many situations produce zones similar to those the investigator would have selected by inspection. This cut marks off all clusters formed up to a certain stage in the agglomeration process. A good strategy might be to delimit zones by a straight line across dendrogram (2), and then, if desired, divide certain zones further into subzones (Hedberg, 1976, p. 49) that have interpretive value. Alternatively, different parts of the stratigraphy may be considered separately. The geologically most reasonable dendrogram clusters may require different cutoffs at different stratigraphic levels. The hierarchical nature of the cluster analysis leads to qualitative decisions concerning criteria for zone definition, but in any instance the analysis provides a quantitative characterization of the zones defined.

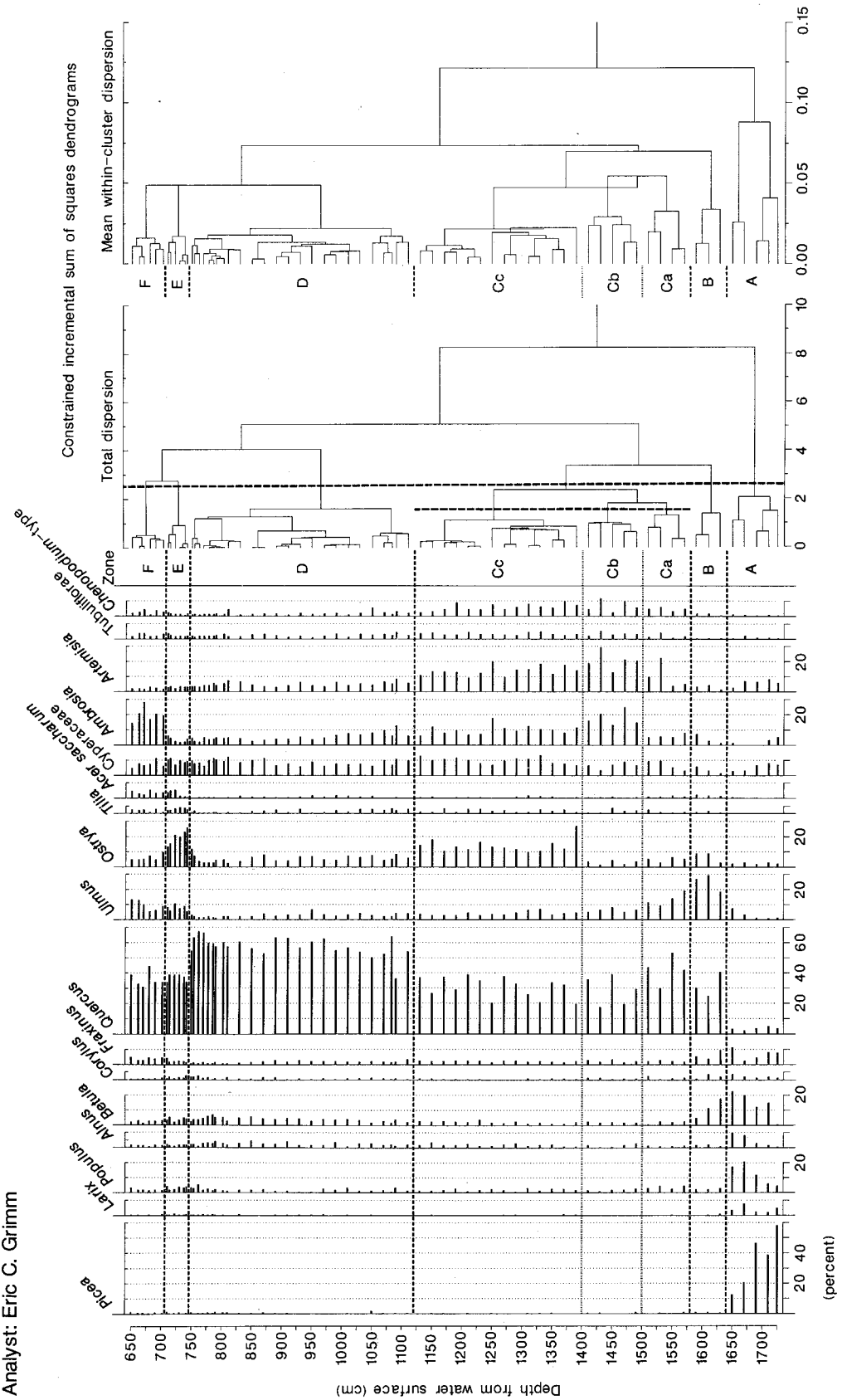

Figure 1. Pollen-percentage diagram and dendrograms from constrained incremental sum of squares cluster analysis. Depth is from water surface; sediment surface is at 650 cm. Percentages are based on sum of 17 types shown. Zones and subzones are defined by dashed lines cutting dendrogram of total dispersion.

EXAMPLE

The following example is of a fossil-pollen sequence consisting of 68 samples from an 11 m core spanning 12,000 radiocarbon yrs from Wolsfeld Lake, Minnesota (Fig. 1) (see Grimm (1983) for site and other details). Seventeen upland pollen-types were used in the analysis. Eliminated were all types < 3% at every level, aquatics, Gramineae, *Salix*, and *Pinus*. Elimination of rare taxa has little effect on the analysis. Aquatics were eliminated because the interest was upland vegetation. Gramineae and *Salix* were eliminated because of highly local representation, probably from aquatic grasses and willows growing around the lake. *Pinus* exceeds 3% but probably was all transported from long distances; it is a prolific pollen producer and probably never occurred around the site. A square-root transformation was made of the data, thus the implicit dissimilarity coefficient is Edwards and Cavalli-Sforza's chord distance. Six zones are defined by a straight cut of the dendrogram of total dispersion (Fig. 1). The zones simply are given letters for identification. In addition, Zone C is divided into three subzones by a lower cut of the dendrogram.

The dendrogram of mean within-cluster dispersion shows the relative homogeneity of the zones. Basal Zone A, the late-glacial "spruce-zone," has high within-cluster dispersion, because of the constant decline of *Picea* and its replacement by deciduous trees. Zones B and Ca also are relatively heterogeneous because of their transitional nature. The heterogeneity of Zone Cb is more the result of haphazard fluctuations in pollen types. Zones Cc through A all have similar degrees of relative homogeneity. Zone Ca is characterized by the decline of deciduous tree pollen and the increase in herb pollen, Zone Cb by the maximum of herb pollen, and Zone Cc by somewhat decreased herb pollen and high *Ostrya*. Zone C represents a time when prairie existed in the region. The zone as a whole is heterogeneous, but the subzones, particularly Cc, are less so. The mean dispersions of Subzones Ca and Cb are high, but the addition of the many samples from the relatively homogeneous Subzone Cc lowers the mean dispersion for the entire zone, hence the reversal in the dendrogram.

The unconstrained analysis (Fig. 2) allows examination of the distinctiveness of zones and of the relationships among zones. The zones and subzones defined all fall out as distinct clusters. Only two samples (36 and 58) cluster with samples primarily from another zone. Sample 36 (1090 cm) near the bottom of Zone D has lower *Quercus* and higher prairie pollen-types compared to the rest of Zone D, but clusters with samples from Subzone Cc, rather than with the underlying Subzone Ca. Sample 58 (1530 cm) in Subzone Ca has *Quercus* and *Artemisia* values resembling Subzone Cb and clusters with samples from that subzone. The remaining samples from Subzone Ca form a cluster that merges with Zone B, rather than with Subzone Cb as in the constrained analysis. Thus the inclusion or exclusion of sample 58 from Zone Ca determines whether this subzone clusters with Zone C or Zone B. In any instance, Subzone Ca is transitional in nature.

The unconstrained analysis can indicate revertence —the reoccurrence of a pollen assemblage higher in the sequence. The cluster composed of samples from Zone B and Subzone Ca merges with Zone E, from near the top of the section. The analysis shows the close resemblance of these early and late Holocene pollen-assemblages that indicate mixed deciduous forest. Although similar, however, they are distinct in the analysis; the main difference being the relative importance of *Ulmus* and *Ostrya*. Complete revertence would be demonstrated by mixing of samples in one cluster from two stratigraphically nonadjacent zones. Substantial mixing of samples from adjacent zones in the unconstrained clusters might suggest that the zone boundary is unwarranted.

The unconstrained cluster analysis also can be used for data from more than one stratigraphic section to delimit regional assemblage zones (Birks and Gordon, 1985). Clusters containing samples from different sections would define regional assemblage zones. A cluster containing samples from a single section, may indicate a zone of only local occurrence. Birks and Gordon (1985, p. 102–105) provide an example of such an analysis, using incremental sum of squares cluster analysis.

PROGRAM SPECIFICATIONS

The program is written in FORTRAN 77 and is in complete accordance with the ANSI standard. It has been compiled and run successfully on a CRAY-1 computer and with two different compilers on a CDC CYBER 845. It will run on much smaller computers. The dissimilarity matrix is held in a 1-dimensional array to reduce core requirements, but as a result the code is difficult to follow in places. Subroutines for plotting dendrograms are included, but they can be eliminated if the plotting package is not available.

Output

(1) The first section of output indicates the progress of the program up to the actual clustering.

(2) The second section is the merge data, which are printed out in the order of the agglomerative clustering process (see Anderberg, 1973). For n samples, there are $n - 1$ cluster mergers or stages. Initially each cluster is assigned a number corresponding to its order in the input data. The first merger, and probably the next few, will involve only single-sample clusters. After two clusters are merged, the new cluster is assigned the lower number of the two original clusters. Thus, if clusters 10 and 19 are merged, the new cluster now is known as cluster 10. A dendrogram can be constructed easily by following the table of

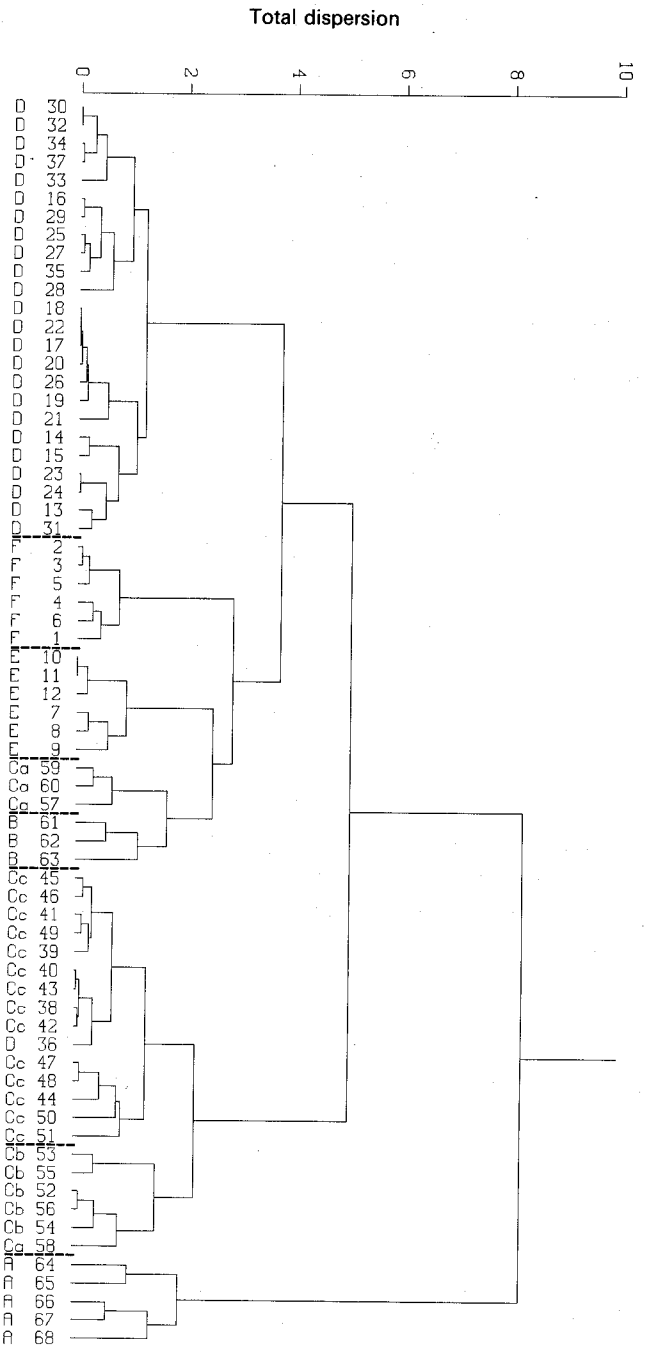

Figure 2. Dendrogram from unconstrained incremental sum of squares cluster analysis of 68 samples from Wolsfeld Lake. Samples are identified by zone (letter) and stratigraphic position (numbered from top to bottom). Dashed lines separate clusters corresponding to zones in pollen diagram.

merge data. Printed for each merger and the new cluster are the increase in dispersion, total dispersion, within-cluster dispersion, and mean within-cluster dispersion.

(3) The third section prints out sample numbers. The first column is the sequence order of the samples, the numbers identifying the samples in the merger table. For constrained analyses the second column is sample depths. For unconstrained analyses the second column is sample numbers as assigned by the investigator, which may be depths for stratigraphic data or arbitrary numbers for nonstratigraphic data.

(4) For unconstrained analyses the order of samples on the dendrogram, as established by subroutine DORDER, is printed. This order is not unique, for the dendrogram is similar to a mobile, and branches can be swung around. However, the order is useful for an initial plot of the dendrogram. For constrained

analyses, of course, the order of samples is stratigraphic.

Dendrogram plots

The program contains three subroutines (DDGRAM, TREE, DEPCOL) for plotting dendrograms with any of the four possible scales for either the constrained or unconstrained analysis. For constrained analyses a column indicating the depth of each level is plotted along side the dendrogram, and the depth of every tenth sample is plotted. For unconstrained analyses the depth column is eliminated, and the number of each sample is plotted. The order of samples is that established by subroutine DORDER.

The plotting routines use the DISSPLA graphics package, version 9.0 (Integrated Software Systems Corporation, 1981). They assume a plotting device with a continuous paper roll (e.g. CALCOMP plotter). Installation and device dependent subroutines and arguments are indicated in the program listing. If DISSPLA is not available, these subroutines can be deleted from the program, and no other program modifications are necessary, but lines 94–96 and 254–260 can be deleted if desired. For other plotting packages, the algorithm in subroutine TREE can be adapted for plotting dendrograms.

Files

The data are read from a file connected to unit 1 and program control cards from a file connected to unit 5. These files must be preconnected with systems commands. The cluster results are written to unit 6. As an option, the data after any transformations may be written to unit 7.

Control cards

A set of instructions read from unit 5 controls action of the program. Some cards are required and others are optional. Each instruction begins in column 1. The following instructions can be placed in any order.

NUMBER OF SAMPLES = *n* [required; *n* is an integer value]

NUMBER OF VARIABLES = *m* [required; *m* is an integer value]

DATA ARE PROPORTIONS [optional; default; this or one of the following two instructions may be selected]

DATA ARE PERCENTAGES [optional]

DATA ARE COUNTS [optional]

CONVERT DATA TO PROPORTIONS [optional; converts either counts or percentages to proportions]

SQUARE ROOT TRANSFORMATION [optional; this or one of the following instructions may be selected; default is no transformation]

STANDARDIZE VARIABLES [optional; standardizes variables to mean 0, standard deviation 1]

NORMALIZE SAMPLES [optional; normalizes sample vectors to unit length]

CONSTRAINED [optional; default; this or the following instruction may be selected]

UNCONSTRAINED [optional]

INPUT FORMAT = format [optional, default = (F5.0,5X,10F7.2,/(10X,10F7.2))]

OUTPUT FORMAT = format [optional, for data written to unit 7, default = (F8.2,2X,5E14.6,/(10X,5E14.6))]

WRITE DATA [optional; causes data to be written to unit 7; useful if a data transformation has been made]

The following instruction, if present, must be placed after those listed.

PLOT DENDROGRAM [optional; plots dendrogram; this card alone will plot a default dendrogram]

The following instructions alter features of the dendrogram. They must follow the PLOT DENDROGRAM card and can occur more than once; their order will affect the plot.

HEIGHT = *x* [optional; *x* = the height of the dendrogram in inches; default = 4.0]

WIDTH = *x* [optional; *x* = the width of the dendrogram in inches; default = 10.0]

CHARACTER HEIGHT = *x* [optional; *x* = height of lettering in inches; default = 0.10]

SCALE = *n* [optional; *n* = an integer value from 1–4; determines the scale of the dendrogram; default = 2 (see next)]

Every time a SCALE card is encountered, a dendrogram is plotted with the current values of HEIGHT, WIDTH, and CHARACTER HEIGHT. These values may be changed between SCALE cards. If no SCALE card is present, a dendrogram with the default scale is plotted. The values of SCALE are as follows:

SCALE = 1 [Increase in dispersion]

SCALE = 2 [Total dispersion]

SCALE = 3 [Within-cluster dispersion]

SCALE = 4 [Mean within-cluster dispersion]

Core requirements and array dimensions

The amount of memory required depends on the dimensions of the data and distance arrays, on whether double precision is used (see next), and on whether the graphics subroutines are used. The program is dimensioned for a maximum of 150 samples and 25 variables. These dimensions are changed easily by changing the values of MAXV, MAXL, and MXDM (see lines 29–31 in program) in the PARAMETER statements in the main program and in subprograms CCLUS, UCLUS, UPDATE, and D2.

To reduce amount of memory required, some arrays have been overlaid. To maximize execution time, however, the two largest arrays, which hold the data and distance matrices, are separate. In some situations overlaying these arrays can reduce substantially core requirements. The necessary program modifications involve writing out the distance matrix as it is generated, then reading it back in. Core requirements will be reduced only if MAXV is > 12. The following modifications will overlay the data and distance matrices.

Replace lines 35–36 with:

```
      DIMENSION X(MAXV,MAXL)
      EQUIVALENCE (D,X)
      DIMENSION Z(MAXL,12),N(MAXL,12)
      EQUIVALENCE (Z,N)
```

Insert after line 211:

```
      OPEN (10,STATUS='SCRATCH',FORM=
     'UNFORMATTED')
```

Replace line 220 with:

```
            WRITE (10) DSQD
```

Insert after line 222:

```
            REWIND 10
            DO 175 I=1,ID
              READ (10) D(I)
        175 CONTINUE
```

Precision

After each iteration the dissimilarity matrix is updated, and rounding error accumulates. Depending on the number of samples, on the degree of sample similarity, and on the size of numeric storage units of a particular computer, single precision arithmetic may not be sufficient. No simple rule can determine when double precision arithmetic is necessary. However, experience with the same data set on an IBM 370 (32-bit words) and CRAY-1 (64-bit words) suggests that on computers with 32-bit numeric storage units, double precision arithmetic likely will be necessary for > 100 samples. Double precision doubles the size of the distance matrix, and on smaller computers the above modifications for overlaying the distance and data matrices also may become necessary. The following modifications to the program will invoke double precision arithmetic where necessary.

Insert after line 34:

```
      DOUBLE PRECISION D,DSHORT,DSQD
```

Insert after lines 273 and 367:

```
      DOUBLE PRECISION D,DSHORT,DE,
     E,UPDATE
```

Replace lines 317–319 and 414–416 with:

```
      ESS(NAMP) = ESS(NAMP)+ESS(NAMQ)
                  +REAL(DE)
      ES(ITER,1) = REAL(DE)
      ES(ITER,2) = REAL(E)
```

Replace line 465 with:

```
      DOUBLE PRECISION FUNCTION UPDATE(R)
```

Insert after line 471:

```
      DOUBLE PRECISION D,DSHORT,D2
```

Replace line 480 with:

```
      DOUBLE PRECISION FUNCTION D2(I,J)
```

Insert after line 485:

```
      DOUBLE PRECISION D
```

Acknowledgments—I express my appreciation to E. J. Cushing and H. E. Wright, Jr., who critically read the manuscript. I received financial support from NSF grant BSR-8311473 and a grant from the University of Minnesota Computer Center.

REFERENCES

American Commission on Stratigraphic Nomenclature, 1970, Code of stratigraphic nomenclature: Am. Assoc. Petroleum Geologists, Tulsa, Oklahoma, 22 p.

Anderberg, M. R., 1973, Cluster analysis for applications: Academic Press, New York, 359 p.

Anderson, A. J. B., 1971, Numeric examination of multivariate soil samples: Jour. Math. Geology, v. 3, no. 1, p. 1–14.

Birks, H. J. B., 1981, Late Wisconsin vegetational and climatic history at Kylen Lake, northeastern Minnesota: Quaternary Res., v. 16, no. 3, p. 322–355.

Birks, H. J. B., 1986, Numerical zonation, comparison and correlation of Quaternary pollen-stratigraphical data, *in* Berglund, B. E., ed., Handbook of Holocene palaeoecology and palaeohydrology: John Wiley & Sons, Chichester, p. 743–774.

Birks, H. J. B., and Berglund, B. E., 1979, Holocene pollen stratigraphy of southern Sweden: a reappraisal using numerical methods: Boreas, v. 8, no. 3, p. 257–279.

Birks, H. J. B., and Gordon, A. D., 1985, Numerical methods in Quaternary pollen analysis: Academic Press, Orlando, 317 p.

Birks, H. J. B., Webb, T., III, and Berti, A. A., 1975, Numerical analysis of pollen samples from central Canada: a comparison of methods: Rev. Palaeobot. Palynol., v. 20, no. 3, p. 133–169.

Burr, E. J., 1970, Cluster sorting with mixed character types II. Fusion strategies: Aust. Comput. Jour., v. 2, no. 3, p. 98–103.

Clifford, H. T., and Stephenson, W., 1975, An introduction to numerical classification: Academic Press, New York, 229 p.

Edwards, A. W. F., and Cavalli-Sforza, L. L., 1964, Reconstruction of evolutionary trees, *in* Heywood, V. H., and McNeill, J., eds., Phenetic and phylogenetic classification: Systematic Assoc. Publ. 6, p. 67–76.

Gill, D., 1970, Application of a statistical zonation method to reservoir evaluation and digitized-log analysis: Am. Assoc. Petroleum Geologists Bull., v. 54, no. 5, p. 719–729.

Gordon, A.D., 1973, Classification in the presence of constraints: Biometrics, v. 29, no. 4, p. 821–827.

Gordon, A. D., 1980, Methods of constrained classification, *in* Tomassone, R., ed., Analyse de donnees et informatique, Fontainebleau, du 19 au 30 mars 1979: Institut National de Recherche en Informatique et en Automatique, Le Chesney, France, p. 161–171.

Gordon, A. D., 1981, Classification methods for the exploratory analysis of multivariate data: Chapman and Hall, London, 193 p.

Gordon, A. D., and Birks, H. J. B., 1972, Numerical methods in Quaternary palaeoecology I. Zonation of pollen diagrams: New Phytol., v. 71, no. 5, p. 961–979.

Gordon, A. D., and Birks, H. J. B., 1974, Numerical methods in Quaternary palaeoecology II. Comparison of pollen diagrams: New Phytol., v. 73, no. 1, p. 221–249.

Grimm, E. C., 1983, Chronology and dynamics of vegetation change in the prairie-woodland region of southern Minnesota, U.S.A.: New Phytol., v. 93, no. 2, p. 311–350.

Hawkins, D. M., 1976, FORTRAN IV program to segment multivariate sequences of data: Computers & Geosciences, v. 1, no. 4, p. 339–351.

Hawkins, D. M., and Merriam, D. F., 1973, Optimal zonation of digitized sequential data: Jour. Math. Geology, v. 5, no. 4, p. 389–395.

Hawkins, D. M., and Merriam, D. F., 1974, Zonation of multivariate sequences of digitized geologic data: Jour. Math. Geology, v. 6, no. 3, p. 263–269.

Hedberg, H. D., ed., 1976, International stratigraphic guide: John Wiley & Sons, New York, 200 p.

Imbrie, J., and Purdy E. G., 1962, Classification of modern Bahamian carbonate sediments: Am. Assoc. Petroleum Geologists Mem. 1, p. 253–272.

Integrated Software Systems Corporation, 1981, DISSPLA user's manual version 9.0: 4186 Sorrento Valley Blvd, San Diego, California.

Jacobson, G. L., Jr., and Grimm, E. C., 1986, A numerical analysis of Holocene forest and prairie vegetation in central Minnesota: Ecology, v. 67, no. 4, p. 958–966.

Mehringer, P. J., Jr., Arno, S. F., and Peterson, K. L., 1977, Postglacial history of Lost Trail Pass Bog, Bitteroot Mountains, Montana: Arct. Alp. Res., v. 9, no. 4, p. 345–368.

Oltz, D. F., Jr., 1969, Numerical analysis of palynological data from Cretaceous and Early Tertiary sediments in east central Montana: Palaeontographica, pt. B, v. 128, no. 3–6, p. 90–166.

Oltz, D. F., Jr., 1971, Cluster analyses of Late Cretaceous–Early Tertiary pollen and spore data: Micropaleontology, v. 17, no. 2, p. 221–232.

Orloci, L., 1967, An agglomerative method for classification of plant communities: Jour. Ecology, v. 55, no. 1, p. 193–206.

Orloci, L., 1978, Multivariate analysis in vegetation research (2nd ed.): Dr. W. Junk b.v., Publishers, The Hague, 451 p.

Overpeck, J. T., Webb, T., III, and Prentice, I. C., 1985, Quantitative interpretation of fossil pollen spectra: dissimilarity coefficients and the method of modern analogs: Quaternary Res., v. 23, no. 1, p. 87–108.

Pielou, E. C., 1984, The interpretation of ecological data: a primer on classification and ordination: John Wiley & Sons, New York, 263 p.

Prentice, I. C., 1980, Multidimensional scaling as a research tool in Quaternary palynology: a review of theory and methods: Rev. Palaeobot. Palynol., v. 31, no. 1/2, p. 71–104.

Ward, J. H., Jr., 1963, Hierarchical grouping to optimize an objective function: Jour. Am. Stat. Assoc., v. 58, no. 301, p. 236–244.

Wishart, D., 1969, An algorithm for hierarchical classifications: Biometrics, v. 25, no. 1, p. 165–170.

APPENDIX 1

Computer Program CONISS

```
      PROGRAM CONISS
***
***   CONSTRAINED INCREMENTAL SUM OF SQUARES CLUSTER ANALYSIS
***
***   BY ERIC C. GRIMM
***
***   THIS PROGRAM CARRIES OUT AN INCREMENTAL SUM OF SQUARES CLUSTER
***   ANALYSIS OF STRATIGRAPHIC DATA, CONSTRAINED SO THAT ONLY
***   STRATIGRAPHICALLY ADJACENT CLUSTERS ARE MERGED.  AS AN OPTION, THE
***   PROGRAM WILL ALSO PERFORM AN UNCONSTRAINED ANALYSIS.  THE GENERAL
***   METHOD IS THAT OF J. H. WARD, JR. (HIERARCHICAL GROUPING TO
***   OPTIMIZE AN OBJECTIVE FUNCTION. JOURNAL OF THE AMERICAN
***   STATISTICAL ASSOCIATION 58:236-244; 1963), AND THE ALGORITHM IS
***   THAT OF D. WISHART (AN ALGORITHM FOR HIERARCHICAL CLASSIFICATIONS.
***   BIOMETRICS 25:165-170; 1969).
***
***   INPUT DATA ARE READ BY SAMPLE (LEVEL) AND CAN BE COUNTS, PERCENTS,
***   OR PROPORTIONS.
***
```

```
20.  ***    CONTROL CARDS ARE READ FROM UNIT 5.
21.  ***    DATA ARE READ FROM UNIT 1.
22.  ***    RESULTS ARE WRITTEN TO UNIT 6.
23.  ***    DATA ARE WRITTEN TO UNIT 7, IF REQUIRED.
24.  ***
25.  ***    ================================================================
26.  ***
27.        PARAMETER (MAXV=25,MAXL=150,MXDM=11175)
28.  ***
29.  ***...MAXV = MAXIMUM NUMBER OF VARIABLES PERMITTED; MUST BE AT LEAST 12
30.  ***...MAXL = MAXIMUM NUMBER OF SAMPLES PERMITTED
31.  ***...MXDM = SIZE OF DISTANCE MATRIX:   (MAXL*(MAXL-1))/2
32.  ***
33.        COMMON D(MXDM),NCLUS(MAXL),NAME(MAXL)
34.        COMMON /B1/P,Q,NP,NQ,DSHORT
35.        DIMENSION X(MAXV,MAXL),Z(MAXL,MAXV),N(MAXL,MAXV)
36.        EQUIVALENCE (X,Z,N)
37.        DIMENSION YLEVS(MAXL)
38.        CHARACTER CCARD*80,FMTIN*64,FMTOUT*64
39.        LOGICAL PROP,PERC,COUNTS,SRT,STAND,NORM,PLOT,WOUT,CON
40.        DATA SRT,STAND,NORM,COUNTS,PERC,PROP,PLOT,WOUT/8*.FALSE./
41.        DATA CON/.TRUE./
42.        DATA FMTIN/'(F5.0,5X,10F7.2,/(10X,10F7.2))'/
43.        DATA FMTOUT/'(F8.2,2X,5E14.6,/(10X,5E14.6))'/
44.  ***
45.        WRITE (6,'(A,//5X,A)') '1','PROGRAM CONISS'
46.  ***
47.  ***...READ CONTROL CARDS
48.  ***
49.     10 READ (5,'(A)',END=11) CCARD
50.          IF (CCARD(1:19) .EQ. 'NUMBER OF SAMPLES =') THEN
51.              READ (CCARD(20:27),'(BN,I8)') NLEVS
52.              WRITE (6,'(//5X,A,I6)') 'NUMBER OF SAMPLES =',NLEVS
53.              IF (NLEVS .GT. MAXL) THEN
54.                WRITE (6,'(//A,I6,/A)')
55.       +          ' ***NUMBER OF SAMPLES EXCEEDS MAXIMUM ALLOWED:',MAXL,
56.       +          ' ***EXECUTION TERMINATED'
57.                STOP
58.              ENDIF
59.            ELSEIF (CCARD(1:21) .EQ. 'NUMBER OF VARIABLES =') THEN
60.              READ (CCARD(22:29),'(BN,I8)') NVARS
61.              WRITE (6,'(/5X,A,I6)') 'NUMBER OF VARIABLES =',NVARS
62.              IF (NVARS .GT. MAXV) THEN
63.                WRITE (6,'(//A,I6,/A)')
64.       +          ' ***NUMBER OF VARIABLES EXCEEDS MAXIMUM ALLOWED:',MAXV,
65.       +          ' ***EXECUTION TERMINATED'
66.                STOP
67.              ENDIF
68.            ELSEIF (CCARD .EQ. 'DATA ARE COUNTS') THEN
69.              COUNTS = .TRUE.
70.              WRITE (6,'(/5X,A)') 'INPUT DATA ARE COUNTS'
71.            ELSEIF (CCARD .EQ. 'DATA ARE PERCENTAGES') THEN
72.              PERC = .TRUE.
73.              WRITE (6,'(/5X,A)') 'INPUT DATA ARE PERCENTAGES'
74.            ELSEIF (CCARD .EQ. 'DATA ARE PROPORTIONS') THEN
75.              WRITE (6,'(/5X,A)') 'INPUT DATA ARE PROPORTIONS'
76.            ELSEIF (CCARD .EQ. 'CONVERT DATA TO PROPORTIONS') THEN
77.              PROP = .TRUE.
78.            ELSEIF (CCARD .EQ. 'SQUARE ROOT TRANSFORMATION') THEN
79.              SRT = .TRUE.
80.            ELSEIF (CCARD .EQ. 'STANDARDIZE VARIABLES') THEN
81.              STAND = .TRUE.
82.            ELSEIF (CCARD .EQ. 'NORMALIZE SAMPLES') THEN
83.              NORM = .TRUE.
84.            ELSEIF (CCARD(1:14) .EQ. 'INPUT FORMAT =') THEN
85.              FMTIN = CCARD(15:78)
86.            ELSEIF (CCARD(1:15) .EQ. 'OUTPUT FORMAT =') THEN
87.              FMTOUT = CCARD(16:79)
88.            ELSEIF (CCARD .EQ. 'UNCONSTRAINED') THEN
89.              CON = .FALSE.
90.            ELSEIF (CCARD .EQ. 'CONSTRAINED') THEN
91.              CON = .TRUE.
92.            ELSEIF (CCARD .EQ. 'WRITE DATA') THEN
93.              WOUT = .TRUE.
94.            ELSEIF (CCARD .EQ. 'PLOT DENDROGRAM') THEN
```

```
95.               PLOT = .TRUE.
96.               GOTO 11
97.             ELSE
98.               WRITE (6,'(//2A,/A)')
99.      +          ' ***CONTROL CARD NOT UNDERSTOOD: ',CCARD,
100.     +          ' ***EXECUTION TERMINATED'
101.              STOP
102.          ENDIF
103.          GOTO 10
104.    11 CONTINUE
105.       IF(NLEVS*NVARS .GT. MXDM) THEN
106.         WRITE (6,'(//A,/A,I10,/A)')
107.     +     ' ***ARRAY SPACE REQUIRED (NO. SAMPLES * NO. VARIABLES)',
108.     +     'EXCEEDS ARRAY SPACE ALLOTTED:',MXDM,
109.     +     ' ***EXECUTION TERMINATED'
110.         STOP
111.       ENDIF
112.       DO 20 I=1,NLEVS
113.         NAME(I) = I
114.    20 CONTINUE
115. ***
116. ***...READ DATA
117. ***
118.       WRITE (6,'(/5X,2A)') 'FORMAT OF INPUT DATA: ',FMTIN
119.       DO 30 I=1,NLEVS
120.         READ (1,FMTIN) YLEVS(I),(X(J,I),J=1,NVARS)
121.    30 CONTINUE
122. ***
123. ***...CONVERT TO PROPORTIONS IF NECESSARY
124. ***
125.       IF (PROP) THEN
126.         IF (COUNTS) THEN
127.           DO 60 J=1,NLEVS
128.             SUM = 0.0
129.             DO 40 I=1,NVARS
130.               SUM = SUM+X(I,J)
131.    40         CONTINUE
132.             DO 50 I=1,NVARS
133.               X(I,J) = X(I,J)/SUM
134.    50         CONTINUE
135.    60       CONTINUE
136.           WRITE (6,'(/5X,A)') 'DATA CONVERTED TO PROPORTIONS'
137.         ELSEIF (PERC) THEN
138.           DO 80 J=1,NLEVS
139.             DO 70 I=1,NVARS
140.               X(I,J) = 0.01*X(I,J)
141.    70         CONTINUE
142.    80       CONTINUE
143.           WRITE (6,'(/5X,A)') 'DATA CONVERTED TO PROPORTIONS'
144.       ENDIF
145.     ENDIF
146. ***
147. ***...TRANSFORM DATA IF NECESSARY
148. ***
149.       IF (SRT) THEN
150.         DO 100 J=1,NLEVS
151.           DO 90 I=1,NVARS
152.             X(I,J) = SQRT(X(I,J))
153.    90       CONTINUE
154.   100     CONTINUE
155.         WRITE (6,'(/5X,A)') 'SQUARE ROOT TRANSFORMATION'
156.         IF (PROP .OR. PERC) WRITE (6,'(/5X,2A)')
157.     +     'DISSIMILARITY COEFICIENT IS EDWARDS AND CAVALLI-SFORZA''S',
158.     +     ' CHORD DISTANCE'
159.       ELSEIF (STAND) THEN
160.         XLEVS = REAL(NLEVS)
161.         XLEVS1 = XLEVS-1.0
162.         DO 130 I=1,NVARS
163.           SX = 0.0
164.           SX2 = 0.0
165.           DO 110 J=1,NLEVS
166.             XIJ = X(I,J)
167.             SX = SX+XIJ
168.             SX2 = SX2+XIJ*XIJ
169.   110       CONTINUE
```

```
170.              IF (SX .GT. 0.0) THEN
171.                XBAR = SX/XLEVS
172.                SD = SQRT((SX2-SX*SX/XLEVS)/XLEVS1)
173.                DO 120 J=1,NLEVS
174.                  X(I,J) = (X(I,J)-XBAR)/SD
175.    120         CONTINUE
176.              ENDIF
177.    130     CONTINUE
178.            WRITE (6,'(/5X,A,//5X,2A)')
179.       +      'VARIABLES STANDARDIZED TO MEAN 0, STANDARD DEVIATION 1',
180.       +      'DISSIMILARITY COEFICIENT IS STANDARDIZED EUCLIDIAN ',
181.       +      'DISTANCE'
182.          ELSEIF (NORM) THEN
183.            DO 160 J=1,NLEVS
184.              SX2 = 0.0
185.              DO 140 I=1,NVARS
186.                XIJ = X(I,J)
187.                SX2 = SX2+XIJ*XIJ
188.    140       CONTINUE
189.              SX = SQRT(SX2)
190.              DO 150 I=1,NVARS
191.                X(I,J) = X(I,J)/SX
192.    150       CONTINUE
193.    160     CONTINUE
194.            WRITE (6,'(/5X,A,//5X,A)')
195.       +      'SAMPLE VECTORS NORMALIZED TO LENGTH 1',
196.       +      'DISSIMILARITY COEFICIENT IS ORLOCI''S CHORD DISTANCE'
197.          ELSE
198.            WRITE (6,'(/5X,A,//5X,A)') 'NO DATA TRANSFORMATION',
199.       +      'DISSIMILARITY COEFICIENT IS EUCLIDIAN DISTANCE'
200.         ENDIF
201.  ***
202.  ***...WRITE OUT DATA IF NECESSARY
203.  ***
204.        IF (WOUT) THEN
205.          DO 170 I=1,NLEVS
206.            WRITE (7,FMTOUT) YLEVS(I),(X(J,I),J=1,NVARS)
207.    170   CONTINUE
208.        ENDIF
209.  ***
210.  ***...GENERATE DISTANCE MATRIX
211.  ***
212.        ID = 0
213.        DO 200 I=2,NLEVS
214.          DO 190 J=1,I-1
215.            DSQD = 0.0
216.            DO 180 K=1,NVARS
217.              DSQD = DSQD+(X(K,I)-X(K,J))**2
218.    180     CONTINUE
219.            ID = ID+1
220.            D(ID) = DSQD
221.    190   CONTINUE
222.    200 CONTINUE
223.  ***
224.  ***...CLUSTER
225.  ***
226.        IF (CON) THEN
227.            CALL CCLUS(NLEVS,Z(1,1),Z(1,5),N(1,6),N(1,7))
228.            WRITE (6,'(A,//5X,A)') '1',
229.       +      'CONSTRAINED INCREMENTAL SUM OF SQUARES CLUSTER ANALYSIS'
230.          ELSE
231.            CALL UCLUS(NLEVS,Z(1,1),Z(1,5),N(1,6),N(1,7))
232.            WRITE (6,'(A,//5X,A)') '1',
233.       +      'UNCONSTRAINED INCREMENTAL SUM OF SQUARES CLUSTER ANALYSIS'
234.         ENDIF
235.  ***
236.  ***...WRITE RESULTS
237.  ***
238.        WRITE (6,'(//T69,A,/T55,A,T69,A,/T15,A,T27,A,T41,A,T55,A,T69,A,
239.       +  /T5,A,T15,A,2X,4(4X,A)/)') 'MEAN','WITHIN-','WITHIN-',
240.       +  'CLUSTERS','INCREASE IN','TOTAL','CLUSTER','CLUSTER','STAGE',
241.       +  'MERGED','DISPERSION','DISPERSION','DISPERSION','DISPERSION'
242.        DO 210 I=1,NLEVS-1
243.          WRITE (6,'(5X,I4,2X,2I5,2X,4E14.7)')
244.       +    I,N(I,6),N(I,7),(Z(I,J),J=1,4)
```

```
  210 CONTINUE
      WRITE (6,'(A,//5X,A/)') '1','SAMPLE NUMBERS'
      DO 220 I=1,NLEVS
        WRITE (6,'(5X,I4,F10.2)') I,YLEVS(I)
  220 CONTINUE
      IF (.NOT. CON) THEN
        CALL DORDER(NLEVS,MAXL,N(1,6),N(1,7),N(1,8),N(1,9),N(1,10),
     +    N(1,11),N(1,12))
      ENDIF
***
***...PLOT DENDROGRAM
***
      IF (PLOT) THEN
        CALL DDGRAM(NLEVS,YLEVS,Z(1,1),Z(1,5),N(1,6),N(1,7),N(1,8),
     +    N(1,9),Z(1,10),Z(1,11),MAXL,CON)
      ENDIF
      STOP
      END
***
***   ------------------------------------------------------------------
***
      SUBROUTINE CCLUS(NLEVS,ES,ESS,NAMEP,NAMEQ)
***
***   CONSTRAINED CLUSTER ANALYSIS
***   ==================================================================
***
      PARAMETER (MAXL=150,MXDM=11175)
      COMMON D(MXDM),NCLUS(MAXL),NAME(MAXL)
      COMMON /B1/P,Q,NP,NQ,DSHORT
      DIMENSION ES(MAXL,4)
      DIMENSION NAMEP(MAXL),NAMEQ(MAXL),ESS(MAXL)
      INTEGER P,Q,R
***
***...INITIALIZE ARRAYS AND VARIABLES
***
      DO 10 I=1,MAXL
        NCLUS(I) = 1
   10 CONTINUE
      DO 20 I=1,MAXL
        ESS(I) = 0.0
   20 CONTINUE
      MSIZ = NLEVS-1
      E = 0.0
***
***...BEGIN CLUSTERING
***
      DO 100 ITER=1,MSIZ
***
***.....FIND MOST SIMILAR CLUSTERS
***
        DSHORT = D(1)
        P = 1
        ID = 1
        DO 30 N=2,MSIZ
          ID = ID+N
          IF (D(ID) .LT. DSHORT) THEN
            DSHORT = D(ID)
            P = N
          ENDIF
   30   CONTINUE
        Q = P+1
        NAMP = NAME(P)
        NAMQ = NAME(Q)
        NAMEP(ITER) = NAMP
        NAMEQ(ITER) = NAMQ
        NP = NCLUS(NAMP)
        NQ = NCLUS(NAMQ)
***
***.....CALCULATE MERGE DATA
***
        DE = 0.5*DSHORT
        E = E+DE
        ESS(NAMP) = ESS(NAMP)+ESS(NAMQ)+DE
        ES(ITER,1) = DE
        ES(ITER,2) = E
```

```
320.           ES(ITER,3) = ESS(NAMP)
321.           ES(ITER,4) = ESS(NAMP)/REAL(NP+NQ)
322.     ***
323.     ***.....UPDATE DISTANCE MATRIX
324.     ***
325.           IP = (P-1)*(P-2)/2
326.           IQ = (Q-1)*(Q-2)/2
327.           DO 50 J=1,P-1
328.             ID = IP+J
329.             D(ID) = UPDATE(J)
330.             DO 40 I=Q,MSIZ
331.               IJ = J+(I-1)*(I-2)/2
332.               I1J = IJ+I-1
333.               D(IJ) = D(I1J)
334.        40     CONTINUE
335.        50   CONTINUE
336.           IR = IQ-Q+2
337.           DO 60 I=Q,MSIZ
338.             IR = IR+I-2
339.             ID = IR+P
340.             D(ID) = UPDATE(I+1)
341.        60   CONTINUE
342.           DO 80 J=Q,MSIZ-1
343.             DO 70 I=J+1,MSIZ
344.               IJ = J+(I-1)*(I-2)/2
345.               I1J = IJ+I
346.               D(IJ) = D(I1J)
347.        70     CONTINUE
348.        80   CONTINUE
349.           DO 90 M=Q,MSIZ
350.             NAME(M) = NAME(M+1)
351.        90   CONTINUE
352.           NCLUS(NAMP) = NP+NQ
353.           MSIZ = MSIZ-1
354.       100 CONTINUE
355.           RETURN
356.           END
357.     ***
358.     ***   ------------------------------------------------------------------
359.     ***
360.           SUBROUTINE UCLUS(NLEVS,ES,ESS,NAMEP,NAMEQ)
361.     ***
362.     ***   UNCONSTRAINED CLUSTER ANALYSIS
363.     ***   ==================================================================
364.     ***
365.           PARAMETER (MAXL=150,MXDM=11175)
366.           COMMON D(MXDM),NCLUS(MAXL),NAME(MAXL)
367.           COMMON /B1/P,Q,NP,NQ,DSHORT
368.           DIMENSION ES(MAXL,4)
369.           DIMENSION NAMEP(MAXL),NAMEQ(MAXL),ESS(MAXL)
370.           INTEGER P,Q,R
371.     ***
372.     ***...INITIALIZE ARRAYS AND VARIABLES
373.     ***
374.           DO 10 I=1,MAXL
375.             NCLUS(I) = 1
376.        10 CONTINUE
377.           DO 20 I=1,MAXL
378.             ESS(I) = 0.0
379.        20 CONTINUE
380.           MSIZ = NLEVS-1
381.           E = 0.0
382.     ***
383.     ***...BEGIN CLUSTERING
384.     ***
385.           DO 200 ITER=1,MSIZ
386.     ***
387.     ***.....FIND MOST SIMILAR CLUSTERS
388.     ***
389.             DSHORT = D(1)
390.             P = 1
391.             Q = 2
392.             ID = 1
393.             DO 40 I=3,MSIZ+1
394.               DO 30 J=1,I-1
```

```
395.                ID = ID+1
396.                IF (D(ID) .LT. DSHORT) THEN
397.                  DSHORT = D(ID)
398.                  P = J
399.                  Q = I
400.                ENDIF
401.     30       CONTINUE
402.     40     CONTINUE
403.            NAMP = NAME(P)
404.            NAMQ = NAME(Q)
405.            NAMEP(ITER) = NAMP
406.            NAMEQ(ITER) = NAMQ
407.            NP = NCLUS(NAMP)
408.            NQ = NCLUS(NAMQ)
409.   ***
410.   ***.....CALCULATE MERGE DATA
411.   ***
412.            DE = 0.5*DSHORT
413.            E = E+DE
414.            ESS(NAMP) = ESS(NAMP)+ESS(NAMQ)+DE
415.            ES(ITER,1) = DE
416.            ES(ITER,2) = E
417.            ES(ITER,3) = ESS(NAMP)
418.            ES(ITER,4) = ESS(NAMP)/REAL(NP+NQ)
419.   ***
420.   ***.....UPDATE DISTANCE MATRIX
421.   ***
422.            IP = (P-1)*(P-2)/2
423.            DO 60 J=1,P-1
424.              ID = IP+J
425.              D(ID) = UPDATE(J)
426.              DO 50 I=Q,MSIZ
427.                IJ = J+(I-1)*(I-2)/2
428.                I1J = IJ+I-1
429.                D(IJ) = D(I1J)
430.     50       CONTINUE
431.     60     CONTINUE
432.            DO 70 I=P+1,Q-1
433.              ID = P+(I-1)*(I-2)/2
434.              D(ID) = UPDATE(I)
435.     70     CONTINUE
436.            DO 80 I=Q,MSIZ
437.              ID = P+(I-1)*(I-2)/2
438.              D(ID) = UPDATE(I+1)
439.     80     CONTINUE
440.            DO 100 J=P+1,Q-1
441.              DO 90 I=Q,MSIZ
442.                IJ = J+(I-1)*(I-2)/2
443.                I1J = IJ+I-1
444.                D(IJ) = D(I1J)
445.     90       CONTINUE
446.    100     CONTINUE
447.            DO 120 J=Q,MSIZ-1
448.              DO 110 I=J+1,MSIZ
449.                IJ = J+(I-1)*(I-2)/2
450.                I1J = IJ+I
451.                D(IJ) = D(I1J)
452.    110       CONTINUE
453.    120     CONTINUE
454.            DO 130 M=Q,MSIZ
455.              NAME(M) = NAME(M+1)
456.    130     CONTINUE
457.            NCLUS(NAMP) = NP+NQ
458.            MSIZ = MSIZ-1
459.    200   CONTINUE
460.          RETURN
461.          END
462.   ***
463.   ***    ----------------------------------------------------------------
464.   ***
465.          FUNCTION UPDATE(R)
466.   ***
467.   ***...THIS FUNCTION SOLVES THE UPDATE EQUATION
468.   ***
469.          PARAMETER (MAXL=150,MXDM=11175)
```

```
      COMMON D(MXDM),NCLUS(MAXL),NAME(MAXL)
      COMMON /B1/P,Q,NP,NQ,DSHORT
      INTEGER P,Q,R
      NR = NCLUS(NAME(R))
      UPDATE = ((NR+NP)*D2(R,P)+(NR+NQ)*D2(R,Q)-NR*DSHORT)/(NR+NP+NQ)
      RETURN
      END
***
***   ------------------------------------------------------------------
***
      FUNCTION D2(I,J)
***
***...THIS FUNCTION LOCATES VALUE IN LOWER HALF MATRIX
***
      PARAMETER (MAXL=150,MXDM=11175)
      COMMON D(MXDM)
      IF (I .GT. J) THEN
          ID = (I-1)*(I-2)/2+J
        ELSE
          ID = (J-1)*(J-2)/2+I
      ENDIF
      D2 = D(ID)
      RETURN
      END
***
***   ------------------------------------------------------------------
***
      SUBROUTINE DORDER(NLEVS,MAXL,IP,JP,NORD,IL,JL,LAST,NEXT)
***
***   THIS SUBROUTINE ESTABLISHES THE ORDER OF SAMPLES ON THE
***   DENDROGRAM FOR UNCONSTRAINED CLUSTER ANALYSIS.  THE ALGORITHM IS
***   ADAPTED FROM SUBROUTINE TREE IN M. R. ANDERBERG (CLUSTER ANALYSIS
***   FOR APPLICATIONS. ACADEMIC PRESS, NEW YORK. 1973).
***   ==================================================================
***
      DIMENSION IP(MAXL),JP(MAXL),IL(MAXL),JL(MAXL),NORD(MAXL),
     +  LAST(MAXL),NEXT(0:MAXL-1)
***
      N = NLEVS-1
      DO 10 K=1,NLEVS
        LAST(K) = 0
        NORD(K) = 0
   10 CONTINUE
      DO 20 K=1,N
        IK = IP(K)
        JK = JP(K)
        IL(K) = LAST(IK)
        JL(K) = LAST(JK)
        LAST(IK) = K
        NEXT(IL(K)) = K
        NEXT(JL(K)) = K
   20 CONTINUE
      NEXT(N) = 0
      K = 1
      NO = 0
   30 IK = IP(K)
      JK = JP(K)
      IF (IL(K) .EQ. 0) THEN
        NO = NO+1
        NORD(NO) = IK
      ENDIF
      IF (JL(K) .EQ. 0) THEN
        NO = NO+1
        NORD(NO) = JK
      ENDIF
      KLAST = K
      K = NEXT(K)
      IF (K .LE. N .AND. K .GE. 1) THEN
        IF (IL(K) .LE. 0) THEN
          IL(K) = -IL(K)
          GOTO 30
        ENDIF
        IF (JL(K) .LE. 0) THEN
          JL(K) = -JL(K)
          GOTO 30
```

```
545.            ENDIF
546.            IF (IL(K) .NE. KLAST) THEN
547.                JL(K) = -JL(K)
548.                K = IL(K)
549.              ELSE
550.                IL(K) = -IL(K)
551.                K = JL(K)
552.            ENDIF
553.     40     ILK = IL(K)
554.            JLK = JL(K)
555.            IF (ILK .LT. JLK) THEN
556.                IF (ILK .EQ. 0) THEN
557.                    K = JLK
558.                  ELSE
559.                    K = ILK
560.                ENDIF
561.              ELSEIF (ILK .GT. JLK) THEN
562.                IF (JLK .EQ. 0) THEN
563.                    K = ILK
564.                  ELSE
565.                    K = JLK
566.                ENDIF
567.              ELSE
568.                GOTO  30
569.            ENDIF
570.            GOTO 40
571.          ENDIF
572.  ***
573.  ***...WRITE SAMPLE ORDER
574.  ***
575.          WRITE (6,'(A,//5X,A/)') '1','ORDER OF SAMPLES ON DENDROGRAM'
576.          WRITE (6,'(2014)')  (NORD(I),I=1,NO)
577.          RETURN
578.          END
579.  ***
580.  ***    ----------------------------------------------------------------------
581.  ***
582.          SUBROUTINE DDGRAM(NLEVS,YLEVS,ES,DEPTHS,IP,JP,NORD,IPOS,Y,XDD,
583.         +  MAXL,CON)
584.  ***
585.  ***    THIS SUBROUTINE PLOTS A DENDROGRAM WITH THE DISSPLA GRAPHICS
586.  ***    PACKAGE.
587.  ***    ======================================================================
588.  ***
589.          DIMENSION ES(MAXL,4),DEPTHS(MAXL),IP(MAXL),JP(MAXL),NORD(MAXL),
590.         +  IPOS(MAXL),Y(MAXL),XDD(MAXL),YLEVS(MAXL)
591.          CHARACTER CCARD*80,LXNAME(4)*48
592.          LOGICAL DEFLT,CON
593.          DATA DEFLT/.TRUE./
594.          DATA LXNAME(1)/'INCREASE IN DISPERSION$'/
595.          DATA LXNAME(2)/'TOTAL DISPERSION$'/
596.          DATA LXNAME(3)/'WITHIN-CLUSTER DISPERSION$'/
597.          DATA LXNAME(4)/'MEAN WITHIN-CLUSTER DISPERSION$'/
598.          DATA HITE,XAXIS,YAXIS/0.10,4.0,10.0/
599.          DATA XAXIS1,XORGN/0.0,2.0/
600.  ***
601.  ***...INITIALIZE DEPTHS (Y-AXIS VALUES)
602.  ***
603.          IF (CON) THEN
604.              DO 10 I=1,NLEVS
605.                DEPTHS(I) = YLEVS(I)
606.     10       CONTINUE
607.            ELSE
608.              DO 20 I=1,NLEVS
609.                DEPTHS(I) = REAL(I)
610.     20       CONTINUE
611.          ENDIF
612.  ***
613.  ***...INITIALIZE DISSPLA.  THIS CALL IS INSTALLATION DEPENDENT.
614.  ***
615.          CALL MNPLOT
616.  ***
617.          CALL RESET('ALL')
618.          CALL NOBRDR
619.          CALL PAGE(100.0,14.08)
```

```
***
***...ARGUMENTS IN CALL TO PAGE ARE DEPENDENT ON PLOTTING DEVICE.
***
***...READ CONTROL CARDS
***
      IGR = 1
   30 READ (5,'(A)',END=40) CCARD
        IF (CCARD(1:8) .EQ. 'HEIGHT =') THEN
            READ (CCARD(9:16),'(BN,F8.0)') XAXIS
          ELSEIF (CCARD(1:7) .EQ. 'WIDTH =') THEN
            READ (CCARD(8:15),'(BN,F8.0)') YAXIS
          ELSEIF (CCARD(1:7) .EQ. 'SCALE =') THEN
            READ (CCARD(8:15),'(BN,I8)') ISCALE
            DEFLT = .FALSE.
***
***.....PLOT DEPTH COLUMN
***
            IF (CON .AND. IGR .EQ. 1) THEN
              CALL DEPCOL(NLEVS,DEPTHS,XORGN,XAXIS1,YAXIS,HITE,MAXL)
              IGR = 2
            ENDIF
***
***.....PLOT TREE
***
            CALL TREE(NLEVS,XAXIS,XORGN,XAXIS1,YAXIS,DEPTHS,IP,JP,NORD,
     +        IPOS,ES(1,ISCALE),Y,XDD,IGR,LXNAME(ISCALE),HITE,MAXL,CON)
            IGR = IGR+1
          ELSEIF (CCARD(1:19) .EQ. 'CHARACTER HEIGHT = ') THEN
            READ (CCARD(20:27),'(BN,F8.0)') HITE
          ELSE
            WRITE (6,'(//2A,/A)')
     +        ' ***PLOTTING CONTROL CARD NOT UNDERSTOOD: ',CCARD,
     +        ' ***EXECUTION TERMINATED'
            STOP
        ENDIF
        GOTO 30
   40 CONTINUE
***
***...IF SCALE NOT SPECIFIED, PLOT DEFAULT DENDROGRAM.
***
      IF (DEFLT) THEN
        IF (CON) THEN
          CALL DEPCOL(NLEVS,DEPTHS,XORGN,XAXIS1,YAXIS,HITE,MAXL)
          IGR = 2
        ENDIF
        CALL TREE(NLEVS,XAXIS,XORGN,XAXIS1,YAXIS,DEPTHS,IP,JP,NORD,
     +    IPOS,ES(1,2),Y,XDD,IGR,LXNAME(2),HITE,MAXL,CON)
      ENDIF
***
***...TERMINATE PLOT
***
      CALL ENDPL(1)
      CALL DONEPL
      RETURN
      END
***
***   ------------------------------------------------------------------
***
      SUBROUTINE TREE(NLEVS,XAXIS,XORGN,XAXIS1,YAXIS,DEPTHS,IP,JP,NORD,
     +  IPOS,X,Y,XDD,IGR,LXNAME,HITE,MAXL,CON)
***
***   THIS SUBROUTINE PLOTS THE TREE
***   ==================================================================
***
      DIMENSION DEPTHS(MAXL),IP(MAXL),JP(MAXL),X(MAXL),IPOS(MAXL),
     +  Y(MAXL),XARAY(4),YARAY(4),XDD(MAXL),NORD(MAXL)
      LOGICAL CON
      CHARACTER LXNAME*48
***
***...SET PARAMETER VALUES
***
      DO 10 I=1,MAXL
        XDD(I) = 0.0
   10 CONTINUE
      DO 20 I=1,NLEVS
```

```
695.            Y(I) = DEPTHS(I)
696.        20 CONTINUE
697.           NAMALG = NLEVS-1
698.           TDEPTH = DEPTHS(NLEVS)-DEPTHS(1)
699.     ***
700.     ***...DETERMINE MAXIMUM VALUE STEP SIZE FOR TREE
701.     ***
702.           XMAX = X(NAMALG)
703.           DO 30 I=1,NAMALG-1
704.             IF (X(I) .GT. XMAX) XMAX = X(I)
705.        30 CONTINUE
706.           XF = 1.0E-10
707.           DO 40 I=1,100
708.             XF = XF*10.0
709.             IF (XF .GT. XMAX) GOTO 50
710.        40 CONTINUE
711.        50 XSTP = 0.1*XF
712.           IF (3.0*XSTP .GT. XMAX) THEN
713.               XSTP = 0.5*XSTP
714.             ELSEIF (7.0*XSTP .LT. XMAX) THEN
715.               XSTP = 2.0*XSTP
716.           ENDIF
717.           XF = 0.0
718.           DO 60 I=1,10
719.             XF = XF+XSTP
720.             IF (XF .GT. XMAX) THEN
721.               XMAX = XF
722.               GOTO 70
723.             ENDIF
724.        60 CONTINUE
725.     ***
726.     ***...SETUP SUBPLOT
727.     ***
728.        70 IF (CON) THEN
729.               XORGN = XORGN+XAXIS1+0.3
730.               CALL PHYSOR(XORGN,0.8)
731.             ELSE
732.               XORGN = XORGN+YAXIS+2.0
733.               CALL PHYSOR(XORGN,0.0)
734.               CALL BANGLE(90.0)
735.           ENDIF
736.           XAXIS1 = XAXIS+1.0
737.           YAXIS1 = YAXIS+0.15
738.           CALL AREA2D(XAXIS1,YAXIS1)
739.     ***
740.     ***...SETUP AXIS SYSTEM
741.     ***
742.           XORIG = -XMAX/XAXIS
743.           YORIG = DEPTHS(NLEVS)+0.15*TDEPTH/YAXIS
744.           YMAX = DEPTHS(1)
745.           CALL XTICKS(0)
746.           CALL YTICKS(0)
747.           CALL XNONUM
748.           CALL YNONUM
749.           YSTP = (YMAX-YORIG)/10.0
750.           CALL GRAF(XORIG,XSTP,XMAX,YORIG,YSTP,YMAX)
751.     ***
752.     ***...PLOT TREE
753.     ***
754.           IF (CON) THEN
755.               DO 80 I=1,NLEVS
756.                 IPOS(I) = I
757.        80       CONTINUE
758.             ELSE
759.               DO 90 I=1,NLEVS
760.                 IPOS(NORD(I)) = I
761.        90       CONTINUE
762.           ENDIF
763.           DO 100 I=1,NAMALG
764.             IPI = IPOS(IP(I))
765.             JPI = IPOS(JP(I))
766.             YI = Y(IPI)
767.             YJ = Y(JPI)
768.             XI = XDD(IPI)
769.             XJ = XDD(JPI)
```

```
770.          YARAY(1) = YI
771.          YARAY(2) = YI
772.          YARAY(3) = YJ
773.          YARAY(4) = YJ
774.          XARAY(1) = XI
775.          XARAY(2) = X(I)
776.          XARAY(3) = X(I)
777.          XARAY(4) = XJ
778.          CALL CURVE(XARAY,YARAY,4,0)
779.          Y(IPI) = (YI+YJ)/2.0
780.          XDD(IPI) = X(I)
781.    100 CONTINUE
782.        YROOT = Y(IPOS(IP(I-1)))
783.        CALL RLVEC(X(NAMALG),YROOT,XMAX,YROOT,0000)
784.  ***
785.  ***...DRAW X-AXES BELOW AND ABOVE; LABEL GRAPH
786.  ***
787.        CALL RESET('XNONUM')
788.        CALL HEIGHT(1.4*HITE)
789.        CALL XTICKS(2)
790.        CALL XINTAX
791.        YPOS = YAXIS1+0.1
792.        IF (CON) THEN
793.          CALL XGRAXS(0.,XSTP,XMAX,XAXIS,' ',1,1.0,0.)
794.          CALL XNONUM
795.          CALL HEIGHT(HITE)
796.          CALL XREVTK
797.          YPOS = YPOS+0.05
798.        ENDIF
799.        CALL XGRAXS(0.,XSTP,XMAX,XAXIS,LXNAME,-100,1.0,YPOS)
800.  ***
801.  ***...WRITE SAMPLE DEPTHS OR NAMES
802.  ***
803.        IF (CON) THEN
804.            CALL MSHIFT(0.0,-0.5*HITE)
805.            YV = DEPTHS(1)
806.            XL = XREAL(YV,1)
807.            XIN = 0.8-XL
808.            XV = XINVRS(XIN,0.0)
809.            CALL RLREAL(YV,1,XV,YV)
810.            DO 110 I=10,NLEVS,10
811.              YV = DEPTHS(I)
812.              XL = XREAL(YV,1)
813.              XIN = 0.8-XL
814.              XV = XINVRS(XIN,0.0)
815.              CALL RLREAL(YV,1,XV,YV)
816.    110       CONTINUE
817.          ELSE
818.            H = YAXIS/REAL(NLEVS)-0.03
819.            IF (H .LT. HITE) THEN
820.                CALL HEIGHT(H)
821.                H = -0.5*H
822.              ELSE
823.                CALL HEIGHT(HITE)
824.                H = -0.5*HITE
825.            ENDIF
826.            CALL MSHIFT(0.0,H)
827.            DO 120 I=1,NLEVS
828.              NORDI = NORD(I)
829.              XIN = 0.8-XINT(NORDI)
830.              XV = XINVRS(XIN,0.0)
831.              CALL RLINT(NORDI,XV,DEPTHS(I))
832.    120       CONTINUE
833.        ENDIF
834.        CALL ENDGR(IGR)
835.        CALL RESET('BANGLE')
836.        CALL RESET('XTICKS')
837.        CALL RESET('YTICKS')
838.        CALL RESET('XNONUM')
839.        CALL RESET('YNONUM')
840.        CALL RESET('XINTAX')
841.        CALL RESET('XREVTK')
842.        CALL RESET('MSHIFT')
843.        CALL RESET('HEIGHT')
844.        RETURN
```

```
      END
***
***   ------------------------------------------------------------------
***
      SUBROUTINE DEPCOL(NLEVS,DEPTHS,XORGN,XAXIS1,YAXIS,HITE,MAXL)
***
***   THIS SUBROUTINE PLOTS A DEPTH COLUMN
***   ==================================================================
***
      DIMENSION DEPTHS(MAXL)
***
***...SET PARAMTER VALUES
***
      H = HITE/2.0
      H2 = H+0.02
      H22 = 2.0*H2
      TDEPTH = DEPTHS(NLEVS)-DEPTHS(1)
***
***...DEFINE SUBPLOT AREA
***
      CALL PHYSOR(XORGN,0.8)
      YAXIS1 = YAXIS+0.25
      XAXIS1 = 0.6
      CALL AREA2D(XAXIS1,YAXIS1)
***
***...DEFINE AXES
***
      YORIG = DEPTHS(NLEVS)+0.15*TDEPTH/YAXIS
      YMAX = DEPTHS(1)-0.1*TDEPTH/YAXIS
      CALL XTICKS(0)
      CALL YTICKS(0)
      CALL XNONUM
      CALL YNONUM
      CALL GRAF(0.0,0.1,0.6,YORIG,10.0,YMAX)
***
***...DRAW VERTICAL LINES FOR DEPTH COLUMN
***
      CALL VECTOR(0.0,0.0,0.0,YAXIS1)
      CALL VECTOR(0.6,0.0,0.6,YAXIS1)
***
***...BLANK AREAS FOR SAMPLE NUMBERS
***
      YBLNK = YPOSN(0.0,DEPTHS(1))
      CALL BLREC(0.2,YBLNK-H2,0.2,H22,0)
      DO 30 I=10,NLEVS,10
        YBLNK = YPOSN(0.0,DEPTHS(I))
        CALL BLREC(0.15,YBLNK-H2,0.3,H22,0)
   30 CONTINUE
***
***...DRAW LEVEL LINES
***
      DO 40 I=1,NLEVS
        CALL RLVEC(0.0,DEPTHS(I),0.6,DEPTHS(I),0000)
   40 CONTINUE
***
***...WRITE SAMPLE NUMBERS
***
      CALL HEIGHT(HITE)
      CALL MSHIFT(0.0,-H)
      CALL BLOFF(1)
      CALL RLINT(1,0.3,DEPTHS(1))
      XVAL = 0.3-0.8*HITE
      ID = 1
      DO 50 I=10,NLEVS,10
        ID = ID+1
        CALL BLOFF(ID)
        CALL RLINT(I,XVAL,DEPTHS(I))
   50 CONTINUE
      CALL ENDGR(1)
      CALL RESET('MSHIFT')
      CALL RESET('HEIGHT')
      CALL RESET('XTICKS')
      CALL RESET('YTICKS')
      CALL RESET('XNONUM')
      CALL RESET('YNONUM')
```

```
920.        RETURN
921.        END
```

APPENDIX 2

Example of Program Input and Output. The Data are the top 12 Levels from the Wolsfeld Lake Core

Control cards

```
NUMBER OF SAMPLES = 12
NUMBER OF VARIABLES = 17
DATA ARE COUNTS
CONVERT DATA TO PROPORTIONS
SQUARE ROOT TRANSFORMATION
PLOT DENDROGRAM
CHARACTER HEIGHT = 0.12
HEIGHT = 3.5
WIDTH = 6.0
SCALE = 2
```

Input data

```
650.    7.5   1.0  19.0   9.0  10.0   1.0  26.0  226.0   77.0   28.0
650.   15.0  26.0  25.0  83.0  12.0   9.0  12.0
662.    3.5   1.0   8.0   7.0  15.0   2.0  14.0  178.0   69.0   25.0
662.   10.0  16.0  41.0 112.0  12.0  18.0  15.0
670.    3.5     0   6.0   4.0   4.0   2.0  11.0  120.0   38.0   19.0
670.   10.0   8.0  24.0 109.0   6.0  13.0  17.0
680.    2.5   1.0   7.0   3.0  14.0   1.0  25.0  264.0   32.0   41.0
680.    4.0  30.5  43.0 100.0  17.0   8.0   6.0
690.    4.5     0   9.0   9.0  11.0   3.0  17.0  173.0   30.0   24.0
690.   15.0  16.0  56.0 103.0  13.0  10.0  19.0
702.    1.0   2.0   5.0   5.0  10.0   4.0  17.0  139.0   36.0   39.0
702.    4.0  15.0  24.0  79.0   7.0  13.0  12.0
710.    6.5   1.0  21.0  10.0  16.0   4.0  19.0  172.0   41.0   66.0
710.   10.0  18.0  53.0  29.0  13.0  18.0  12.0
714.    1.5   2.0  10.0   8.0  30.0  11.0   9.0  228.0   33.0   88.0
714.   11.0  29.5  65.0  25.0  19.0  13.0   9.0
722.    2.0   6.0  13.0   4.0  11.0   4.0  13.0  254.0   69.0  137.0
722.   20.0  35.0  44.0  15.0  14.0  10.0   9.0
730.    6.5   2.0  26.0  10.0  22.0   9.0  14.0  265.0   49.0  134.0
730.   27.0   7.0  65.0  11.0  21.0  12.0   7.0
738.    1.5   2.0  24.0  13.0  33.0   8.0  13.0  274.0   66.0  168.0
738.   27.0   6.0  60.0  12.0  20.0  11.0   3.0
742.    2.5   3.0  25.0  11.0  28.0  19.0   6.0  238.0   38.0  179.0
742.   17.0   4.0  63.0  28.0  21.0  14.0  11.0
```

Output data

```
1

     PROGRAM CONISS


     NUMBER OF SAMPLES =    12

     NUMBER OF VARIABLES =    17

     INPUT DATA ARE COUNTS

     FORMAT OF INPUT DATA: (F5.0,5X,10F7.2,/(10X,10F7.2))

     DATA CONVERTED TO PROPORTIONS

     SQUARE ROOT TRANSFORMATION

     DISSIMILARITY COEFICIENT IS EDWARDS AND CAVALLI-SFORZA'S CHORD DISTANCE
1
```

CONSTRAINED INCREMENTAL SUM OF SQUARES CLUSTER ANALYSIS

STAGE	CLUSTERS MERGED		INCREASE IN DISPERSION	TOTAL DISPERSION	WITHIN-CLUSTER DISPERSION	MEAN WITHIN-CLUSTER DISPERSION
1	10	11	0.4494294E-02	0.4494294E-02	0.4494294E-02	0.2247147E-02
2	2	3	0.9415898E-02	0.1391019E-01	0.9415898E-02	0.4707949E-02
3	10	12	0.1351028E-01	0.2742047E-01	0.1800457E-01	0.6001524E-02
4	7	8	0.1462318E-01	0.4204365E-01	0.1462318E-01	0.7311590E-02
5	5	6	0.1816634E-01	0.6020999E-01	0.1816634E-01	0.9083171E-02
6	4	5	0.1905571E-01	0.7926570E-01	0.3722205E-01	0.1240735E-01
7	1	2	0.2396281E-01	0.1032285E+00	0.3337870E-01	0.1112623E-01
8	1	4	0.2538838E-01	0.1286169E+00	0.9598913E-01	0.1599819E-01
9	7	9	0.2574420E-01	0.1543611E+00	0.4036738E-01	0.1345579E-01
10	7	10	0.4416988E-01	0.1985310E+00	0.1025418E+00	0.1709031E-01
11	1	7	0.3831516E+00	0.5816826E+00	0.5816826E+00	0.4847355E-01

1

SAMPLE NUMBERS

1	650.00
2	662.00
3	670.00
4	680.00
5	690.00
6	702.00
7	710.00
8	714.00
9	722.00
10	730.00
11	738.00
12	742.00