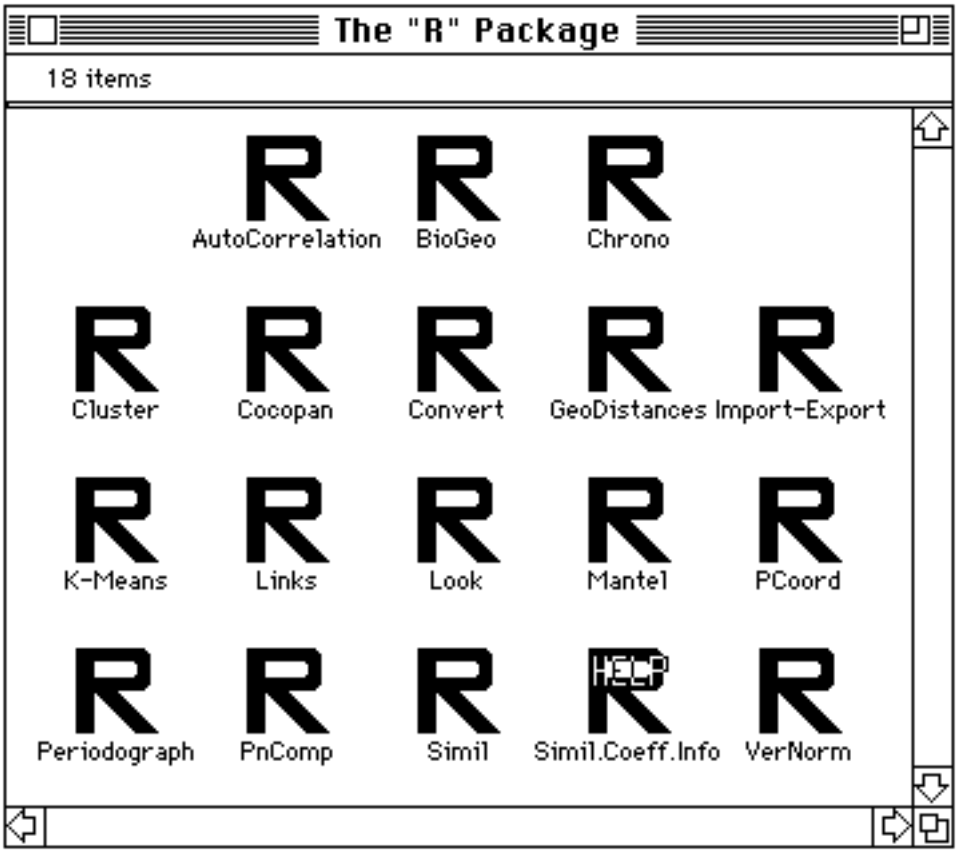
The R Package:

Multidimensional analysis, spatial analysis

CMS (IBM), VMS (VAX) and Macintosh versions

Pierre Legendre / Alain Vaudor



Université de Montréal

The R Package:

Multidimensional analysis, spatial analysis

CMS (IBM), VMS (VAX) and Macintosh versions

Pierre Legendre and **Alain Vaudor**

Département de sciences biologiques
Université de Montréal
C.P. 6128, Succursale A
Montréal, Québec
Canada H3C 3J7

Electronic mail — P. Legendre: Legendre @ Ere.UMontreal.CA
A. Vaudor: Vaudor @ Ere.UMontreal.CA

This manual was prepared with the editorial assistance of
Chantal Ouimet, **François-Joseph Lapointe** and **Gilles Lavoie**

Université de Montréal, September 1991
Update:

Disclaimer

These programs are provided without any explicit or implicit warranty of correct functioning. They have been developed as part of a university-based research program. If, however, you should encounter problems with one or another of the programs in this package, we will be happy to help solve them (see section 5, page 6). Researchers can use these programs for scientific purposes, but the source code remains the property of the authors of this manual.

The following fonts must be available to print this document: *Times*, *Courier* and *Symbol*. Pages were formatted for a PostScript laser printer.

Proper reference of this manual:

Legendre, P. and A. Vaudor. 1991. The R Package: Multidimensional analysis, spatial analysis. Département de sciences biologiques, Université de Montréal. iv + 142 p.

Table of Contents

HISTORICAL NOTES

This package of computer programs has been written along the years by Alain Vaudor (Computer analyst) and Pierre Legendre. The development of the package started in 1978, at the Université du Québec à Montréal, on PDP-10 and CDC/CYBER equipments. In 1980, the package moved with us to Université de Montréal, where its development has continued since. General-purpose methods of data analysis were implemented first (similarity and distance computations, various types of clustering methods, ordination procedures, etc., plus several utility programs), followed by more specific methods for time-series and spatial analysis (contingency periodogram, chronological clustering, space-constrained clustering, spatial autocorrelation, Mantel tests, Cocopan). The programs were gradually made more reliable and user-friendly, with the help of successive generations of graduate students and other users.

The first IBM versions were developed independently at the University of Waterloo (Ontario) and the Université de Sherbrooke (Québec), for batch use only. The IBM conversational version was developed by P. Legendre since 1985, first on the C.N.U.S.C. machines in Montpellier (France) and at the Department of Ecology and Evolution, State University of New York (Stony Brook, U.S.A.), then on the mainframe of École Polytechnique de Montréal. That version was adapted to the VAX at the Université de Montréal in 1989. The programs became bilingual (French/English) at the time of the Stony Brook implementation. It took 13 years to complete the development of this package and write the present documentation; this includes the time needed to develop several of the methods now available through this package, and to write the concomitant publications.

The programming language is PASCAL, while the calling programs for IBM mainframes are in REXX, and in DCL for VAX machines. The package has been distributed to a number of institutions in North and South America and in Europe. The presently available versions are:

Type of computer	Conversation language	Operating system	Calling programs
IBM mainframes	English or French	VM/CMS	EXEC files (REXX)
VAX	English or French	VAX/VMS	DCL files
Apple Macintosh	English or French		Click the mouse!

The programs can be obtained against 25 $ (Canadian, U.S., or Australian), which covers the costs of the diskette, photocopy, and postage. Specify the version you wish to receive; for the CMS and VMS versions, indicate whether you wish to receive a diskette for Macintosh or for an MS/DOS machine (if you prefer 5.25-inch diskettes, please say so). A paper copy of the documentation will also be sent; specify the language (English or French). Single programs can be sent by electronic mail. The Macintosh version is distributed already compiled, while the mainframe versions are sent as PASCAL source files; this makes it easy for the users to change the size of the matrices that can be analyzed, as well as the conversation language. The implication is however that the users must compile the programs themselves before they can be used (PASCALVS or VSPASCAL compilers for IBM; PASCAL compiler for VAX).

The name of the package, "R", comes from our 1978 work on PDP-10 equipment, where R (for *Run*) was the command to start a program. In the CDC implementation, we kept "R" as the name of the main calling program written in CCL, which is the equivalent of a large IBM EXEC file, thus mimicking the PDP way of calling the programs. This is why "R" has gradually shifted to become the pleasantly short name of the package.

ACCESS TO THE PROGRAMS

1. In conversational mode, CMS operating system (IBM)

In order to use these program from his own virtual machine, the user must first get access to the minidisk where the "R" EXEC and program files are stored, unless he is already working on the virtual machine that contain these files.

Write down HERE the necessary commands for your computer:

The EXEC command files that are available are the following. Each one starts the execution of the corresponding program.

* **AUTOCOR**
* **BIOGEO**
* **CHRONO**
* **COCOPAN**
* **CONVERT**
* **DIST**
* **EXPNTS**
* **EXPORT**
* **IMPORT**
* **INTERLNK**
* **KMEANS**
* **LANCE**
* **LOOK**
* **MANTEL**
* **PCOORD**
* **PERIOD**
* **SIMIL**
* **VERNORM**

These command files start the following programs:

* **AUTOCOR**: Univariate spatial autocorrelation analysis (Moran's *I* and Geary's *c* coefficients). This program also allows to compute a list of spatial link edges among localities, according to a variety of algorithms. Such lists are used by BioGeo, KMeans (with constraint) and Cocopan.
* **BIOGEO**: Clustering with constraint of spatial contiguity. Method: proportional-link linkage.
* **CHRONO**: Chronological clustering (with temporal, or one-dimensional spatial contiguity constraint).
* **COCOPAN**: Analysis of variance for spatially autocorrelated data.
* **CONVERT**: Converts **S**imilarities into **D**istances, or **D**istances into **S**imilarities.
* **DIST**: Computes distances along earth's curvature, from longitude and latitude data.
* **EXPNTS**: Converts a SIMIL-type binary matrix into a NT-SYS-type binary matrix (NT-SYS is F. James Rohlf's Numerical Taxonomy and Multivariate Analysis System).
* **EXPORT**: Converts a SIMIL-type binary matrix into a square ASCII matrix.
* **IMPORT**: Converts a square ASCII matrix into a SIMIL-type binary matrix.
* **INTERLNK**: Proportional-link linkage clustering.
* **K-MEANS**: K-Means (or minimum-variance) clustering, with or without spatial contiguity constraint.
* **LANCE**: Lance & Williams clustering algorithm, including Ward's method.
* **LOOK**: To look at a binary file computed by SIMIL.
* **MANTEL**: Mantel test, partial Mantel tests, multivariate correlogram.
* **PCOORD**: Principal coordinates analysis.
* **PERIOD**: Periodic analysis using the contingency periodogram.
* **SIMIL**: 50 measures of resemblance. SIMIL computes coefficients among the ROWS of the input file only. For Q-mode coefficients, the rows of the data matrix must correspond to the objects; in the R-mode, the rows must correspond to descriptors.
* **VERNORM**: To verify and normalize the columns (variables) of a data file.

Depending on the dimensions given to their parameters, some programs may require, to run, more memory than the default amount attributed to users. This problem may arise for instance when the dimensions have been substantially increased to analyze particularly large data sets. One has to use command DEF STOR to get access to additional memory space.

When running conversational programs, the text written on the screen by the programs, as well as the user's answers, usually appear only on the screen and are not copied into a file for later reference. Furthermore, and contrary to the other programs of this package that produce an output file, CHRONO, MANTEL and PERIOD display the computations' results on the screen only. If one wishes to save these questions, answers, and computation results in a file, for further reference or printing, one has to give the following command before running the programs:

```
CP SPOOL CONS START TO *
```

This command has to be given *outside of any FILELIST.* Or, that command may have been written in advance in an EXEC file (call it REMEMBER EXEC, for instance). After running one or several programs, and again *outside of any FILELIST*, type:

```
CLOSE CONS NAME SCREEN MEMORY
CP SPOOL CONS STOP
```

(these commands may also have been written in an EXEC file). The file containing the dialogue and results, to which the name *SCREEN MEMORY* had been given in the example above, is now found in the "Reader list", to which one can access using command RDRL. That file may be edited to remove useless sections before it is printed.

2. In conversational mode, VMS operating system (VAX)

On VAX machines, the programs are started by DCL command files which are equivalent to the EXEC files of the IBM CMS system. They are called VERNORM.COM, SIMIL.COM, etc. The user who has a copy of the "R" package on his own account number may call the programs directly by typing @ followed by the name of the program she wishes to run; for example: @VERNORM, @SIMIL, etc.

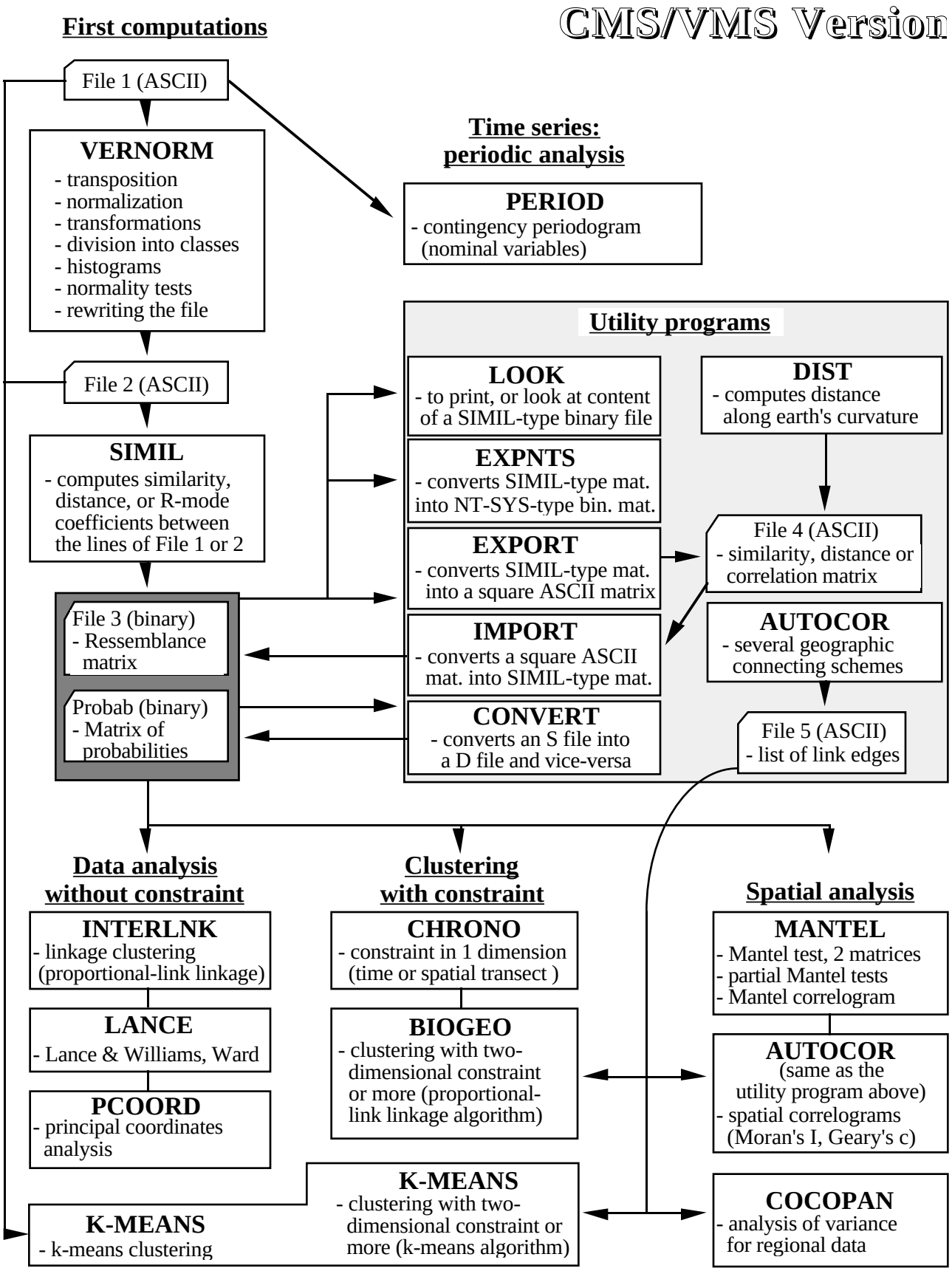
A second possibility is to call the R.COM command file by typing @R. That file first gives the names and address of the authors of the package, followed by a list of the programs available. The user may then call any one program by typing its name directly, without symbol @; for example: VERNORM, SIMIL, etc.

The package may be installed in such a way that it is accessible to the other users of the VAX. The person in charge of the package has to modify all the command files, including the R.COM file, adding his own machine-address everywhere a program or another command file is called (by RUN or by @). For example:

```
@VERNORM        may become      @DUA1:[JohnDoe]VERNORM
RUN SIMIL       may become      @DUA1:[JohnDoe]SIMIL
```

He may then ask each potential user to add in his LOGIN.COM file an instruction like:

```
$ R:=="@DUA1:[JohnDoe]R.COM"
```


CMS/VMS Version
First computations
File 1 (ASCII)
VERNORM
- transposition
- normalization
- transformations
- division into classes
- histograms
- normality tests
- rewriting the file
File 2 (ASCII)
SIMIL
- computes similarity, distance, or R-mode coefficients between the lines of File 1 or 2
File 3 (binary)
- Ressemblance matrix
Probab (binary)
- Matrix of probabilities
Time series: periodic analysis
PERIOD
- contingency periodogram (nominal variables)
Utility programs
LOOK
- to print, or look at content of a SIMIL-type binary file
EXPNTS
- converts SIMIL-type mat. into NT-SYS-type bin. mat.
EXPORT
- converts SIMIL-type mat. into a square ASCII matrix
IMPORT
- converts a square ASCII mat. into SIMIL-type mat.
CONVERT
- converts an S file into a D file and vice-versa
DIST
- computes distance along earth's curvature
File 4 (ASCII)
- similarity, distance or correlation matrix
AUTOCOR
- several geographic connecting schemes
File 5 (ASCII)
- list of link edges
Data analysis without constraint
INTERLNK
- linkage clustering (proportional-link linkage)
LANCE
- Lance & Williams, Ward
PCOORD
- principal coordinates analysis
K-MEANS
- k-means clustering
Clustering with constraint
CHRONO
- constraint in 1 dimension (time or spatial transect)
BIOGEO
- clustering with two-dimensional constraint or more (proportional-link linkage algorithm)
K-MEANS
- clustering with two-dimensional constraint or more (k-means algorithm)
Spatial analysis
MANTEL
- Mantel test, 2 matrices
- partial Mantel tests
- Mantel correlogram
AUTOCOR
(same as the utility program above)
- spatial correlograms (Moran's I, Geary's c)
COCOPAN
- analysis of variance for regional data

Using this new command LOGIN file, the user only has to type

```
R
```

to get access to the programs and receive the welcoming message. Thereafter, for the ongoing VAX session, each program may be called by typing its name only.

3. In batch mode, CMS operating system (IBM)

To run a program in batch mode, the names of the various input and output files have to be spelled out in the program's EXEC file. Answers to the questions of the program, usually found after the informative message "EXECUTION BEGINS ...", must have been written beforehand in a file whose name appears in the EXEC file.

Four program sometimes have to be used in batch mode, to analyze very large data files; they are SIMIL, MANTEL, AUTOCOR and PCOORD. The corresponding EXEC files (SIMILBAT, MANTBAT, AUTOBAT, PCOORBAT) may reside on the RPACKAGE machine where the package is stored. For batch use, you must copy the EXEC file you wish to use onto your own machine, and make the necessary adaptations (file names). A batch run is started in the usual way; for instance:

```
SUBMIT SIMILBAT (CPU ...
```

Example: SIMILBAT EXEC file — /* These lines are comments */

```
/* SIMIL batch run file  */
GLOBAL TXTLIB VSPASCAL
FI OUTPUT PRINTER
/* Name of the file containing the answers to the program's questions: */
FI INPUT DISK answers simil a
/* Name of the input data file: */
FI ENTREEC DISK myfile data a
/* Name of the output file for the resemblance matrix computed by SIMIL: */
FI SORTIE DISK myfile s17 a
/* Name of the file containing the partial similarity matrices: */
FI PART DISK myfile partial a
/* Name of the file that will receive the matrix of probabilities, if any: */
FI PROBAB DISK myfile probab a
/* Start the run the larger version of the program, called SIMILBAT here: */
"LOAD SIMILBAT (START"
/* If necessary, change the name of the virtual machine: */
"SENDFILE myfile s17 a    TO RPACKAGE"
"SENDFILE myfile probab a TO RPACKAGE"
"SENDFILE myfile probab a TO RPACKAGE"
```

The names of the various files must, of course, correspond to your data. The file of answers must contain only the answers to the questions asked by the program for that very run.

Example of a file of answers to the questions of program SIMIL:

```
A title of your choice.
380       [number of rows, or blocs of rows]
109       [number of columns]
```

`N` *[there are no object names in columns 1-10]*
`S01` *[code for the similarity coefficient to be computed]*
`5` *[a value of "1" will be given to any data larger than or equal to 5]*

An easy way to get the list of questions whose answers have to be included in that file is to make a conversational run using a small portion of the input data.

4. Macintosh version

In the Macintosh version, the programs are essentially the same as in the mainframe versions. In a few cases, rearrangements have been made to take best advantage of the Macintosh user's interface. The available programs are the following:

* **AUTOCORRELATION**: Univariate spatial autocorrelation analysis (Moran's *I* and Geary's *c* coefficients).
* **BIOGEO**: Clustering with constraint of spatial contiguity. Method: proportional-link linkage.
* **CHRONO**: Chronological clustering (with temporal, or one-dimensional spatial contiguity constraint).
* **CLUSTER**: Proportional-link linkage and Lance & Williams algorithms, including Ward's method (replaces LANCE and INTERLNK in the CMS and VMS versions).
* **COCOPAN**: Analysis of variance for spatially autocorrelated data.
* **CONVERT**: Converts **S**imilarities into **D**istances, or **D**istances into **S**imilarities.
* **GEOGRAPHIC DISTANCES**: Computes distances along earth's curvature, from longitude and latitude data.
* **IMPORT-EXPORT**: Converts a SIMIL-type binary matrix into a square ASCII matrix, or a square ASCII matrix into a SIMIL-type binary matrix. Replaces IMPORT and EXPORT of the mainframe versions.
* **K-MEANS**: K-Means (or minimum-variance) clustering, with or without spatial contiguity constraint.
* **LINKS**: Computes a list of spatial link edges among localities, following a variety of algorithms. Such lists are used by BioGeo, KMeans (with constraint), Autocorrelation and Cocopan.
* **LOOK**: To look at a binary file computed by SIMIL.
* **MANTEL**: Mantel test, partial Mantel tests, multivariate correlogram.
* **PCOORD**: Principal coordinates analysis.
* **PERIODOGRAPH**: Periodic analysis using the contingency periodogram.
* **PNCOMP**: Principal components analysis.
* **SIMIL**: 50 measures of resemblance. SIMIL computes coefficients among the ROWS of the input file only. For Q-mode coefficients, the rows of the data matrix must correspond to the objects; in the R-mode, the rows must correspond to descriptors.
* **VERNORM**: To verify and normalize the columns (variables) of a data file.

For routine work, it is preferable to transfer the programs to a hard disk, or to use two diskettes. Make sure that your working environment includes a SYSTEM FILE, an icon corresponding to the type of printer you are using, as well as an ASCII file editor (reason given below).

If you want to use the printer — to print a dendrogram for instance, make sure that the diskette where the System File is found includes at least 30 to 50K of free space; the System needs that much space to create temporary files during printing.

The input data files must be rectangular matrices of real or integer numbers; their type must be "text only" (ASCII files). They can be extracted in "text only" mode from spreadsheets or word processors, or from statistical programs; or, they can be typed directly using an ASCII file or programming editor (Edit, TeachText, and so on). Files transferred from mainframes by MODEM are

usually in the “text only” type.

To select the input data file for a program, one simply has to click on “Open” after clicking on the name of the file, when the file menu is presented. Only the names of the files that are of a type appropriate for the program being run are presented: “text only” files as input for VERNORM, SIMIL, PERIODOGRAPH or IMPORT-EXPORT (depending on the option); SIMIL-type binary files for IMPORT-EXPORT (depending on the option) and most of the other programs.

A default name is always proposed by the programs for their output files; that name can be changed at will, after which the user clicks on “Save”. Many of the data analysis programs offer “Printer” as the default option for the output. If you agree, you click “Save”; if not, you change the name and give a file name of your choice, before clicking “Save”. The output file thus created can be read back and edited using an ASCII file or programming editor.

In the raw data files, or when it is necessary to provide numbers as answers to questions of a program, one has to remember that the programs are written in PASCAL; it is a requirement of PASCAL to write “0.5” and not “.5”, for instance. That recommendation also applies to the CMS and VMS versions. In the Macintosh versions, the programs numbered 3.0 and more are freed from this constraint and can read correctly data such as .2, -.57, +0.1, -0., 5E+2, +1.0e-8, etc.

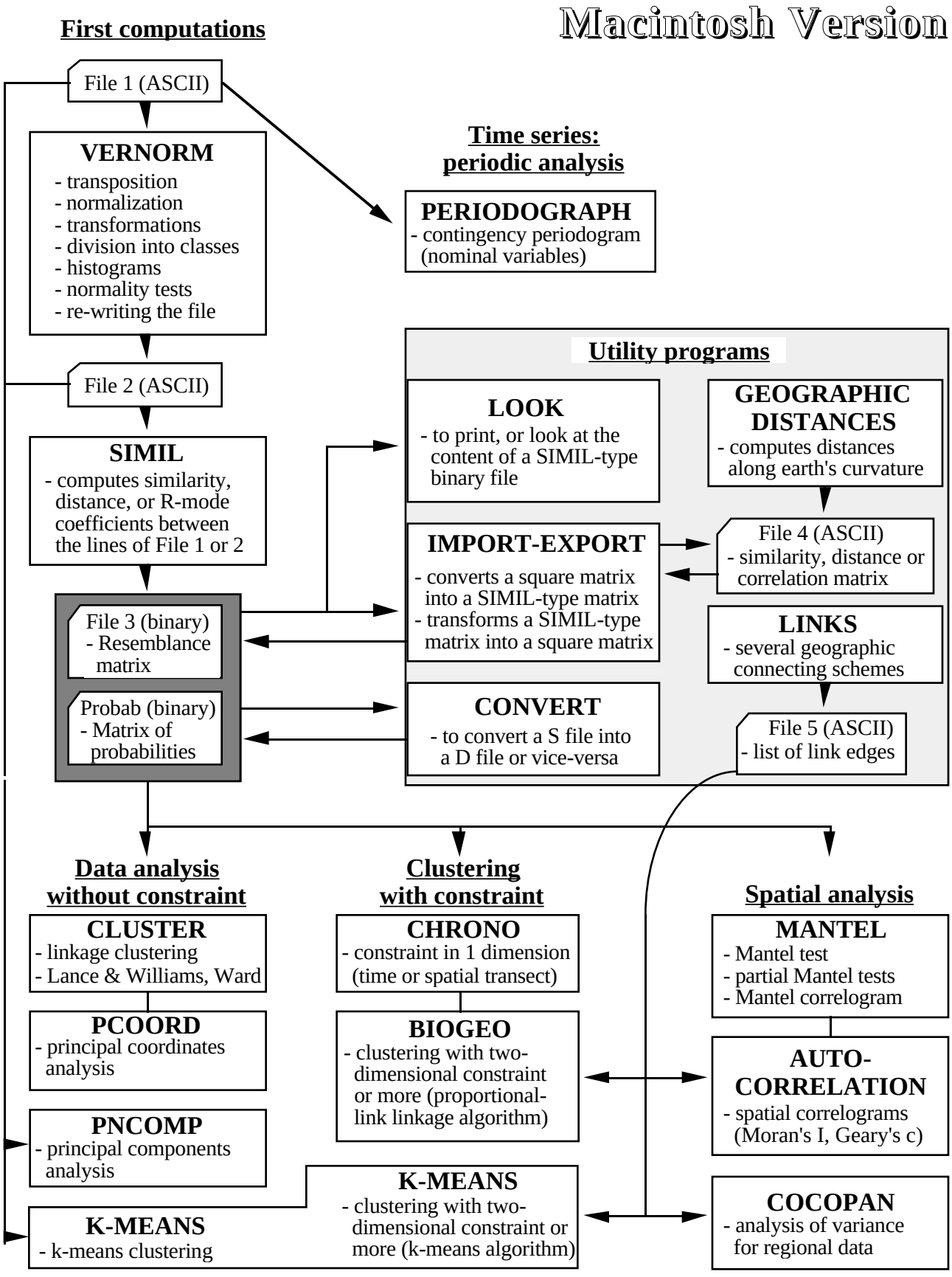
5. Documenting a problem

These programs are provided without any explicit or implicit warranty of correct functioning. They have been developed as part of a university-based research program. If, however, you should encounter problems with one or another of the programs in this package, we will be happy to help solve them and, at the same time, to solve the problem for all the users of the “R” package. For that reason, it is important to provide us with a maximum of information, including the following:

- The version of the program that you are using (see the “Info” window); give us also the date of the program (same “Info” window), or the date where you got it.
- The input file(s) that you have been using; in many instances, the problems brought to us are simply due to structure (no blanks between data, etc.) or content (illegal characters, etc.) mistakes in the input data files. On Macintosh, the binary SIMIL-type files can be compacted using BINHEX or STUFFIT, and sent by electronic mail. On IBM mainframes, binary SIMIL-type files can be sent directly by E-mail.
- The output file(s), including messages that may be found in those files.
- Any other message that you got on the screen.

Please send these information to Alain Vaudor *via* E-mail, using the address on the cover of this manual; or, if unavailable, by regular mail (paper and/or diskette).

If you wish to install these programs on machines other than those for which they have been tested, you will have to check in detail that they run correctly and produce correct results. Dialect differences are found among PASCAL compilers; differences in machine word lengths, and in the minimum or maximum values that real numbers can take, are to be watched carefully for potential problems.


Macintosh Version
First computations
File 1 (ASCII)
VERNORM
- transposition
- normalization
- transformations
- division into classes
- histograms
- normality tests
- re-writing the file
File 2 (ASCII)
SIMIL
- computes similarity, distance, or R-mode coefficients between the lines of File 1 or 2
File 3 (binary)
- Resemblance matrix
Probab (binary)
- Matrix of probabilities
Time series: periodic analysis
PERIODOGRAPH
- contingency periodogram (nominal variables)
Utility programs
LOOK
- to print, or look at the content of a SIMIL-type binary file
GEOGRAPHIC DISTANCES
- computes distances along earth's curvature
IMPORT-EXPORT
- converts a square matrix into a SIMIL-type matrix
- transforms a SIMIL-type matrix into a square matrix
File 4 (ASCII)
- similarity, distance or correlation matrix
LINKS
- several geographic connecting schemes
CONVERT
- to convert a S file into a D file or vice-versa
File 5 (ASCII)
- list of link edges
Data analysis without constraint
CLUSTER
- linkage clustering
- Lance & Williams, Ward
PCOORD
- principal coordinates analysis
PNCOMP
- principal components analysis
K-MEANS
- k-means clustering
Clustering with constraint
CHRONO
- constraint in 1 dimension (time or spatial transect)
BIOGEO
- clustering with two-dimensional constraint or more (proportional-link linkage algorithm)
K-MEANS
- clustering with two-dimensional constraint or more (k-means algorithm)
Spatial analysis
MANTEL
- Mantel test
- partial Mantel tests
- Mantel correlogram
AUTO-CORRELATION
- spatial correlograms (Moran's I, Geary's c)
COCOPAN
- analysis of variance for regional data

DESCRIPTION OF THE PROGRAMS

*AUTOCORRELATION*Macintosh *or AUTOCOR*CMS/VMS

What does AUTOCORRELATION do ?

Program AUTOCORRELATION analyzes the spatial autocorrelation structure of a variable by producing structure functions, called correlograms, following various schemes of connections or distances among the data locations. This is a strictly univariate method; program MANTEL makes it possible to produce a correlogram from multivariate data. For quantitative data, autocorrelation is measured using Moran's I and Geary's c indices. For ordinal or nominal data, standard normal deviates (*S.N.D.*'s) are computed for each distance class. Each value is accompanied by the probability of it not being significantly different from zero (one-tailed test). How to interpret correlograms is discussed by Legendre & Fortin (1989).

Versions CMS and VMS of this program may be used to produce a list of neighboring locations, either on a regular grid (following different connecting schemes), a Delaunay triangulation, or a Gabriel graph. The file containing the list of link edges may be used as constraint for programs BIOGEO or KMEANS, or for any other program that requires such a list, such as COCOPAN. In the Macintosh version, the production of that list of link edges has been put in a different program called LINKS. Finally, AUTOCORRELATION may also produce a file containing an upper triangular matrix of distance classes among localities; that file, whose default name is CLASSEF, is required by program MANTEL to compute a multivariate correlogram.

Input and output files

There are many questions in the program to describe the input and output files; this reflects the many options offered by the program. Read the questions carefully. The program requires information relative to (a) the value of the variable at each location, and (b) the position of the points with respect to one another. There are **five types of input files** in the CMS and VMS versions. In the Macintosh version, file type (2) is not allowed because the function that computes the list of link edges and writes it onto a file has been transferred to the new program LINKS.

(1) List of data values (Z)

This type of input file only contains the values of the variable, called Z here. These values are real numbers, or strictly positive integers in the case of a nominal variable. In that file, the values may be written one after the other, separated by one or several spaces, following the order of the points, but without object names or other identifiers; the program assumes that the first value in the list corresponds to locality 1, and so on. The list is written from left to right, using successive lines as needed, as one writes a page of text. The user may also write only one value per line if she so wishes. That file type is the only one accepted by the Macintosh version to provide the values of the variable; its length is limited to 16000 observation points. In the CMS and VMS versions, that file type is used only in the case of a regular grid of observations; the available connecting scheme are named after the game of chess (Upton & Fingleton, 1985, ch. 3): rook's (horizontal and vertical links only), bishop's (diagonal links only), or queen's connections (combination of the rook's and bishop's moves).

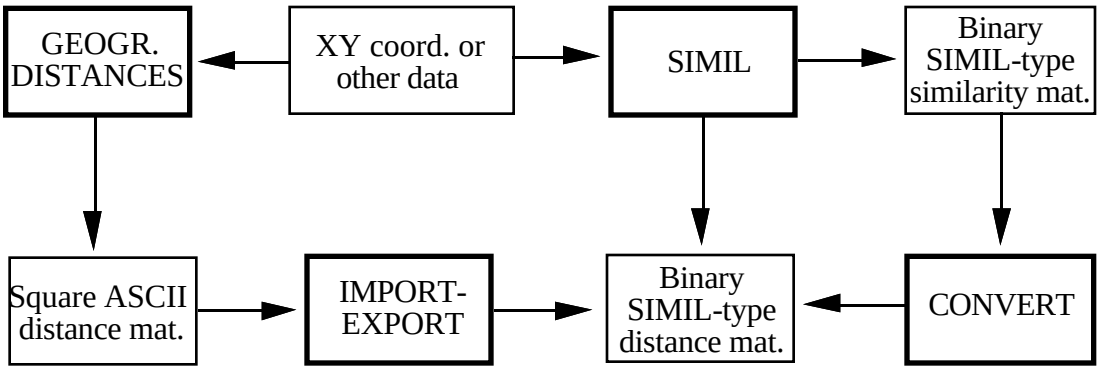
(2) List of coordinates (X, Y) and of values (Z)

In the CMS and VMS versions, when the points do not form a regular grid, the location coordinates are provided in the same file as the variable values. Thus each data line must contain three pieces of information:

X coordinate Y coordinate Value of the variable

The coordinates are presented as integer or real numbers (with decimals), but not in the form of degrees-minutes-seconds. The data are read in free format, so that it is not necessary to type the values in preassigned columns. As with the other CMS and VMS programs of this package, one has to write "0.37" for instance instead of ".37", which is a requirement of the PASCAL language; see also p. 6.

(3) Distance matrix file



The relative positions of the localities may be described by a binary SIMIL-type matrix of geographic distances computed by SIMIL, or transferred from another program by IMPORT (in the CMS or VMS versions) or IMPORT-EXPORT (Macintosh version). The program assumes in all cases that the SIMIL-type matrix contains distances and **not** similarities. A similarity matrix may easily be converted into distances using program CONVERT. The initial data file from which SIMIL computes the Euclidean distances (D01) must contain the following pieces of information only:

X coordinate Y coordinate

The coordinates are presented as integer or real numbers (with decimals), but not in the form of degrees-minutes-seconds. The data are read in free format. The interest of this type of file is that the user may decide to use another distance function than the Euclidean (geographic) distance among objects. Program GEOGRAPHIC DISTANCES (DIST in the CMS and VMS versions) may also be used here to compute distances following earth's curvature; these distances come out in the form of a square ASCII-type matrix, which can easily be converted to the SIMIL format using IMPORT (in the CMS and VMS versions) or IMPORT-EXPORT (in the Macintosh version).

(4) Matrix of distance classes among objects

This ASCII file may represent the whole square matrix of distances already divided into classes, or else the upper triangular part only of that distance matrix, in which case it is similar to file CLASSEF (file type 8, below) produced by this program. Distance classes are numbered by the successive integer numbers, starting with 1. The size of that matrix is $n \times n$, where n is the number of localities. Allowing a square matrix makes it possible for the user to use a non-symmetrical matrix of distance classes, where the distance from *a* to *b* is not the same as the distance from *b* to *a*.

(5) List of link edges among objects

This ASCII file provides the program with a list of connecting edges among neighboring localities. Each link edge is represented by a pair of object numbers, written in free format and separated by at least one space. That file may have been produced either by AUTOCOR (in the CMS and VMS versions) or by LINKS (in the Macintosh version). The following example corresponds to

the rook's connection scheme among 12 localities forming a regular grid of 3 rows and 4 columns; each link edge is represented by a pair of numbers:

```
 1  2     2  3     3  4     5  6     6  7     7  8     9 10    10 11
11 12     5  1     6  2     7  3     8  4     9  5    10  6    11  7
12  8
```

Three types of output files may be created by this program:

(6) Results file containing the correlogram statistics

The default name of that file in the CMS and VMS versions is "RESULT CORR A". This ASCII file is presented in a different way depending on the analysis being for a quantitative or a nominal variable. An example for each type is presented hereunder. For the analysis of quantitative data, Moran's *I* and Geary's *c* indices are computed for each distance class *d* (Cliff & Ord, 1981):

$$I(d) = [n \sum\sum w_{ij}(y_i - y\text{-}bar)(y_j - y\text{-}bar)]/[W \sum(y_i - y\text{-}bar)^2] \qquad \text{for } i \neq j$$

$$c(d) = [(n\text{-}1) \sum\sum w_{ij}(y_i - y_j)^2]/[2W \sum(y_i - y\text{-}bar)^2] \qquad \text{for } i \neq j$$

The values of the variable are the *y*'s, while *y-bar* is the mean of these values. The w_{ij}'s take the value 1 when the pair (i,j) pertains to distance class *d* (the one for which the coefficient is computed), and 0 otherwise. *W* is the sum of the w_{ij}'s, or in other words the number of pairs, in the whole square matrix of distances among points, that are taken into account when computing the coefficient for the given distance class. Moran's coefficient varies generally from -1 to +1, although sometimes it can exceed -1 or +1; positive values of Moran's *I* correspond to positive autocorrelation. Geary's coefficient varies from 0 to some indeterminate positive value which rarely exceeds 3 in real cases; values of *c* smaller than 1 correspond to positive autocorrelation.

These coefficients are computed for each distance class *d*; each value is accompanied by the probability of it not being significantly different from zero (one-tailed test). Formulas for the computation of the standard error of these statistics are found in Cliff & Ord (1981), Sokal & Oden (1978) and Legendre & Legendre (1984a). The hypotheses are the following:

H_0: there is no spatial autocorrelation. The values of the variable are spatially independent. Each value of the *I* coefficient is equal to $E(I) = -(n - 1)^{-1} \approx 0$, where $E(I)$ is the expectation of *I* while *n* is the number of data points; each value of the *c* coefficient equals $E(c) = 1$.

H_1: there is significant spatial autocorrelation. The values of the variable are spatially dependent. The value of the *I* coefficient is significantly different from $E(I) = -(n - 1)^{-1} \approx 0$; the value of *c* is significantly different from $E(c) = 1$.

As recommended by Oden (1984), one should use a Bonferroni correction to check whether the correlogram contains at least one value which is significant. That correction consists of using a significance level *alpha*' = *alpha*/(number of simultaneous tests); for example, a correlogram with 5 distance classes can be declared significant only if it contains values that remain significant at the corrected level *alpha*' = 0.05/5 = 0.01.

Here is an example of a file of results for quantitative data, computed by the Macintosh version of the program; the output file of the CMS and VMS versions is almost identical. The corresponding correlogram has been published as Figure 3 of Legendre & Troussellier (1988).

```
 P R O G R A M   A u t o C o r r e l a t i o n
```

```
Author: A. Vaudor


Distance matrix:
    INPUT FILE: XY, Thau
    TITLE: Geographic distances, Thau (63 stations)
    DATE: 10/8/88
    FUNCTION: D01
    Number of objects : 63
    Number of descriptors :  2

Classes with equal width

    Class  Upper bound   Freq.                  [frequency histogram data,
        1   1.00518        97                        from the triangular
        2   2.01036       162                           distance matrix]
        3   3.01553       250
      etc.            etc.
       17  17.08802         4

Input data file: CHLAtr

OPTIONS:   Binary SIMIL matrix
NOTE: The most significant probabilities are close to zero.
      Probabilities are computed with a precision of   0.00100

H0:             I = 0    I = 0            C = 1    C = 1
H1:             I > 0    I < 0            C < 1    C > 1
Dist.,I(Moran), p(H0),   p(H0), C(Geary),  p(H0),   p(H0),  Card.
  1   0.4646    0.000            0.3355    0.000            194
  2   0.3833    0.000            0.4151    0.000            324
  3   0.3284    0.000            0.5352    0.000            500
  4   0.3382    0.000            0.5280    0.000            450
  5   0.2251    0.000            0.6708    0.000            484
  6   0.0773    0.101            0.8055    0.018            336
  7  -0.1109             0.121   1.0151              0.373   280
  8  -0.1992             0.011   1.1111              0.085   288
  9  -0.3517             0.000   1.3626              0.000   274
 10  -0.5869             0.000   1.7343              0.000   222
 11  -0.6228             0.000   1.8906              0.000   154
 12  -0.8550             0.000   2.2102              0.000   138
 13  -0.7459             0.000   2.4051              0.000   120
 14  -0.8355             0.000   2.5375              0.000    68
 15  -0.6122             0.001   2.4070              0.000    48
 16  -0.6631             0.023   2.4416              0.003    18
 17  -1.4980             0.001   3.3191              0.002     8

Total                                                       3906
```

Moran's I value is found in column 2, and Geary's c in column 5, for the various distance classes (column 1). Probabilities of the one-tailed tests for Moran's I are in columns 3 and 4; they are presented as two separate columns, depending on the coefficient value being positive or negative, in order to read them more easily. The same goes for the probabilities associated with Geary's c. One-tailed hypotheses H_0 and H_1 are specified above these columns. The number of pairs of points in each distance class (cardinality) is shown in the right-hand column. Each value is twice the value given in the frequency histogram; this is the value we would obtain if we were working in a square matrix (excluding the diagonal), and not in a triangular matrix.

In the Macintosh version, the program draws the correlograms on the screen, from where they can either be printed, or saved as PICT files. A correlogram is a graph where autocorrelation values are plotted in ordinate, against distance classes *d* among localities in abscissa; see for instance Figure 11.22 of Legendre & Legendre (1984a). See also Legendre & Fortin (1989) for the ways of interpreting spatial autocorrelograms.

For nominal (qualitative) data, or for ordinal (semi-quantitative) data treated as nominal, the program computes for each distance class the standard normal deviates (*S.N.D.*'s) as well as the associated probabilities, for each distance class and for each pair of states of the variable. The theory behind these computations can be found in Sokal & Oden (1978), Cliff & Ord (1981) or Upton & Fingleton (1985). Here is an example of the results file for nominal data with 4 classes, computed with the CMS version of the program. Few comparisons are significant in this example.

```
S P A T I A L   A U T O C O R R E L A T I O N

for quantitative or nominal data.
Version IBM 2.0B
Author: Alain VAUDOR

Movement option:   13

NOTE: The most significant probabilities are close to zero
The probabilities are printed with precision   0.00100

H0:                                S.N.D.=0,    S.N.D.=0
H1:                                S.N.D.>0,    S.N.D.<0
          CLASSES     S.N.D.          P(H0) ,       P(H0),  PAIRS
DISTANCE  1                                                  312
           [1][1]     -0.272                         0.434
           [1][2]     -0.522                         0.301
           [1][3]     -1.068                         0.143
           [1][4]     -0.408                         0.342
           [2][2]      1.721          0.052
           [2][3]      0.889          0.187
           [2][4]     -1.687                         0.046
           [3][3] CARD. CLASS [3]/NOBJ < 0.2 or > 0.8
           [3][4]     -1.523                         0.064
           [4][4]      3.047          0.004
    [Total diff.]     -2.821                         0.002

DISTANCE  2                                                  586
           [1][1]     -2.204                         0.007
           [1][2]     -1.822                         0.034
           [1][3]     -0.246                         0.403
           [1][4]      2.001          0.023
           [2][2]      1.510          0.069
           [2][3]      1.485          0.069
           [2][4]      0.348          0.364
           [3][3] CARD. CLASS [3]/NOBJ < 0.2 or > 0.8
           [3][4]     -2.406                         0.008
           [4][4]     -0.082                         0.495
    [Total diff.]     -0.056                         0.478

DISTANCE  3                                                  732
           etc.
```

```
DISTANCE  4                                                    716
           etc.

DISTANCE  5                                                    544
           etc.

DISTANCE  6                                                    254
           etc.

DISTANCE  7                                                     48
           etc.

TOTAL                                                         3192
```

(7) File of link edges

Only versions CMS and VMS of the program can compute the file of link edges, which receives the default name of "LINKS DATA A". This ASCII file contains a list of pairs of locations that are connected, according to the connecting scheme (options 1 to 13) used when running the program. That file may be used as constraint for programs BIOGEO or KMEANS, or for any other program that requires such a list, such as COCOPAN. In the Macintosh version, the production of that list of link edges has been put in a different program called LINKS. An example of such a file is shown in (5) above. Notice that the user may edit that ASCII file, adding or removing link edges, depending on the requirements and hypotheses of his study.

(8) Half-matrix of distance classes (CLASSEF)

That file, whose default name is "CLASSEF DATA A" in the CMS and VMS versions, contains an upper triangular matrix of distance classes among localities. It is required by program MANTEL to compute a multivariate correlogram; see that program name.

Options of the program

Versions CMS and VMS offer 16 computation options, numbered 0 to 15 (see the example below). In the Macintosh version, options 0, 14 and 15 only are found. These options can be described under the five headings that follow, which depend on the input data files of the run.

(1) Option 0 — Distance matrix from SIMIL

Two input files are necessary for that option. The first one is the List of data values (file of type 1), while the second one is the Distance matrix file (file of type 3) computed using the distance coefficient chosen by the user (see Table 4 for the coefficients available in program SIMIL). This option does not require the points to form a regular grid. The following questions are presented to the user:

- "Equidistant (0) or equal frequency (1) classes ?" — One cannot get both. The intervals of equidistant classes are of the same width; equal frequency classes all contain the same number of pairs, except for ties (equal distances) which may force some classes to contain more pairs.

- "Number of classes ?" — The user must specify how many classes she wants.

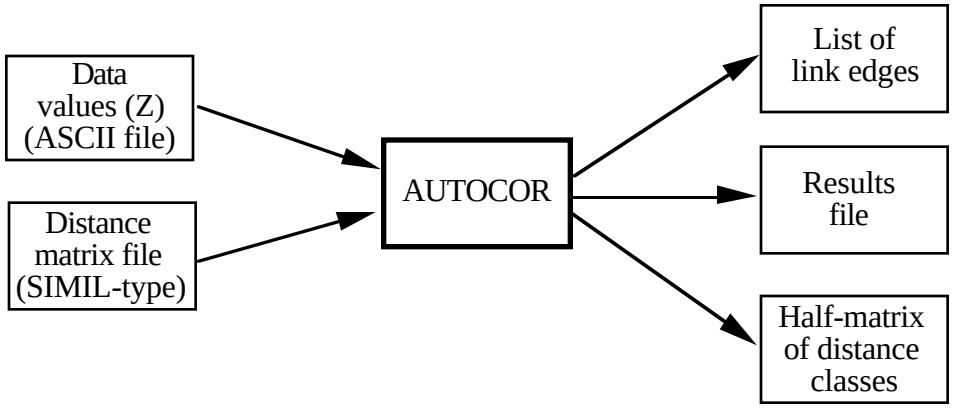
- "Do you want to see the histogram ?" — An histogram makes it easier to appreciate the shape of the frequency distribution.

- “Would you prefer a different number/type of classes ?” — One can get back to the first two questions to change the division into classes.

- “Do you wish to have the CLASSEF file of distance classes written out, for Mantel correlogram ?” — See file type (8) above, where this matrix is briefly described.

- “Do you wish to have the list of link edges written on file "LINK" ?” — See output file type (5) above, as well as output file type (7), where that list is described.

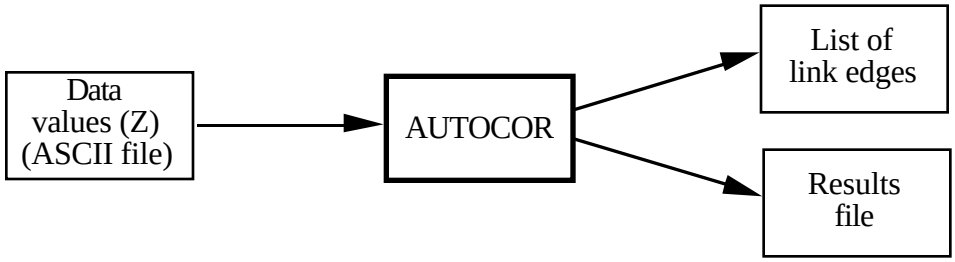
The user may then obtain the three output files described as file types (6), (7) and (8) above.Only option 0 gives access to the CLASSEF output file (type 8).



(2) Options 1 to 11 — Regular grid

These options can only be used for points forming a regular rectangular grid. A single input file is needed: the List of data values (file of type 1). Most of these options’ names refer to movements of the chess game (Upton & Fingleton, 1985, ch. 3), except for the Euclidean distance among points of the grid. The program asks the width and the height of the regular grid it has to construct. The distance between two points is the number of link edges separating them, along the shortest possible path.

The user may obtain output files of types (6) and (7).



(3) Options 12 and 13 — Non-regular point locations

A single input file is needed for these options: the List of coordinates (X, Y) and of values (Z) (file type 2). With option 12, connections among points are computed following the Gabriel graph method (Gabriel & Sokal, 1969), or the Delaunay triangulation method (Dirichlet, 1850; Miles, 1970; Ripley, 1981; Watson, 1981; Upton & Fingleton, 1985; Isaaks & Srivastava, 1989) with option 13. A more detailed description of these connection methods is given with the description of program LINKS. The distance between two points is the number of link edges separating them, along the shortest possible path.

In the Delaunay triangulation method (option 13), there are two ways of imposing "constraints" to the formation of the planar triangulation. See the section dealing with the Delaunay triangulation in program LINKS. Let us recall that a "constraint" is here a set of supplementary points located around the cloud of real locations in the study; these additional points are included in the computation of the triangulation. In the final solution, however, all the link edges that reach the supplementary points are eliminated. In the meantime, these additional points have prevented the formation of some long edges between points located on the outskirt of the cloud of points; had these edges been allowed to be formed, they may not have represented real affinities in the case of distant peripheral points, but simply an edge effect of the sampling design.

Two methods are available in program AUTOCOR to impose such "constraints" to the formation of the triangulation. The question of the program is the following:

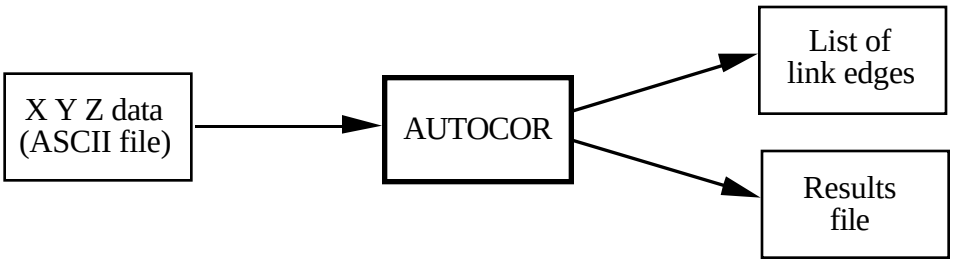
```
Number of constraint points? (-1 = rectangular constraint)
```

1) If you don't wish to impose constraints, answer "0".

2) To impose a rectangular constraint, it is not necessary to explicitly describe them; simply answer by "1". Four supplementary points are then generated by the program at the corners of the cloud of points. See the section dealing with the Delaunay triangulation in program LINKS.

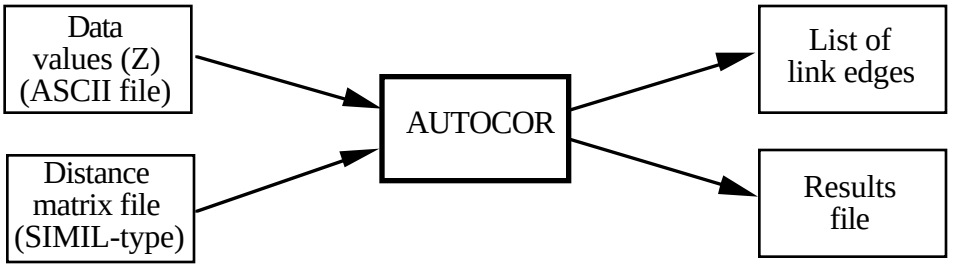
3) If the user wants to impose "constraint" at locations that he has carefully chosen, these must be described at the end of the List of coordinates (X, Y) and of values (Z). Each "constraint" presents itself as the X and Y coordinates of the two extremities of the segment making up the "constraint"; in other words, each "constraint" is represented by four numbers: X_1 Y_1 X_2 Y_2. The program then computes the coordinates of the middle of that segment, to be used as "constraint" in the computations that follow. Answer the question of the program by specifying the number of such "constraint" segments in the file. **This differs from the procedure used in program LINKS.**

The user may obtain output files of types (6) and (7).



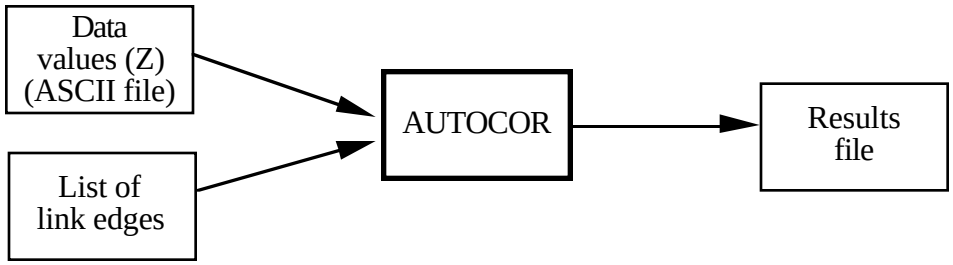
(4) Option 14 — Your own matrix of distance classes

Two input files are necessary for this option: the List of data values (file type 1) and the Matrix of distance classes among objects (file type 4). The connections among objects will be taken as described in the matrix of distance classes. It is not necessary for the points to form a regular grid. In output, the user may obtain output files of types (6) and (7).



(5) Option 15 — Your own list of link edges

Two files are needed when using this option: the List of data values (file type 1) and the list of link edges among objects (file type 5). The connections among objects will be taken as described in the list of link edges, the distance between two points being the number of link edges separating them along the shortest possible path. It is not necessary for the points to form a regular grid. Only the results file (type 6) can be obtained as output.



Questions of the program

The example presented hereunder shows the dialog of the CMS and VMS versions of the program. In the example, the user's answers are underscored and in bold. The questions of the Macintosh version are essentially the same, although they may slightly differ in their formulation. The explanations that follow correspond to the numbers in the left-hand margin of the listing.

(1) The user declares that the data are not nominal.

(2) Among the observation locations, a Delaunay triangulation will be computed (option 13); the distance between two points points is the *number of link edges* separating them along the shortest possible path that follows the triangulation edges.

(3) There are 57 points in the data file. Had there been "constraints" described by a list of supplementary points, these additional points would not be counted in the answer to this question.

(4) A rectangular "constraint" is chosen (see above).

(5) In the case of a completely or partially regular grid of points, it may happen that two solutions are totally equivalent, which expresses itself by crossing edges. The user may choose either to keep both of these edges, or to eliminate one of them. This type of situation cannot happen with the new algorithm used in program LINKS of the Macintosh version.

(6) The user asks to write the list of link edges in file LINK for future use.

Example

```
SPATIAL AUTOCORRELATION analysis.

For all options except 12 and 13, you need to provide a file of
DATA VALUES. For options 12 and 13 on the other hand, you
have to provide a file of COORDINATES containing also, in the
third column, the VALUES of the variable.

For option 13 (Delaunay), if you want to impose constraint
segments, they must be written in that same file, at the end
```

```
     of the list of object data, each in the form of 2 points
     (4 coordinates) describing the segment.

     What is the name of this file? (Defaults are "... data a")
     *** You MUST provide a data file name, even if you don't
     *** need the correlogram but need only the list of link edges.
     file  data  a

     For option 0, you need to provide a binary DISTANCE matrix,
     produced by SIMIL or IMPORT.
     Make sure that it is NOT a similarity matrix.

     What is the name of the file containing this matrix, if any ?
     (Defaults are "... data a")


     For option 14, what is the name of the matrix of DISTANCE
     CLASSES, if any (square or upper triangular) ?
     (Defaults are "... data a")


     For option 15, what is the name of the file of LINK EDGES that
     you have prepared, if any? (Defaults are "... data a")


     What name do you want to give to the output file, containing
     the correlogram? (Defaults are "Result corr a")


     What name do you want to give to the file of LINK EDGES written
     by this program, if any? (Defaults are "Links data a")


     What name do you want to give to the file containing the CLASSEF
     matrix (upper triangular matrix of distance classes, needed
     for Mantel correlograms), if any? (Defaults are "Classef data a")


     S P A T I A L   A U T O C O R R E L A T I O N

     for quantitative or nominal data.
     Version IBM 2.0B
     Author: Alain VAUDOR

     Is your data file already in classes ?
     In other words, are you analyzing NOMINAL DATA ?
(1)  n

     OPTIONS:
       0: Matrix of distances from SIMIL (File "ENTREEB")

      MOVEMENTS IN ONE DIRECTION ONLY:
       1: Horizontal movement (Rows)
       2: Vertical movement (Columns)
```

```
       3: Diagonal movement (positive slope)
       4: Diagonal movement (negative slope)

      DIRECT CHESS MOVEMENTS ONLY:
       5: Rook's move
       6: Bishop's move
       7: Queen's move

      DIRECT AND INDIRECT CHESS MOVEMENTS:
       8: Rook's move
       9: Bishop's move
      10: Queen's move

      11: Euclidean distance on regular grid

      NON-REGULAR POINT LOCATIONS:
      12: Gabriel graph
      13: Delaunay triangulation
      14: Your own matrix of distance classes
      15: Your own list of links (attach file "LINKS")

(2)  13

     Total number of points ?
(3)  57

     Number of constraint points? (-1 = rectangular constraint)
(4)  -1

     Elimination of crossing edges?
(5)  y

     Do you wish to have the list of link edges written on file "LINK" ?
(6)  y

     ***  312 links have been written onto the LINK file ***
     End of the program.
```

BIOGEO

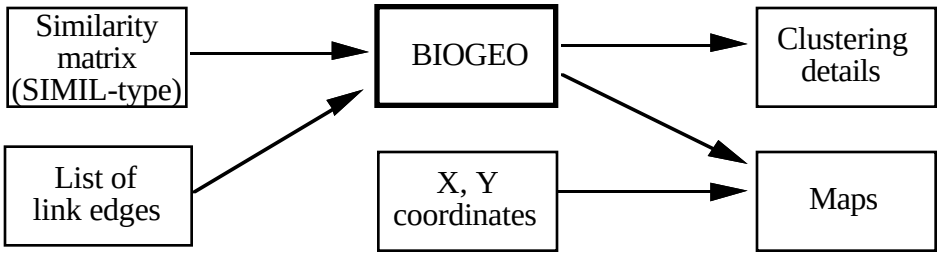
What does BIOGEO do ?

Program BIOGEO computes a hierarchical agglomerative proportional-link linkage clustering with constraint of spatial contiguity, as proposed by Legendre & Legendre (1984b), and presents the results as a series of maps, one for each clustering level. Since the clustering is computed from a similarity matrix, which is of multivariate origin in most instances, then this method may be considered as a form of multivariate mapping.

This agglomerative clustering uses a proportional-link linkage algorithm; another program of this package, K-MEANS, allows to compute a non-hierarchical space-constrained clustering. The connectedness of the clustering algorithm is determined by the user, between 0 (for single linkage) and 1 (for complete linkage). Legendre (1987) has shown that the results of constrained clustering are fairly stable through a wide range of values of connectedness.

If the dimensions of the program (in the CMS and VMS versions) are not large enough, they can be adapted to your data set by changing the values of the parameters at the beginning of the program listing, followed by re-compiling. This is also the case with all the other programs of this package. In the Macintosh version, the program is limited to a maximum of 150 simultaneous groups at any one clustering level. Problems larger than 1000 localities have been analyzed using this program; it may become necessary in such cases to request more core memory than attributed to users by default.

Input and output files



(1) File of similarities

A file of similarities produced by programs SIMIL, IMPORT (CMS and VMS versions) or IMPORT-EXPORT (Macintosh version) is always needed to run this program; that file contains the information about the resemblance relationships among objects. A distance matrix must be converted into a similarity matrix, using program CONVERT, before using it in BIOGEO.

(2) File of link edges

The spatial relationships are provided to the program in the form of an ASCII list of connecting edges among neighboring localities. Each link edge is represented by a pair of object numbers, written in free format and separated by at least one space. That file may have been produced either by AUTOCOR (in the CMS and VMS versions) or by LINKS (in the Macintosh version). The following example corresponds to the rook's connection scheme among 12 localities forming a regular grid of 3 rows and 4 columns; each link edge is represented by a pair of numbers:

```
 1  2     2  3     3  4     5  6     6  7     7  8     9 10    10 11
11 12     5  1     6  2     7  3     8  4     9  5    10  6    11  7
12  8
```

It is easy to modify that file using an ASCII editor, if one needs to add or remove link edges from the list. Notice that the file may also be entirely written by hand using the ASCII editor; unorthodox geographic connection files of interest for the particular study, such as one containing the list of first- AND second-order neighbors for each locality, should be prepared by hand if no program is available to do so. Notice also that if the list of connections includes all possible pairs of localities, the clustering procedure is now without spatial contiguity constraint; this is available as an option in the Macintosh version. One may choose to use BIOGEO in that way in order to obtain maps for each clustering step, instead of a dendrogram.

Before starting the program, make sure that you know how many link edges (pairs of values) there are in the LINK file. Hint: write down that number as part of the file name.

(3) File of spatial coordinates (X, Y)

If one wants to ask the program to draw maps for each clustering level, which is an option of the program, one must provide a list of the coordinates of the point locations. That list of coordinates is used solely for the purpose of drawing the maps. The coordinates are provided as an ASCII file of integer or real numbers; the coordinates must not be in the degree-minute-second form. The number of coordinates in that file must correspond to the number of objects in the study. When using the CMS or VMS versions, don't forget to put a zero before the decimal mark ("0.376" instead of ".376").

In certain cases, a more didactic representation may be obtained by imposing coordinates that do not exactly correspond to the geographic positions. For instance, to analyze in a single run repeated observations (through time) of a set of localities, the output maps may be organized in such a way that the various time slices come out as a mosaic of separate parts in the final picture. This may be done because the coordinates provided in the file of spatial coordinates are used solely to draw the maps; the spatial or spatio-temporal relationships that are taken into account during the clustering process are those described in the file of link edges only.

(4) File of sorted similarities

In the CMS and VMS versions, one may choose to save the file of sorted similarities in order to use it during another run. That option is particularly interesting when the similarity matrix is large (beware: long sorting time), and one wishes to obtain results with various values of connectedness.

(5) File of results

In the CMS and VMS versions, only one type of output file is available; it contains the clustering results and the maps, if requested. The maximum number of maps is equal to the number of clustering steps, which is n - 1. The user may choose not to have all the maps copied onto the output file, because the resulting file would be bulky while the first maps (which correspond to the highest similarity levels) are not very informative in most instances; so, it is possible to tell the program how many of the last maps should be copied onto the output file. See section "Contents of the file of results" for more details.

In the Macintosh version, the mapping function has been separated from the file of clustering results. The file with the clustering steps is optional. If one wishes to see the maps, they are first produced on the screen, one at a time. There are two ways to choose the map that the user wants to see: either by writing down a similarity linkage level, or with the help of a cursor that appears on the screen and which indicates the number of groups formed at each linkage level (the first steps of the agglomerative clustering, with high similarity values, are at the bottom of the screen); one brings the cursor to a given linkage similarity level, knowing the number of groups formed at that level, and clicks the mouse. There are also other options available to choose a map from; see the pull-down menu "Maps". Notice that several linkage level may correspond to the same similarity level, each "clustering

materialized by envelopes surrounding the members of each cluster, insofar as the situation allows; dendrites may be added to envelopes to bring in a remote location, when there is no easier graphical way of representing a cluster. In areas of the picture with many points, it may not be easy to tell where the group separations are; one may zoom in any portion of the picture by drawing a rectangle around it using the mouse. A subsection of the enlarged portion may be zoomed in again; command “Finish” in the pull-down menu allows to zoom out. Maps may also be printed, or saved as PICT files; maps are identified by a title and a linkage similarity level.

Options of the program

The following options are available. The numbers refer to the flags in the left-hand margin of the example below.

- Choosing the connectedness level (*Co*) of the proportional-link linkage clustering procedure (4).
- Drawing the maps, or not (2 and 5).
- In the Macintosh version: obtaining, or not, details about the groups formed at each similarity level.
- With the CMS and VMS versions: saving the sorted similarity file, or not (1 and 3).

Example

The following example shows how to use the program to compute a clustering with spatial contiguity constraint. The calling file, whose dialog makes up the first part of the example, asks the names of the various files; this example has been run under CMS. The questions of the Macintosh version are essentially the same, although their formulation may differ slightly.

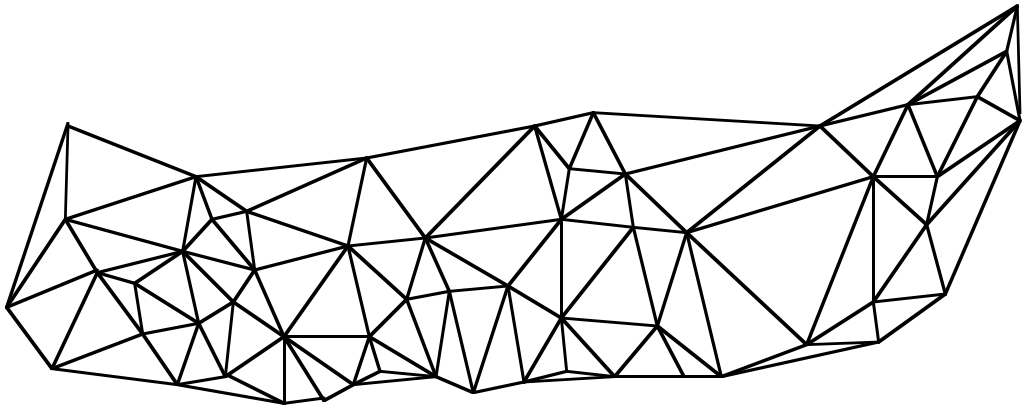
The first point to notice concerns the file of sorted similarities (1): one answers that question only if one wishes to save the sorted similarities, or else if one wishes to use here a file that has been saved during a previous run (in which case the answer is *Yes* to the question flagged (3)). The second point is that the file of coordinates is optional (2); it is needed only if one wishes to have the program draw the maps (in which case the answer is *Yes* to the question flagged (5)). when drawing the maps, the CMS and VMS versions assume that the first column of the file of coordinates is the abscissa (with values increasing from left to right) while the second column is the ordinate (with values increasing from bottom to top); the user must determine the width of the map by answering question (6). In the Macintosh version, the coordinate with the largest range is always taken to be the abscissa, which may produce a rotation of the map to use screen space to best advantage.

```
     BIOGEO: Clustering with spatial contiguity constraint.

     What is the name of the SIMIL similarity matrix file?
     (Defaults are "... data a")
     (You MUST provide this file name even if you provide a sorted
     file of similarities in the next question.)
     myfile  s16  a

(1)  Do you want to save the sorted similarity file for further use?
     Or, do you already possess that file? In either case,
     what is the name of the file? (Defaults are: "FICHTRI data a")
     myfile  fichtri  a

     What is the name of the file of CONSTRAINT LINKS ?
     (Defaults are "... data a")
     myfile  link146  a
```

```
(2)  What is the name of the file of COORDINATES, if any?
     (Defaults are "... data a")
     myfile  coord  a

     What is the name of the OUTPUT file from BIOGEO?
     (Defaults are "MAPS BIOGEO a")
     myfile  maps  a

     Execution begins...

      P r o g r a m  B I O G E O

      Author: A. Vaudor

(3)   Did you provide a sorted similarity file (FICHTRI) ?
       (Y or N)
     n
      Title of this run
     Clustering  with  spatial  contiguity  constraint
      Number of edges in the list of constraint links?
     146
(4)   Connectedness for clustering algorithm (Max: 4 significant digits)
     1.0
      There are 56 clustering steps.
       How many of the last steps do you want to see?
     20
(5)   Do you want the maps printed? (Y or N)
     y
(6)   Width of the maps (in characters, without the frame):
     60
      End of the program.
```

Contents of the file of results

The file shown below is the result of a run under CMS. To give substance to the clustering details, maps have been requested for each clustering level. The connectedness of the proportional-link linkage clustering has been chosen to be $Co = 1.0$. The spatial relationships among localities, which are described by the file of link edges, are illustrated by this map produced by program LINKS:



```
B I O G E O : Space-constrained clustering

Author: A. Vaudor
```

In the list of 57 objects below, each object is identified by the number of the group to which it pertains. Group numbers are not necessarily sequential. Unclustered objects receive a zero.

```
Level:               1.00000
Connectedness:       1.00000
Number of groups:    9

  1   0   0   1   1   0   1   1   1   5   5   0   5   1   1  16  16  16
  1   0   1   1   1   1   1   0   2   2   2   2   2   2   2   6   2   2
  2   6   6   0   4   4   4   4   0   0   4  12  12   0  13  13  13  12
 15  15  13
Number of clustered localities:  47
------------------------------------------------------------
!                                                  1       !
!                                                          !
!                                                 1        !
!   +                        2  2                  1       !
!          %       2             1           1  1          !
!   %        %                1   1              1         !
!         *        2  2             1                      !
!     * *     4      2  2  1                 5   5         !
!+      4      4    6             1       5                !
!  *        4      62  2   =  = =   1 1                     !
!         4      6       2                                 !
------------------------------------------------------------
```

After no. 9, the symbols used in the map are not related to group numbers any more.

```
Level:               0.12500
Connectedness:       1.00000
Number of groups:    4

  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   2   2   2   2   2   2   2   4   2   2
  2   4   4   4   4   4   4   4   3   3   4   2   2   2   4   4   4   2
  2   2   4
Number of clustered localities:  57
------------------------------------------------------------
!                                                  1       !
!                                                 1        !
!                                                1   1     !
!   2                        2  2           1       1      !
!         2       2          1  1             1  1         !
!   2       22               1   1              1         !
!         4        2  2             1                     !
!     4 4    34      2  2  1                 1   1        !
!2      4  3   4    4        1    1       1   1            !
!  4       4      42  2   1  1 1   1 1                     !
!         4    4 4       2                                !
------------------------------------------------------------
```

The maps produced by the Macintosh version are of better graphical quality; see examples in the description of program K-MEANS. Objects are represented by their sequential number in the input file. The groups are materialized by envelopes surrounding the members of each cluster.

CHRONO

What does CHRONO do ?

The chronological clustering proposed by Legendre, Dallot & Legendre (1985) is computed by program CHRONO. This clustering method, which had first been described for multivariate time series, can also be used to segment spatial series (Galzin & Legendre, 1987). The non-hierarchical method uses a hierarchical proportional-link linkage algorithm whose connectedness level (*Co*) is determined by the user as an answer to a question of the program; it is the test of significance, described in the next paragraph, that makes the method non-hierarchical. The constraint of spatial or temporal contiguity imposed to the clustering results means that only objects or object groups that are adjacent along the series may eventually cluster. Notice that it is unlikely that changing the connectedness would produce a major change in the clustering results, as can be seen in the examples of the Legendre, Dallot & Legendre (1985) paper.

At each step of the agglomeration, a permutation test is performed to decide whether a fusion should be made between the two groups whose fusion is proposed by the agglomerative algorithm. The null hypothesis of that test is explicitly described in the output of versions CMS and VMS:

```
H is the probability that the null hypothesis is true. The
null hypothesis says that the two groups being tested  are
an artifact and should be fused in a single group.  Fusion
occurs if H is larger than the probability level ALPHA set
by the user (above).
```

Answering a question of the program, the user must determine the *alpha* rejection level of the null hypothesis (often chosen values are 0.01, 0.05 or 0.10; one may choose to use a higher level in order to identify singletons — see below, as well as the example). One must realize, though, that this is not a genuine test of statistical hypothesis, since the data used during the test are the same as those from which the hypothesis of division into groups has been generated. Numerical simulations described in the main reference have shown, however, that for random data sets, the probability for this test of producing a significant result is equal to the preselected *alpha* value.

The program allows to identify *singletons*, which are aberrant samples found along the data series. Because of the constraint of contiguity imposed on the algorithm, the presence of a singleton may prevent the formation of a group that should have included objects from both sides of the aberrant sample. At least three reasons may produce such aberrant samples: (1) random events, such as modified strata in sediment cores, or else movements of water masses during repeated samplings at the same station in aquatic environment; (2) improper sampling or inadequate preservation of the samples before they are analyzed; (3) extreme stochastic variations, which lead to rejecting the null hypothesis while no break has occurred in the succession (type II error).

If the user requests to identify the singletons, the clustering will be interrupted, and started again from the beginning after removing the singleton (see example); the only exceptions to this rule are the singletons located at the beginning or the end of the data series, since no group is interrupted by their presence. It is unlikely that singletons will be identified if the *alpha* level is low (less than 10%), because it then becomes difficult, when testing a single object against p, to obtain a value which is smaller than that in the first column of Table 1. Final rule: if an object has a similarity of zero with all its immediate neighbors, the agglomerative algorithm does not go down to level $S = 0$ to force such an object to pertain to a group; these unclustered objects are represented by dashes (-) in the final solution, or by a white square in the Macintosh output graph. It is recommended to check the data for any object coming out with that symbol; if its presence in the series seems to have interrupted a group, this object may either be removed from the analysis if it is considered aberrant or exceptional (which may have given it a null similarity with its neighbors).

Table 1 — The lowest possible probabilities of fusion for two groups with p_1 and p_2 objects respectively (except in cases of tied similarity values). From Legendre *et al.* (1985), Table C1.

	p_1				
p_2	1	2	3	4	5
2	0.66667	0.33333			
3	0.25000	0.10000	0.10000		
4	0.20000	0.06667	0.02857	0.02857	
5	0.16667	0.04762	0.01786	0.00794	0.00794
6	0.14286	0.03571	0.01190	0.00476	0.00217
7	0.12500	0.02778	0.00833	0.00303	0.00126
8	0.11111	0.02222	0.00666	0.00202	0.00078
9	0.10000	0.01818	0.00455	0.00140	0.00050
10	0.09091	0.01515	0.00350	0.00100	0.00033
11	0.08333	0.01282	0.00275	0.00073	0.00023
12	0.07692	0.01099	0.00220	0.00055	0.00016
13	0.07143	0.00952	0.00179	0.00042	0.00012
14	0.06667	0.00833	0.00147	0.00033	0.00009
15	0.06250	0.00735	0.00123	0.00026	0.00006
16	0.05882	0.00654	0.00103	0.00021	0.00005
17	0.05556	0.00585	0.00088	0.00017	0.00004
18	0.05263	0.00526	0.00075	0.00014	0.00003
19	0.05000	0.00476	0.00065	0.00011	0.00002
20	0.04762	0.00433	0.00056	0.00009	0.00002

Input and output files

(1) Input file

The input file must imperatively be a similarity, and NOT a distance file, produced by SIMIL, or else by IMPORT (in the CMS or VMS versions) or IMPORT-EXPORT (in the Macintosh version). A distance matrix may easily be converted into similarities using the CONVERT utility program. CHRONO will assume that the chronological or spatial order of the objects is the same as the order of the objects in the input file.

(2) File of results

The results are presented either on the screen only (CMS and VMS versions), or both on the screen and in an output file (Macintosh version). The clustering results are shown first. Although the algorithm is hierarchical, the final result is non-hierarchical, as explained above. This result is presented in the form of a graph in the Macintosh version. It is also presented at the *last line* of the list of clustering steps (on the screen with the CMS and VMS versions; in a file with the Macintosh

version); the lines that precede the last one are not very informative, and are presented only to show the user that the program is still working for him. The last line of that list only is thus to be saved and reproduced in publications.

A posteriori tests may be performed, where each group in turn is expanded, assuming that the other groups of the transect do not exist and their objects are unclustered and available for clustering; group expansion makes it possible to determine whether the groups formed are abruptly separated from one another (succession by jumps, or "saltation"), or if the transition among groups is smooth (gradual succession). Other *a posteriori* tests allow to look at the relationships among distant groups to determine if some of them are similar (refer to the null hypothesis, spelled out above, to understand how these tests must be interpreted; see also the example below). The program does the same with the singletons and tries to determine whether they are similar to one or the other of the more distant groups. In those *a posteriori* tests, fusions may be made between small groups simply because of the fact that it is impossible for the tests to take probability values smaller than the minimum values described in Table 1 — especially if the *alpha* level established at the beginning of the run is small.

Notice that the *a posteriori* tests are expensive in terms of computing time, especially the group expansions. They are usually not requested during exploratory runs of a data file; they are called after the most informative combinations of the *alpha* and connectedness parameters of the program have been found. In the mainframe versions, if the user wishes to save and print the results, it is necessary to have them written in a screen memory file, as explained on page 2 of the present document.

Options of the program

The options of the program are the following. Numbers refer to same numbers in the left-hand margin of the example below.

- Choice of the connectedness level (*Co*) of the proportional-link linkage agglomerative clustering (1).
- Choice of the *alpha* significance level of the permutation test (2).
- Option to eliminate singletons, or not (3).
- *A posteriori* tests for group expansion (4) and among distant groups (5).

Example

The example below illustrates the use of the program to compute a clustering with a one-dimensional contiguity constraint, which is spatial in the present case. The input file represents a 24-station spatial transect where 41 species have been identified. The Steinhaus similarity coefficient (S17) has been used to compare stations. In that example, run under CMS, the calling program requests only the name of the similarity matrix file. In the Macintosh version, the dialogue also asks the name to be given to the file of results.

The last line of the clustering list (6) is the only one that contains information of interest. It reads as follows. The 24 sampling stations of the transect are each represented by a character:

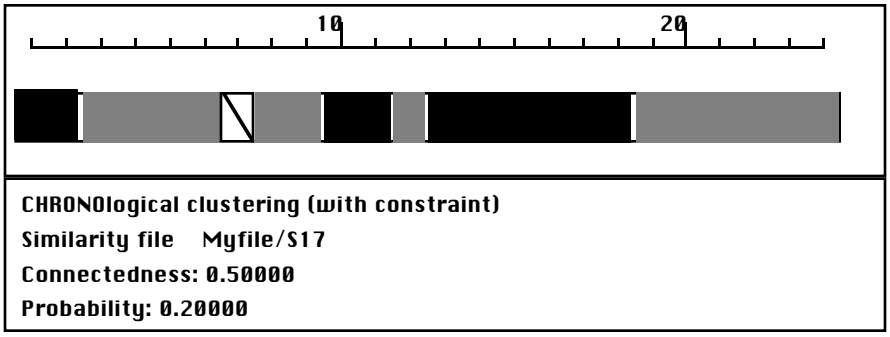
```
AABBBB*BBCC-DDDDDDEEEEEE   S:  0.26667 H: 0.30000
```

The first sampling station is located at the extreme left of the list. The groups formed are represented by letters; in this example for instance, five groups are formed, and they are represented by letters A to E. Unclustered stations are represented by dashes (-) and singletons by asterisks (*). The difference is that singletons have been tested and found to differ from the groups to their left and right, so that they cannot fuse with them; unclustered stations, on the contrary, have not been tested (see section "What does CHRONO do?", as well as below). The value after "S" represents the linkage similarity level

is true for that same fusion.

The Macintosh version produces the following picture that summarizes the clustering results; it may be saved as a PICT file, so that it can be edited and included directly into publications. In the picture, successive groups are represented by alternating black and gray zones. Object 7, represented by a barred white square, is a singleton; it is not the same as object 12 which is a one-sample group and represents a special case. The difference lies in the fact that object 12 has similarities of zero with its neighbors; since the clustering algorithm stops before clustering level $S = 0$, then this object is never clustered, and it remains as a one-object group; since it is not subjected to testing either, it does not receive the barred white square of singletons. Such objects may, by their sheer presence in a series, block the formation of groups; when they are found in an analysis, one must wonder whether they are aberrant samples in one way or another, and if so they must be removed from the study.

Spatial transect



One could also represent the objects in a space of reduced dimension obtained by correspondence analysis, principal coordinates analysis or nonmetric multidimensional scaling of the S17 matrix, etc., and draw lines connecting the members of each group.

```
What is the name of the SIMILARITY matrix file?
(defaults are "... data a")
myfile  s17  a

Execution begins...                 Announces the beginning of the sorting program
Execution begins...                 Announces the beginning of the clustering program

 C H R O N O L O G I C A L   C L U S T E R I N G

 DEPARTEMENT DE SCIENCES BIOLOGIQUES
 UNIVERSITE DE MONTREAL
 C. P. 6128, SUCC "A"
 MONTREAL, QUEBEC  H3C 3J7.

 Reference describing the method:
 Legendre,  P.,  S.   Dallot,  and  L.   Legendre.  1985  --
    Succession of species within a community:  chronological
    clustering,  with applications to marine  and freshwater
    zooplankton.  The American Naturalist, 125 (2): 257-288.

(1)  CONNECTEDNESS OF CLUSTERING ?
     0.5
      Connectedness:     0.50
```

```
(2)  ALPHA LEVEL FOR THE CLUSTER FUSION TEST ?
     0.20
      Cluster fusion level (ALPHA):  0.20000

(3)  REMOVAL OF SINGLETONS (Y or N) ?
     n
      No elimination of singletons.

      WIDTH OF YOUR TERMINAL, IN N. OF COLUMNS ?
     80                          The usual width of a terminal is either 80 or 132 characters

(4,5) DO YOU WANT THE A POSTERIORI COMPUTATIONS (Y or N) ?
     y

      H is the probability that the null hypothesis is true. The
      null hypothesis says that the two groups being tested  are
      an artifact and should be fused in a single group.  Fusion
      occurs if H is larger than the probability level ALPHA set
      by the user (above).

      ------------------AA----   S:  0.84615
      ----------------AABB----   S:  0.84211
      AA--------------BBCC----   S:  0.81818
      AA--------------BBCCC---   S:  0.81481 H: 0.66667
      AABB------------CCDDD---   S:  0.71429
      AABB-----------CCCDDD---   S:  0.66667 H: 0.66667
      AABBB----------CCCDDD---   S:  0.53333 H: 0.66667
      AABBBB---------CCCDDD---   S:  0.53333 H: 1.00000
      AABBBB-------CCDDDEEE---   S:  0.50000
      AABBBB------CCCDDDEEE---   S:  0.50000 H: 0.66667
      AABBBB---CC-DDDEEEFFF---   S:  0.44444
      AABBBB---CC-DDDDDDEEE---   S:  0.42105 H: 0.40000
      AABBBB-CCDD-EEEEEEFFF---   S:  0.40000
      AABBBB-CCCC-DDDDDDEEE---   S:  0.30769 H: 0.33333
      AABBBB-CCCC-DDDDDDEEEFF-   S:  0.30000
      AABBBB-CCCC-DDDDDDEEEFFF   S:  0.28571 H: 0.66667
      AABBBB-CCCC-DDDDDDEEEEEE   S:  0.26667 H: 0.30000
      OBJECT:  7 IS A SINGLETON  H: 0.20000 0.20000

      ------*-----------AA----   S:  0.84615
      ------*---------AABB----   S:  0.84211
      AA----*---------BBCC----   S:  0.81818
      AA----*---------BBCCC---   S:  0.81481 H: 0.66667
      AABB--*---------CCDDD---   S:  0.71429
      AABB--*--------CCCDDD---   S:  0.66667 H: 0.66667
      AABBB-*--------CCCDDD---   S:  0.53333 H: 0.66667
      AABBBB*--------CCCDDD---   S:  0.53333 H: 1.00000
      AABBBB*------CCDDDEEE---   S:  0.50000
      AABBBB*-----CCCDDDEEE---   S:  0.50000 H: 0.66667
      AABBBB*--CC-DDDEEEFFF---   S:  0.44444
      AABBBB*--CC-DDDDDDEEE---   S:  0.42105 H: 0.40000
      AABBBB*CCDD-EEEEEEFFF---   S:  0.40000
      AABBBB*BBCC-DDDDDDEEE---   S:  0.40000 H: 0.26667
```

```
     AABBBB*BBCC-DDDDDDEEEFFF   S:  0.28571 H: 0.66667
(6)  AABBBB*BBCC-DDDDDDEEEEEE   S:  0.26667 H: 0.30000          Clustering results

     ELAPSED TIME: 0.7143 SEC

(4)  GROUP EXPANSION TESTS

          [   1 ..    2]                                  The first group [1 .. 2]
          [   1 ..    3]  H:    0.66667          is used as starting point for expansion
          [etc.]
          [   1 ..    9]  H:    1.00000
          [   1 ..   10]  H:    0.44444
          [   1 ..   11]  H:    0.30000    After expansion, the group spans from 1 to 11


          etc.                                          Each group, in turn, is used
          etc.                                        as starting point for expansion


          [  19 ..   24]                                  The last group [19 .. 24]
          [  18 ..   24]  H:    0.85714          is used as starting point for expansion
          [  17 ..   24]  H:    1.00000
          [  16 ..   24]  H:    1.00000  After expansion, the group spans from 16 to 24

     ELAPSED TIME: 1.0083 SEC

(5)  TESTS AMONG GROUPS
     [  1 ..   2] against [  3 ..   9]  H: 0.03571     No fusion because H ≤ alpha
                          [ 10 ..  11]  H: 0.33333                              *
                          [ 12 ..  12]  H: 0.33333                             **
                          [ 13 ..  18]  H: 0.03571     No fusion because H ≤ alpha
                          [ 19 ..  24]  H: 0.03571     No fusion because H ≤ alpha
     [  3 ..   9] against [ 10 ..  11]  H: 0.03571     No fusion because H ≤ alpha
                          [ 12 ..  12]  H: 0.14286     No fusion because H ≤ alpha
                          [ 13 ..  18]  H: 0.14286     No fusion because H ≤ alpha
                          [ 19 ..  24]  H: 0.02814     No fusion because H ≤ alpha
     [ 10 ..  11] against [ 12 ..  12]  H: 0.33333                             **
                          [ 13 ..  18]  H: 0.10714     No fusion because H ≤ alpha
                          [ 19 ..  24]  H: 0.03571     No fusion because H ≤ alpha
     [ 12 ..  12] against [ 13 ..  18]  H: 0.14286     No fusion because H ≤ alpha
                          [ 19 ..  24]  H: 0.14286     No fusion because H ≤ alpha
     [ 13 ..  18] against [ 19 ..  24]  H: 0.07359     No fusion because H ≤ alpha
```

** This value represents the lowest possible probability of fusion between these two groups, given their size (see Table 1). So, it does not necessarily mean that H_0 is not rejected.*

*** This is also the lowest possible probability of fusion between these two groups, given their size. It is lower than that of Table 1 because of ties among the similarity values.*

```
TESTING SINGLETONS AGAINST ALL GROUPS

[  7] against [  1 ..   2]  H: 0.66667                              *
              [  3 ..   9]  H: 0.14286          No fusion because H ≤ alpha
              [ 10 ..  11]  H: 0.66667                              *
              [ 13 ..  18]  H: 0.14286          No fusion because H ≤ alpha
              [ 19 ..  24]  H: 0.28571          ⇐ Fusion of [7] and [19 .. 24]

ELAPSED TIME: 1.6521 SEC


INPUT FILE:

    NUMBER OF OBJECTS :   24                    Identification of the input file
    NUMBER OF VARIABLES :   41
    TITLE :  Data file
    DATE : 02/04/91
    RESEMBLANCE FUNCTION USED :  s17

End of the program.
```

CLUSTER Macintosh

What does CLUSTER do ?

Program CLUSTER implements several methods of agglomerative hierarchical clustering, which are briefly described below. This program plays on the Macintosh the same role as programs INTERLNK and LANCE of the CMS and VMS versions.

Input and output files

The input file is a SIMIL-type similarity matrix. The output is a dendrogram, which may be accompanied, or not, with some statistics (see below); the user can send it to a printer or to a file of results. If a laser printer is available, the dendrograms are of camera-ready graphic quality and may by directly included in publications; the user may decide their size (width in cm) as well as the fonts used in the layout. Dendrograms sent to a file are of line-printer quality, similar to those produced by INTERLNK and LANCE in the CMS and VMS versions (notice: width is in number of characters).

Options

The user must first choose between the Lance & Williams (1966a, 1967) algorithm, also used in program LANCE of versions CMS and VMS, and the proportional-link linkage algorithm (Sneath, 1966) also used in program INTERLNK.

```
Computation option:
  Lance & Williams
  Proportional-link linkage
```

If proportional-link linkage has been chosen, the value of connectedness (*Co*) must be provided. For the general agglomerative clustering algorithm of Lance & Williams, the choice is as follows:

```
Clustering options:
  Unweighted arithmetic average (UPGMA)
  Weighted arithmetic average (WPGMA)
  Unweighted centroid (UPGMC)
  Weighted centroid (WPGMC)
  Ward's method
  Other
```

If the choice is "Other", the user must give the values of parameters *alpha*[j], *alpha*[m], *beta* and *gamma* required by this algorithm; see description of program LANCE. After each clustering, the user has to answer two more questions relative to the complementary clustering statistics that can be computed (see description below):

```
Minimum spanning tree?                                          [Yes, No]
Cophenetic correlation, Gower's distance & entropy?              [Yes, No]
```

Following this, the program proposes to write the cophenetic distance matrix onto a SIMIL-type file.

Clustering statistics

The following statistics are available to judge of the adequation between the input similarity matrix and the clustering produced by the program.

(1) Chain of primary connections

The *chain of primary* or *external connections*, also called *dendrites*, *network*, *Prim network*, *minimum spanning tree*, *minimum length tree* or *shortest spanning tree*, is the set of links between object pairs that represent the basic structure of the clustering. A *primary connection* is formally defined as the first similarity link that makes an object a member of a group, or permits two groups to fuse, in single linkage clustering (Legendre & Legendre, 1984a). In the agglomerative clustering programs of the "R" package, the notion of primary connection is here extended to represent the similarity link which, in the fusion of two groups, unites the two objects (one in each group) that are the closest to one another (*i.e.*, highest similarity); so, the chain of primary connections is defined as the set of these first similarity links, whatever the clustering method that has been used to agglomerate the objects.

Borrowing the example of 5 pools used by Legendre & Legendre (1983) in their chapter on clustering, a single-linkage agglomerative clustering has been computed. The dendrogram is presented in the chapter of the present document devoted to program INTERLNK. The chain of primary connections provided by the program for this single linkage agglomeration is the following:

```
M i n i m u m  s p a n n i n g  t r e e

    Level    Distance        Chain
-----------------------------------------------

   0.40000   0.40000  (POOL 214   , POOL 212   )
   0.78600   0.78600  (POOL 432   , POOL 214   )
   0.70000   0.70000  (POOL 431   , POOL 233   )
   0.50000   0.50000  (POOL 432   , POOL 431   )
```

The list means that at clustering level D = 0.4, the chain of primary connections contains a first link between pools 212 and 214; these two objects are located at distance D = 0.4 (or S = 1 - 0.4 = 0.6) in the input similarity matrix; and so on. In the case of single linkage clustering, the fusion levels are always the same as the distances between closest neighbors, by definition of the method. This is not the case with other clustering methods, however. In unweighted arithmetic average clustering (UPGMA: below) for instance, the two values differ on the second and third lines; 'level' is still the fusion level of the two groups in the dendrogram, while 'distance' is the one-complement (D = 1 - S) of the similarity between the two closest objects from these two groups:

```
M i n i m u m  s p a n n i n g  t r e e

    Level    Distance        Chain
-----------------------------------------------

   0.40000   0.40000  (POOL 214   , POOL 212   )
   0.94200   0.78600  (POOL 432   , POOL 214   )
   0.75000   0.70000  (POOL 431   , POOL 233   )
   0.50000   0.50000  (POOL 432   , POOL 431   )
```

Notice that in the chain of primary connections, as well as in the dendrogram, the fusion levels as well as the measures of resemblance between objects in the chain are expressed as distances instead of similarities; this choice results from the fact that for the many distance coefficients that can take

values in the interval [0, ∞], the dendrograms would show negative fusion levels if these were expressed as similarities S = (1 - D). This would be of no consequence for the clustering, but it would make it more difficult for unexperienced users to understand and use these fusion levels.

(2) Cophenetic correlations

Any hierarchical clustering can be represented by a *cophenetic matrix* among objects (Sokal & Rohlf, 1962; Legendre & Legendre, 1983; Jain & Dubes, 1988; etc.). In that matrix, the similarity between two objects is equal to the value of the fusion level where these two objects become members of the same group, in the dendrogram. In all methods of agglomerative clustering — except occasionally in unweighted or weighted centroid clustering, or when using some unusual combinations of parameters of the general agglomerative model of Lance & Williams (1966, 1967) — the cophenetic matrix so obtained is also ultrametric.

The linear Pearson correlation between the values in the cophenetic matrix and those in the original similarity matrix (excluding the values on the diagonal) is called *cophenetic correlation, matrix correlation* or *standardized Mantel statistic*. This correlation measures to what extent the clustering results correspond to the original similarity matrix; if the clustering perfectly rendered the similarities in the original matrix, the cophenetic correlation would be 1. Notice that logically, that correlation cannot be tested for significance, since the cophenetic matrix is not independent from the original similarity matrix; one comes from the other through the clustering algorithm. To test that correlation, one would have to pretend that the two matrices are independent from one another under H_0; in other words, one would have to say that the clustering algorithm is likely to have a null efficiency, which is fortunately not the case with most current clustering algorithms…

Drawing a graph (Shepard's diagram) of the similarities (or distances) of the cophenetic matrix against the similarities (or distances) of the original resemblance matrix, the relationship may be observed to be curvilinear instead of linear. If one is more interested in the topological structure of the dendrogram than in the precise branch lengths, it is of greater interest to look for a monotonic instead of a linear relationship between the values in the two matrices. In that case, computing a nonparametric correlation is appropriate, instead of a Pearson correlation; Kendall's nonparametric correlation (τ_b) between the cophenetic matrix and the original resemblance matrix is provided by the program. The correlation coefficients take values in the interval [-1, 1]. The sign of the cophenetic correlation is expected to be positive, since the original similarities are compared to the cophenetic similarities. The higher is the value of the cophenetic correlation, the better the adjustment. Following is an example of the measures of adjustment computed by the program, for an unweighted arithmetic average clustering (UPGMA) of the 5 pools:

```
Cophenetic correlations
      Kendall's tau b    0.77364
      Pearson's r        0.95111
      Gower's distance   0.03962
```

Gower's distance, which is the last measure of adjustment available in the "R" hierarchical clustering programs, is described below.

(3) Gower's distance

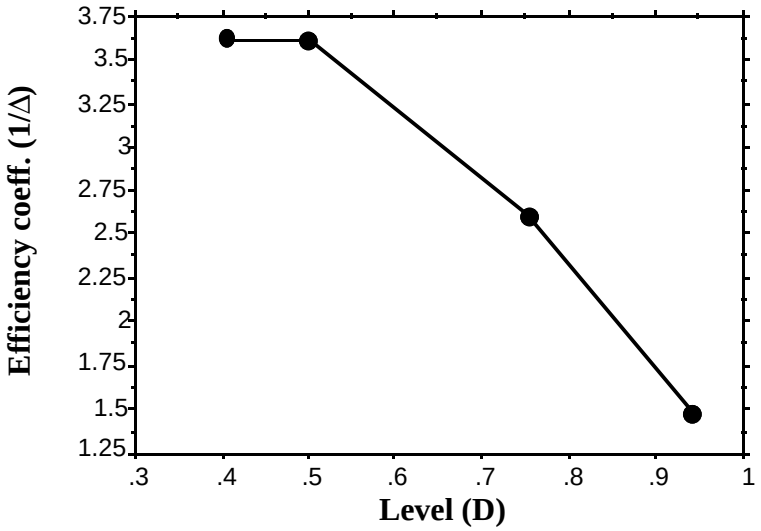
Gower's (1983) distance is the sum of the squared differences between the values in the cophenetic similarity matrix and in the original similarity matrix. That measure of adjustment takes values in the interval [0, ∞]. The smaller the value of Gower's distance, the better the adjustment is. As in the case of the cophenetic correlations, this measure simply has a comparative value among clustering results obtained for the same original similarity matrix.

(4) Efficiency coefficients

The *efficiency coefficients* (Lance & Williams, 1966b) are computed as $1/\Delta$, where Δ (delta) represents the amount by which the information in the classification is reduced due to the fusion of groups. The reduction is computed as the entropy in the classification before the fusion level, minus the entropy after the fusion. An efficiency coefficient is provided by the program for each fusion level. As long as the algorithm only clusters individual objects, or lesser groups, the Δ values are small; consequently, the corresponding values of the efficiency coefficient are large. Plotting a graph of the values of the efficiency coefficient as a function of the agglomerative clustering steps, the minima of that graph indicate the most important fusion levels. If one wants to select a single cutting level through the dendrogram, the efficiency coefficient may help take that decision. It is never a constraining decision criterion, however, since no test of statistical significance has been performed.

```
Efficiency coefficients

           Level    Entropy    delta    1/delta
-------------------------------------------------
         0.00000   1.60944
         0.40000             0.27726   3.60674
                   1.33218
         0.50000             0.27726   3.60674
                   1.05492
         0.75000             0.38191   2.61843
                   0.67301
         0.94200             0.67301   1.48586
                   0.00000
```

The following graph presents the values of the efficiency coefficient ($1/\Delta$) as a function of the fusion levels (distances). The best vertical cutting level in the dendrogram would then be located before the last level, which would produce two groups.



COCOPAN

What does COCOPAN do ?

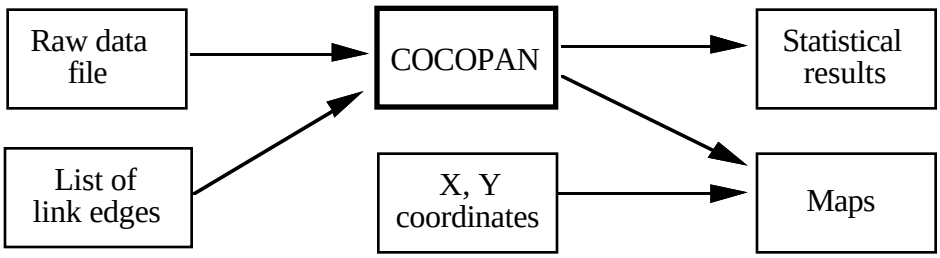
This program carries out a one-way analysis of variance for spatially autocorrelated quantitative data when the classification criterion is a subdivision of the study area into nonoverlapping regions — for instance countries, counties, language areas, geomorphological subdivisions of the study area, and so on, as found in numerous problems where the data points can be plotted on a map. The method has been described by Legendre, Oden, Sokal, Vaudor and Kim (1990). The acronym COCOPAN stands for ***C****ontiguity-****c****onstrained* ***p****ermutational* ***A*** *NO****VA***.

The principle of this permutation test is to keep the localities fixed, each preserving its values for the variables under study, so as to preserve the autocorrelation structure. The classification criterion, which consists of the division of the map into subregions, is permuted instead, with the following constraints: each pseudo-region must contain the same number of localities as the original region that it represents; each pseudo-region must remain connected, that is, it must form a continuous surface on the pseudo-map; finally, the pseudo-regions must occupy exactly the whole of the original map, without omission or excess. The program contains two algorithms to resolve the computation problem; the ring algorithm, created by Alain Vaudor, and the random tree algorithm, developed by Junhyong Kim.

Several variables may be analyzed in a single run. The statistic subjected to permutation testing is the sum, for all groups, of the within-group sums of squares (SSQ). After each permutation, the SSQ statistic is recomputed for that pseudo-map. Finally, the SSQ value for the true map is compared to the distribution of the SSQ values obtained for the pseudo-maps. So, this is a one-tailed test and the critical area is the lower tail of the distribution.

If you are using a mainframe version of the program, check the constants at the beginning of the program (CONST statement) to make sure that your problem can be handled. In particular, check the value of MAXLOC (maximum number of localities), MAXGROUPES (maximum number of groups) and MAXVAR (maximum number of variables). You can change these values at will to treat bigger problems. Also, choose the conversation language of the program: LANG = 2 for English.

Input and output files



Besides the files technically called INPUT and OUTPUT, that represent the keyboard and the screen of your terminal or microcomputer, three other input files are necessary to make this program run; two output files are produced. The first input file is the same as in ordinary analysis of variance, that is, the variables to be analyzed and the classification criterion. To take the spatial structure into account, a second file is necessary, which indicates to the program the localities that are neighbors on the map. Finally, if maps are to be plotted, it is necessary to provide the program with a third file giving the geographic coordinates of each locality. In output, a file of statistics may be obtained, as well as a file of maps in the mainframe versions. All these files are written in readable characters (ASCII).

(1) Raw data file

The rows of this file correspond to the localities (objects of the study). The first *N* columns are the *N* variables to be analyzed; the last column contains the classification criterion (geographic group), coded as integers from 1 to the number of groups *k*; this value must be strictly smaller than the MAXGROUPES constant in the list at the beginning of the program, for the VMS/CMS versions. This file, which is called DATAFILE in the PASCAL source code, should be compatible with the input formats of most standard statistical packages, which makes it easy to compute a standard ANOVA for comparison purposes. Program COCOPAN cannot handle missing values; the user must make sure that the localities with missing data have been eliminated from all three input files, or that the missing values have been replaced by estimates obtained by interpolation or other method, prior to this analysis.

(2) List of link edges among localities

That file, which is called LINKS in the PASCAL program, provides a list of link edges among neighboring locality pairs. Each link is represented by a pair of locality numbers, written in free format and separated by one or more spaces. The file may have been prepared by program LINKS (Macintosh version) or program AUTOCOR (VMS or CMS versions); see description of these programs. Since it is written in ASCII, that file may be edited by the user to add or remove link edges, or else it may have been entirely written by him using an ASCII editor. This element of flexibility makes it possible to analyze problems representing a volume rather than a surface; it is only necessary to provide the program with a list of the link edges representing neighboring relations among points in three dimensions.

(3) List of geographic coordinates (X, Y)

That file, which is called COORD in the PASCAL program, contains the list of geographic coordinates (X and Y) of the localities. It is required if one wishes to print the maps, that is, the original map as well as the permuted maps, and also for the computation of the Set Diameter (SD) statistic for each pseudo-group. To make sure that the maps are printed correctly, the coordinates in abscissa must increase from right to left, as longitudes west of Greenwich, and the values in ordinate from bottom to top, as latitudes in the northern hemisphere. Otherwise, the maps may be inverted. The Macintosh version rotates the maps, if necessary, to make them fit the shape of the screen.

(4) File of statistical results

The first results file, which is called STATIS in the PASCAL program, contains the detailed statistical results (below).

(5) File of maps

That file, which is called GRAPHICS in the PASCAL program, is optional and will be produced only if the user requests that maps be printed. It comes out as a separate file in the mainframe versions; in the Macintosh version, maps are produced directly on the screen. The user may then print and examine the original map, as well as the permuted maps (pseudo-maps). See example below.

Questions of the program

The questions posed by the program are described in the following paragraphs.

(1) "How many permutations of the map do you want ?" — The user states how many permutations he wants the program to perform. Since probabilities are computed after including the real-map

is convenient to count one permutation fewer than the total round-number desired. For example, for a total of 1000 statistics in the distribution, one should request 999 random permutations of the regions defined by the classification criterion. Depending on the size of the problem, the speed of the computer and the precision one desires to attain, 249 to 999 permutations are often requested, or more.

(2) "Initialization of the random number generator: Type an INTEGER between 1 and 100." — The pseudo-random number generator is called the specified number of times, before launching the calculations. Each call changes the seed of the generator for the next call. Changing the seed in successive runs is a simple way to get a different set of permuted maps for the same problem.

(3) "Ring method rather then random tree ? (Y or N) (Type Y for ring, or N for random tree)" — The two algorithms are described in Appendix of the COCOPAN reference paper. The respective advantages are the following: the ring algorithm produces less elongated groups, while the random tree algorithm is faster; it has been used to analyze problems involving more than 1300 localities, and this is by no means its practical limit.

(4) If the ring algorithm has been selected, the following question shows up: "Number of times groups can be moved around ? (Recommended: 3-5)" — This corresponds to step 1.3.2 in the description of the algorithm, in the Appendix of the reference paper. After trying the specified number of times, the map will be discarded if it has not been successfully completed.

(5) Similarly, for the ring algorithm: "Until how many missing points do you want to force completing the maps ? (Recommended: 1)" — This corresponds to step 1.3.3 in the description of the algorithm, in the Appendix of the reference paper. If only a few points are missing for the last pseudoarea to be completed (number of points specified here by the user), the program finds the shortest path between empty spots and the group (pseudoarea) in question, and moves everything in between in a chain reaction to force the creation of empty slots adjacent to the group, where it can expand.

(6) "GRAPHICS FILE: Print the maps ? (Y or N)" — The user may elect to have the maps (the original one as well as the permuted maps) printed onto a file, in the VMS or CMS versions, or to have them show up on the screen in the Macintosh version, from where printing can be requested.

(7) "Print a coded representation of the localities in each group ? (Y or N)" — This part of the output is printed into the GRAPHICS file in the VMS or CMS versions, or in the STATIS file in the Macintosh version. It looks as follows:

```
----++++--++++++-+++++----+-----------------------------------   16
---------------------------++++----++++----++++----++++----+++--+++   19
++++----++------+-----+++++---++++---++++---++++----++++---++---   29
------------------------------------------------------------------    0
```

In this example, there are three groups corresponding to the first, the second and the third line, and identified by cardinalities 16, 19 and 29 respectively. The fourth line (cardinality 0) would represent the localities remaining to be assign to groups, in the case of an incomplete map. For each group of the real map (and similarly for each group of each of the pseudomaps), a line of characters are printed, representing all the localities in the study, in their order of appearance in the data file. These characters are pluses (+) for the localities present in the given group, and minuses (-) for those that are not. When there are more than 80 localities in the study, the string of characters for each group requires more than one line; in this case, the first line for all groups is first printed, then the second line for all groups, and so on. Since this coded representation contains all the group positioning information of the real map and of the permuted maps, then it can be used as input to other programs. In this way, the pseudomaps produced by COCOPAN can be employed either to compute additional statistics on the shape or on the distribution of the permuted groups, or to perform other data analyses based on the permuted maps produced by COCOPAN (MANOVA, non-parametric analysis of variance,

discriminant analysis, and so on). This output is of no use in routine runs of COCOPAN.

(8) "Print the group attributed to every locality ? (Y or N)" — This is another form of output that contains likewise all the group positioning information of the real map and of the permuted maps. This output is written into the GRAPHICS file (VMS and CMS versions) or in the STATIS file in the Macintosh version. It presents itself as a single list of all localities, showing the group number attributed to each locality. Here is an example:

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

This example shows the first four localities pertaining to group 3, the following four localities to group 1, and so on; compare with the coded representation of the previous paragraph. This list can be used for the same purposes as the coded form described in the previous paragraph. It is not useful in routine runs of COCOPAN.

(9) "DEBUG: Print all maps and/or lists of bits, even for the rejected maps ? (Y or N) " — This statement applies to the outputs requested in questions (6) to (8) above. This function is available for the Ring algorithm option only. The answer is ‘Yes’ when one wants to know what the rejected maps looked like. It was created to identify causes of problems in the generation of pseudomaps (connection bottlenecks, etc.)

(10) "Width of the maps (in number of characters) ?" — In the VMS or CMS versions of the program, you can choose the width of the maps requested in question (6), according to the terminal you are using and the size of the maps you wish to obtain. These unsophisticated maps are produced on the line printer using standard characters, and their width is calculated in number of characters (see example below). This question does not appear in the Macintosh version, which produces line-drawn maps on the screen, ready to be sent to a Laser printer.

(11) "Number of variables to be analysed ?" — State here the number of variables in the data file, **excluding** the classification criterion written in the last column of that file (see description of the data file, above).

(12) "Set Diameter (great circle distance) ? (Y or N)" — This shape statistic (DE) for the pseudogroups, described in the reference paper, is the diameter of the smallest possible circle that encloses all localities that are members of a group or a pseudogroup. It is computed along the earth's curvature, using the assumption that the X and Y coordinates in the COORD file are expressed in degrees. The diameters are expressed in minutes of arcs (which is equivalent to nautical miles). These statistics allow to compare the diameters of the pseudogroups to those of the original groups that they are intended to represent.

(13) "Probability of Set Diameter ? (Y or N)" — If this option is requested, a table will be printed in the STATIS file giving the probability, for each group, of finding pseudogroups with set diameters smaller than, or equal to the set diameter of the given group on the real map. See the remarks below on the computation of permutational probabilities; see also the example.

(14) "Path Length (graph diameter) ? (Y or N) (Beware: high computing time for large data sets.)" — This second shape statistic (PL) is not described in the above-mentioned reference. Its purpose is similar to the Set Diameter statistic: to compare the diameter of the pseudogroups to that of the original group. The diameter is computed differently this time; it is measured in terms of the smallest number of link edges (see description of the file of link edges, above) that can be counted between the two most distant localities in the group or pseudogroup (distant in terms of numbers of link edges). The message reminds the user that this statistic is very expensive in computing time for large data sets.

(15) "Probability of Path Length ? (Y or N)" — The probabilities are computed as in (13), for the path length shape statistic.

(16) "Statistics for EACH map ? (Y or N)" — If requested here, the Sum of Squares (SSQ) statistics are printed for each group (i) separately [SSQ(i)] and for the whole analysis of variance problem [SSQ = Sum of SSQ(i)], and this for each map (that is, for the true map and for each of the pseudomaps). If requested in (12) and (14) above, the shape statistics SD and PL are printed as well for each map. See example below.

In all cases, even when this option is not selected, a table is printed on the screen **as well as** on the STATIS file. See the example below. For each group, that table shows the probability of finding among the permuted maps an SSQ(i) value smaller than, or equal to the SSQ(i) value of this group on the real map. This information represents an indication of the internal homogeneity of each group on the real map, compared to the homogeneity of connected pseudogroups formed at random on the map and possessing the same number of localities.

The last line of this table (TOTAL) gives the main ANOVA result, that is, the probability of finding among the pseudomaps SSQ values smaller than or equal to the SSQ value of the real map. The table is repeated for each variable under study.

(17) "Statistics about frequency of localities in each group ? (Y or N)" — For each group in turn, a list of all localities is produced on the STATIS file giving the number of times each locality was selected to be attributed to the group in question; for instance (for 500 random permutations):

```
Frequency of data points in group 1
 100  107  101  112  120  124  119  117  107  120  126  136
 139  135  119  115  113  135  141  136  135  127  114  138
 159  148  139  113   97   87  111  157  149  135  125  128
 109  117  148  144  139  124  119  105  122  147  141  136
 130  119  111   94  123  134  145  151  128  127   98  109
 112  127  129  128
Frequency of data points in group 2
 170  166  167  149  130  129  131  132  169  167  147  163
 136  128  133  133  159  148  140  148  134  128  166  164
 134  128  139  133  127  124  176  146  135  123  144  140
 129  172  167  129  125  138  139  143  182  170  163  136
 137  144  148  147  176  176  158  144  154  148  150  185
 187  162  157  148
Frequency of data points in group 3
 230  227  232  239  250  247  250  251  224  213  227  201
 225  237  248  252  228  217  219  216  231  245  220  198
 207  224  222  254  276  289  213  197  216  242  231  232
 262  211  185  227  236  238  242  252  196  183  196  228
 233  237  241  259  201  190  197  205  218  225  252  206
 201  211  214  224
```

In this problem that contains 64 localities, divided in three groups with 16, 19 and 29 localities respectively, this example shows that the first locality in the list has been chosen 100 times (over 500 trials) to be a member of group 1, while it was part of group 2 170 times and of group 3 230 times. In the case of unevenly-connected sets of localities, these lists tell the user whether the placement of the various groups was random or not. See section 3.2 of the reference paper, where this list was used to demonstrate that strongly unevenly connected networks of localities can bias group positioning.

(18) If the random tree algorithm has been selected, the following question is presented: "How many random trees can be aborted before the program is stopped ? Recommended: 10*(N. of permutations)." — It may become necessary in certain problems to increase the limit number of aborted trees to successfully complete a run. This is very unlikely to happen, however; please report any occurrence to us.

Permutation probabilities are computed following Hope (1968). In this method, also recommended by Edgington (1987), the observed value of the statistic is included among the “equals” in the frequency distribution; because of that, it is never possible for none of the values (probability = 0) to be “smaller than or equal to” the observed value. According to Edgington, this computing method introduces a bias but has the advantage of being valid. The precision of the probability computed in this way is the inverse of the number of permutations requested by the user.

Example of commands

The example below illustrates the use of the program on mainframes (CMS or VMS systems; this example was computed on CMS). The calling program first asks the user to identify the various input and output files that will be used; answers are underscored. Then, after the program heading, come the questions posed by the program itself to determine what computing options are requested.

```
Program COCOPAN

What is the name of the main DATA file (variables,
classification criterion)? (defaults are "... data a")
data

What is the name of the file of LINKS among localities?
(defaults are "... data a")
links

What is the name of the file containing the geographic COORDINATES
of the points (requested only if maps are to be printed,
or if the Set Diameter shape statistics are to be computed).
(defaults are "... data a")
coordxy

What is the name of the file where the MAPS are to be printed?
(optional; defaults are "MAPS data a")


What is the name of the output file for the detailed STATISTICS?
(optional; defaults are "STATIS data a")

Execution begins...

P r o g r a m  C O C O P A N  --  M a p

(Contiguity-constrained permutational ANOVA)

Reference:
Legendre, P., N.L. Oden, R.R. Sokal, A. Vaudor and J. Kim. 1990.
   Approximate analysis of variance of spatially autocorrelated
   regional data. J. Class. 7: 53-75.
```

```
Authorship --
Program and Ring algorithm: Alain Vaudor,
Departement de sciences biologiques,
Universite de Montreal,
C.P. 6128, Succursale A,
Montreal, Quebec H3C 3J7.

Random Tree algorithm: Junhyong Kim,
State University of New York at Stony Brook.

How many permutations of the map do you want ?
999
Initialization of the random number generator:
Type an INTEGER between 1 and 100.
10
RING method rather than RANDOM TREE ? (Y or N)
(Type Y for Ring, or N for Random Tree.)
n
GRAPHICS FILE:
   Print the maps ? (Y or N)
n
   Print a coded representation of the localities in each group ? (Y or N)
n
   Print the group attributed to every locality ? (Y or N)
n
Number of variables to be analysed ?
1
STATISTICS FILE:
   Set Diameter (great circle distance) ? (Y or N)
y
   Probability of Set Diameter ? (Y or N)
y
   Path Length (graph diameter) ? (Y or N)
   (Beware: high computing time for large data sets.)
y
   Probability of Path Length ? (Y or N)
y
   Statistics for EACH map ? (Y or N)
y
   Statistics about frequency of localities in each group ? (Y or N)
n
How many random trees can be aborted before the program
is stopped ?  Recommended: 10*(N. of permutations).
10000

Probabilities for SSQ statistics:

 Variable  Group     Smaller    Equal   N. maps  Prob(H0)
     1         A       613        1      1000    0.6140
     1         B       155        2      1000    0.1570
     1         C       277        1      1000    0.2780
     1     total        21        1      1000    0.0220

End of the program.
```

Example of STATISTICS results file

In this file of results, the numbers at the extreme left refer to the same numbers in the "Questions of the program" section above.

```
     C O C O P A N - Statistics file
     The file of link edges contains the following:
        Number of edges      :       203
        Mean number of edges :   3.17187
        Standard deviation   :   1.84493
        Variance of edges    :   3.40377

     SD  stands for Set Diameter (for each group, in min. of arc)
     PL  stands for Path Length (for each group)
     SSQ stands for Sum of SQuares (for each group, then for the whole ANOVA),
         for variables V(1) to V(n)

     N. loc.          16        19        29       [Number of localities in each group]

(16) Map no.    0                                              [Map "0" is the real map]

      SD         402.259   423.800   509.695
      PL               6         6         9
      SSQ v 1    117.726    94.234   188.758   400.719   ["v 1" means "variable 1"]

     Map no.    1

      SD         432.281   564.031   592.866
      PL               7         8         9
      SSQ v 1    136.580    55.592   270.390   462.562

     Map no.    2

      SD         483.161   483.294   467.252
      PL               7         7         6
      SSQ v 1     88.889   161.118   192.508   442.516

     [etc.]

     Map no.  998

      SD         454.418   483.294   636.466
      PL               7         8        11
      SSQ v 1     59.881   210.855   162.238   432.974

     Map no.  999

      SD         432.194   468.058   509.695
      PL               6         8         8
      SSQ v 1    124.092   104.672   190.885   419.649
```

(16) Probabilities for SSQ statistics:

["Equals" include map "0", which is the real map]

Variable	Group	Smaller	Equal	N. maps	Prob(H0)
1	A	613	1	1000	0.6140
1	B	155	2	1000	0.1570
1	C	277	1	1000	0.2780
1	total	21	1	1000	0.0220

(13) Probabilities for SD statistics:

Group	Smaller	Equal	Prob(H0)
1	450	28	0.4780
2	231	30	0.2610
3	140	13	0.1530

(15) Probabilities for PL statistics:

Group	Smaller	Equal	Prob(H0)
1	353	254	0.6070
2	160	190	0.3500
3	604	150	0.7540

Example MAP file

The three maps below illustrate the type of maps that can be produced by the mainframe versions of the program (VMS and CMS), using a line printer. This example shows how the original groups (map 0) are moved around by the algorithm. The Macintosh version produces laser-quality line-drawn maps in which each group of localities is delimited by a contour envelope.

```
Map no.   0                 Map no.   1                 Map no.   2
------------------------    ------------------------    ------------------------
!C  C  B  B  B         !    !B  B  A  A  A         !    !C  C  C  C  C         !
!                      !    !                      !    !                      !
!C  C  C  C  B  B  B   !    !B  B  B  A  A  A  A   !    !C  C  C  C  C  C  C   !
!                      !    !                      !    !                      !
!C  C  C  C  B  B  B  B!    !B  B  B  A  A  A  A  A!    !C  C  C  C  C  C  C  C!
!                      !    !                      !    !                      !
!C  C  C  C  B  B  B   !    !B  B  B  B  A  A  A   !    !B  C  C  C  C  C  C   !
!                      !    !                      !    !                      !
!C  C  C  C  B  B  B   !    !B  B  B  B  A  C  C   !    !B  B  A  A  A  A  C   !
!                      !    !                      !    !                      !
!C  C  C  C  A  B  B  B!    !B  B  C  C  C  C  C  C!    !B  B  B  A  A  A  C  C!
!                      !    !                      !    !                      !
!C  A  A  A  A  A      !    !B  C  C  C  C  C      !    !B  B  A  A  A  A      !
!                      !    !                      !    !                      !
!C  C  A  A  A  A  A  A!    !C  C  C  C  C  C  C  C!    !B  B  B  B  B  A  A  A!
!                      !    !                      !    !                      !
!C  C  C  C  A  A  A  A!    !C  C  C  C  C  C  C  C!    !B  B  B  B  B  B  A  A!
------------------------    ------------------------    ------------------------
```

CONVERT

What does CONVERT do ?

CONVERT as a utility program to convert a SIMIL-type similarity (S) matrix into a distance (D) matrix, or vice versa. The CMS and VMS versions only use the formula

$S_{ij} = 1 - D_{ij}$ or $D_{ij} = 1 - S_{ij}$

while the Macintosh version also allows to convert using formulas

$S_{ij} = \sqrt{[1 - D_{ij}]}$ or $D_{ij} = \sqrt{[1 - S_{ij}]}$

and $S_{ij} = 1 - [(D_{ij} - Dmin)/(Dmax - Dmin)]$ or $D_{ij} = 1 - [(S_{ij} - Smin)/(Smax - Smin)]$

This program has been written because most of the programs in "R" require that the SIMIL-type file provided as input be of the similarity type. So, distance files must be converted in many instances; when distances are larger than 1, the first type of transformation produces negative "similarities"; notice that despite of that, the programs that use these similarity files have been built to handle the values correctly. The last forms of transformation guarantees that all the similarities obtained after transformation are in the [0, 1] interval.

Input and output files

(1) Input file

The input file for program CONVERT is a binary SIMIL-type file. It contains either a similarity or a distance matrix computed by SIMIL, or written by IMPORT-EXPORT (Macintosh version) or IMPORT (CMS and VMS versions).

(2) Output file

The output file is also a binary SIMIL-type matrix containing the transformed matrix. The converted binary file contains mention "(CONVERT)" in the block of information which is automatically printed out by several programs. For instance, one can use program LOOK that allows to read these information as well as the content of the resemblance matrix:

```
TITLE:  Similarity matrix
DATE 20/02/91
FUNCTION  s15
(CONVERT)
NUMBER OF OBJECTS : 57
NUMBER OF DESCRIPTORS :  3
```

Example

(The user's answers are in **boldface type**)

```
What is the name of the input SIMIL file? (defaults are "... data a")
fichier_s15

What is the name of the output matrix file? (defaults are "... data a")
fichier_dist

Execution begins...

 P R O G R A M   C O N V E R T to convert an S matrix into D
                                      or a D matrix into S
 VERSION 3.0b
 TRANSFORMATION SIMILARITY <-> DISTANCE
 AUTHOR: A. VAUDOR

 End of the program.
```

*EXPNTS*CMS

What does EXPNTS do ?

From a SIMIL-type resemblance matrix, this program produces a new binary file containing a distance matrix that can be used by the NT-SYS package for multivariate analysis (Numerical Taxonomy SYStem: Rohlf *et al.*, 1971).

The NT-SYS package includes a nonmetric multidimensional scaling program, called MDSCALE. Since this very useful method of analysis is not available in the "R" package, program EXPNTS makes it possible to transfer to NT-SYS — on IBM mainframe only — the similarity or distance files computed using "R". Indeed, many of the resemblance measures available in R's SIMIL program are not available in NT-SYS. The users of NT-SYS on MS-DOS machines will use EXPORT (in the CMS or VMS versions of "R") or IMPORT-EXPORT (Macintosh version), instead of EXPNTS, since the MS-DOS version of NT-SYS uses ASCII distance files only.

The NT-SYS package, developed by Prof. F. James Rohlf, is distributed by *Exter Software Inc.*, 100 North Country Road, Bldg. B, Setauket, New York 11733, USA (available versions: for mainframes or MS-DOS machines).

Input and output files

(1) Input file

The input file to EXPNTS is a binary SIMIL-type file produced by programs SIMIL, IMPORT (CMS and VMS versions) or IMPORT-EXPORT (Macintosh version); the internal structure of the SIMIL-type binary resemblance matrices is described in the section devoted to program SIMIL. In other words, the input file to EXPNTS is the output file of SIMIL. This file contains the resemblance matrix, written in binary form. Program EXPNTS automatically reads in the header of the SIMIL-type file what the number of objects is.

(2) Output file

The output file is a new binary file containing a distance matrix in a form compatible with the NY-SYS package. It contains the same upper triangular matrix of distances as the input file, but diagonal terms equal to zero have been added. That file contains no header, contrary to the SIMIL-type files. One cannot read that binary file using LOOK, nor using an ASCII editor or a word processor.

Questions of the program

The calling program simply asks for the name of the input and output files. The program itself asks a single question: "`Transformation Similarities-Distances ?`" — If the input file contains a similarity file, this question offers the possibility of transforming the similarities into distances ($D = 1 - S$), since program NT-SYS expects a distance matrix as input file.

EXPORT CMS/VMS

What does EXPORT do ?

Program EXPORT makes it possible to transform binary resemblance matrices produced by SIMIL into square ASCII matrices (readable characters). EXPORT fills one of the functions of program IMPORT-EXPORT of the Macintosh version. Such square ASCII matrices may be useful in the following situations, for instance:

- to present the resemblance values in publications,
- to pass the SIMIL-type resemblance matrices to other programs or packages,
- or, when it becomes necessary to transfer resemblance matrices from one type of computer to another, since SIMIL-type binary matrices cannot be used directly on computers with different representations for floating-point numbers. The square matrices produced by EXPORT can be transformed back to the SIMIL-type binary format using programs IMPORT (CMS and VMS versions) or IMPORT-EXPORT (Macintosh version) if it is intended to use them in conjunction with one of the "R" programs on the new host computer.

Input and output files

The input file is SIMIL-type resemblance matrix (similarities, distances, or dependence measures among variables). The internal structure of the SIMIL-type binary resemblance matrices is described in the section devoted to program SIMIL. The output file is a square symmetric ASCII file (readable characters) written in format (8F10.7), with values 1.0000000 on the main diagonal. These values may easily be recognized using an ASCII editor and converted to 0.0000000 values if necessary.

Questions of the program

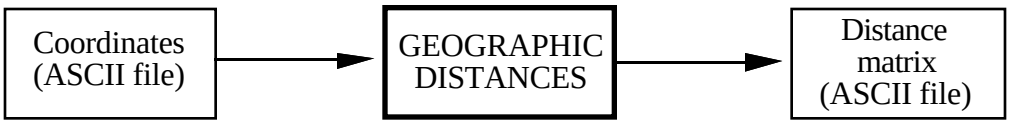
The calling program simply asks for the name of the input and output files. The program itself asks no question. A square ASCII file such as produced by EXPORT is presented at the end of the section on program IMPORT-EXPORT.

*GEOGRAPHIC DISTANCES*Macintosh *or DIST*CMS/VMS

What does GEOGRAPHIC DISTANCES do ?

Starting from a file of geographic coordinates for a set of localities, this program computes geographic distances among these localities *following earth's curvature.*

Input and output files



(1) File of coordinates (input)

The file of geographic coordinates of the localities is a rectangular ASCII file whose rows correspond to the localities while its columns are the *latitude* and *longitude* coordinates (with *latitude* first). Coordinates may be written in several possible ways, described in the **Options** section below.

If the set of localities are located on both sides of longitude 0, coordinates west of Greenwich may be written either with a negative sign (-), or using the 360° system. Latitudes of points south of the equator must be written with a minus sign (-); of course, if all the localities in the study are in the southern hemisphere, negative signs are useless.

(2) Matrix of geographic distances (output)

The output file is written in ASCII, so it can be read directly using an ASCII editor or a word processor. The geographic distance matrix is square, with zeros on the diagonal. Furthermore, in the Macintosh version, the first line of the output file reproduces the name of the input file of coordinates. Programs IMPORT (in the CMS or VMS versions) or IMPORT-EXPORT (in the Macintosh version) can be used to create a binary SIMIL-type matrix from that square matrix. The computed geographic distances can be expressed in one or the other of the units mentioned in the **Options** section.

Options

Options available for the input file are described by the following menu, presented by the program:

```
0: decimal degrees
1: degrees period minutes (ex.: 35.04)
2: degrees space minutes (ex.: 35 04)
3: degrees period minutes period seconds (ex.: 35.04.05)
4: degrees space minutes space seconds (ex.: 35 04 05)
```

The following options are available for the output file:

```
0: distances in radians
1: distances in degrees
2: distances in nautical miles (or minutes of arc)
3: distances in miles
```

IMPORTCMS/VMS

What does IMPORT do ?

This program makes it possible to import resemblance matrices, transforming them from square ASCII files to the binary SIMIL-type format required by the other programs of the "R" package. IMPORT fills one of the roles of program IMPORT-EXPORT in the Macintosh version. The square ASCII matrices to be imported may have been written by other packages or programs. They may also have been computed by SIMIL on one type of computer, then converted to an ASCII format by program EXPORT, before being transferred to a different machine where they are converted back to the SIMIL-type format using IMPORT. Finally there are cases, such as behavioral, sociological or molecular genetics studies, where the raw data present themselves in the form of association matrices among individuals; such matrices may be written down directly on a file using an ASCII editor, then imported using IMPORT to be analyzed by the programs in the "R" package.

Input and output files

The input file contains a square matrix written in ASCII (readable using an ASCII editor or a word processor), produced in one of the ways described above. The output file is a binary SIMIL-type file containing the same information. The structure of the binary SIMIL-type matrices is described with program SIMIL.

Questions of the program

The calling program first asks the name of the input and output files; then the program itself asks the following:

```
SIZE OF THE MATRIX (number of objects or of variables)
```

The answer to this question is a single positive integer, corresponding to either the number of objects (for a Q-mode matrix) or the number of descriptors (for an R-mode matrix) being compared in that resemblance matrix. The next question is:

```
INITIAL NUMBER OF OBJECTS if this is a correlation matrix;
If not, repeat the size of the matrix.
```

Finally the program asks for a title. These pieces are used to complete the block of information which is automatically part of any SIMIL-type binary matrix; see the example in the description of program IMPORT-EXPORT, below. The number of objects, for an R-mode resemblance matrix (covariance or correlation), is needed by some of the other programs, such as our program for computing partial correlations (not included for the time being in the currently distributed version of "R"), in order to compute the tests of significance correctly. In the other cases, the number given as the answer to that question is simply recorded in the information block without being used by the data analysis programs. Notice also that in the information block, item FUNCTION will state that the binary SIMIL-type file has been produced by program IMPORT.

IMPORT-EXPORT Macintosh

What does IMPORT-EXPORT do ?

Program IMPORT-EXPORT allows to import resemblance matrices, transforming them from ASCII to the SIMIL-type binary format required for them to be used by the "R" data analysis programs. It can also perform the opposite operation, transforming SIMIL-type binary resemblance matrices to ASCII files. This programs plays the same role in the Macintosh version as both IMPORT and EXPORT in the CMS and VMS versions.

Input and output files

Binary SIMIL-type as well as ASCII resemblance matrices can be used, either as input or as output files to this program. The internal structure of the SIMIL-type binary resemblance matrices is described in the SIMIL program chapter. ASCII resemblance matrices may be presented under different formats, described below.

Options

The first question of the program concerns the type of conversion requested by the user: either from left to right in the above diagram, or from right to left.

```
Conversion choice
  From ASCII (characters) to SIMIL type
  From SIMIL file to ASCII file
```

If one chooses to convert a SIMIL-type binary file into an ASCII file (characters readable using an ASCII editor or a word processor), the program only requests the name of the input file, as well as the name to be given to the ASCII output file. The output file is a square, symmetric ASCII matrix with 1.0000000 on the diagonal. These values may easily be converted into 0.0000000, if need be, using an ASCII editor.

If the user has chosen to convert an ASCII file into the binary SIMIL-type format instead, the program requests information about the size (*i.e.*, either the number of objects or the number of descriptors being compared) as well as the shape of the matrix:

```
Input file:                                  [Menu: Give the name of the input file]
  Size of the matrix                         [Give a single integer number]

Matrix type:
  Square with diagonal
  Square without diagonal
  Upper triangular with diagonal
  Upper triangular without diagonal
```

Finally, the program requests two pieces of information that would have been readily available

had the resemblance matrix been computed by SIMIL from raw data: the size of the original data matrix in the other dimension, as well as the title to be included in the block of information of the file:

```
Number of objects (in Q mode) or descriptors (R mode)          [An integer]
Title                                    [Give a title of no more than 80 characters]
```

Explanation — Consider a data table of *n* lines and *p* columns. If the resemblance measures in the matrix have been computed among the lines of the data table, then the program now wants to know the number of columns. If on the contrary the resemblances have been computed among the columns of the raw data table, the number of lines must now be provided. That piece of information is first used to complete the block of information which is automatically attached to all SIMIL-type binary files:

```
INPUT FILE: A square resemblance matrix
     TITLE: Hydrology of 32 James Bay lakes
     DATE: 2/5/91
     FUNCTION:  (ImpExp)
     Number of objects: 32
     Number of descriptors: 10
```

(Note that line FUNCTION indicates that the file has been created by program IMPORT-EXPORT.) The information about the number of objects, for an R-type resemblance matrix (covariance or correlation), is needed by some other programs, like our program for computing partial correlations (not included for the time being in the version of "R" sent to you), in order for the tests of significance to be computed correctly.

Example

The following upper triangular matrix, without diagonal, measures travel distances in km among 6 Québec cities. The numbers have been aligned to facilitate reading by humans, although this is not requested for machine reading (which is accomplished in free format). The values may be integers, or else real numbers with or without figures before the decimal period (.138, 0.138 or -.57 are admissible).

```
198 368  57  882 311
    549 238 1063 482
        311  517  80
             824 253
                 594
```

After transformation into a binary SIMIL-type matrix (which cannot be printed here), followed by back-transformation into an ASCII matrix, the following is obtained:

```
1.0000000  198.00000 368.00000  57.00000  882.00000 311.00000
198.00000  1.0000000 549.00000 238.00000 1063.00000 482.00000
368.00000  549.00000 1.0000000 311.00000  517.00000  80.00000
 57.00000  238.00000 311.00000 1.0000000  824.00000 253.00000
882.00000 1063.00000 517.00000 824.00000  1.0000000 594.00000
311.00000  482.00000  80.00000 253.00000  594.00000 1.0000000
```

Of course, in this road distance matrix, the "1" values on the diagonal do not make sense and should be replaced by zeros if they impair the following computations. Value "1" has been chosen because it is the most appropriate in two frequent situations: first, for all similarity matrices, and then for all correlation matrices. The 1.0000000 values are easy to spot for an ASCII editor since these are the only 7-decimal values in the ASCII output file.

INTERLNK CMS/VMS

What does INTERLNK do ?

Program INTERLNK computes a proportional-link linkage agglomerative clustering. The degree of connectedness of the clustering solution (*Co*), which is determined by the user, may vary from 0 to 100%; this represents the whole array of solutions from single linkage (*Co* = 0) to complete linkage (*Co* = 1). Around 50% connectedness, the clustering results approximately preserve the metric properties of the reference space; for small connectedness values, there is a contraction of the reference space that produces chaining, while high connectedness values produce the reverse effect, space dilation around the clustering nuclei (Lance & Williams, 1967).

The INTERLNK calling program launches three different programs in turn: (1) a sorting program, which rewrites the similarity in order of decreasing similarity values; (2) the clustering program itself; and finally (3) the program that plots the dendrogram. The user may request this third program to compute various clustering statistics (chain of primary connections, cophenetic correlations, Gower's distance, efficiency coefficients) which have been described in the chapter dealing with program CLUSTER. INTERLNK exists only in the CMS and VMS versions of "R".

Input and output files

(1) Input file

The input file must imperatively contain a similarity matrix, NOT a distance matrix, written by programs SIMIL or IMPORT; INTERLNK exists only in the CMS and VMS versions. A distance matrix may easily be converted into similarities with the CONVERT utility program.

The maximum number of objects that can be clustered by this program, as well as the maximum number of groups that may exist simultaneously at any one clustering step, are fixed by parameters MAXDIM and MAXGROUPES respectively, at the beginning of the program. These parameters may be changed to accommodate large problems, before compiling the program.

(2) Output file

The output file contains the dendrogram describing the agglomerative clustering results, as well as the clustering statistics. These have been described in detail in the chapter dealing with program CLUSTER. If identifiers have been provided for the objects in the raw data file submitted to SIMIL (10 first characters), the dendrogram shows these identifiers, instead of the order numbers which are otherwise attributed to the objects by the program.

Questions of the program

After the calling program has requested the name of the input and output files, the clustering program itself only asks what connectedness level (*Co*) will have to be used by the proportional-link linkage agglomerative algorithm.

```
CONNECTEDNESS ?
```

The connectedness may take values from 0 (single linkage) to 1.0 (complete linkage). The answer to that question is a real number between 0 and 1. For fractional numbers, the PASCAL language requests the user to type "0.75" instead of ".75", for instance.

The next questions are asked by program DENDRO, which draws the dendrogram and computes the clustering statistics; see the description of these statistics in the chapter on program CLUSTER. The width of the dendrogram to be drawn is determined by the user, who must tell the program <u>how many printing characters</u> should be used to draw the dendrogram. To the question

```
WIDTH OF THE DENDROGRAM IN CHARACTERS (MINIMUM 10, MAXIMUM 279)
```

the answer given must be an integer between 10 and 279, depending on the width of the screen or of the page available for printing. Notice that the width provided here only concerns the dendrogram itself; to this one must add 12 characters on the left for the object names and the dendrogram margin, and 10 characters on the right for the fusion levels (see example below).

Example

The example below, computed under CMS, is the result of a proportional-link linkage with *Co* = 0.5, for the 5 pools already used in the chapter on program CLUSTER to illustrate the clustering statistics. The cophenetic correlation (Pearson's *r*) is 0.94680. On the left of the dendrogram are found the object identifiers. If identifiers have not been provided when the similarity matrix was computed, the clustering program attributes sequential numbers 1 to *n* to the objects. Each fusion level (expressed as <u>distances</u>), shown on the right, corresponds to the vertical line that **begins** immediately on its left and goes downwards. For instance, the vertical line identified by the arrow has the value D = 0.40000 shown on the right.

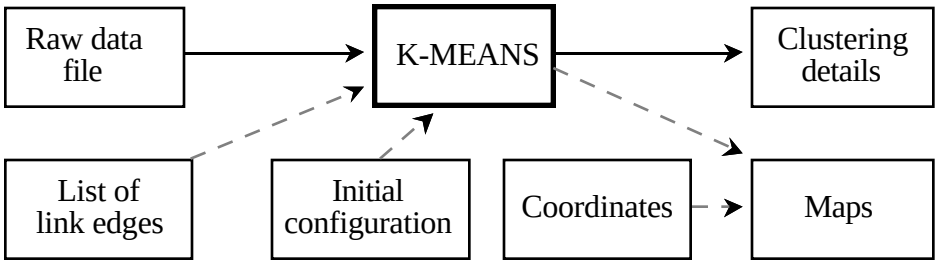
```
P R O G R A M   D E N D R O  to plot dendrograms.

Version 3.0b

NUMBER OF OBJECTS :    5                      [Block of information concerning
NUMBER OF VARIABLES:    8                         the input similarity matrix]
TITLE: 5 pools of Legendre & Chodorowski (1977)
DATE 03/03/91
FUNCTION  s20

D E N D R O G R A M
                                                     LEVEL(D)
--------------------------------------------------------------


POOL 214   |<----------------                          0.40000
                            |<----------------------   0.93700
POOL 212   |<----------------                      |
                                                   |<
POOL 432   |<--------------------                  |   0.50000
                                |<--------         |   0.70000
POOL 431   |<--------------------        |         |
                                         |<---------
POOL 233   |<-----------------------------
```

*K-MEANS*Macintosh *or* *KMEANS*CMS/VMS

What does K-MEANS do ?

Program K-MEANS computes a non-hierarchical clustering by minimization of the within-group variance, following several variants of the method originally proposed by MacQueen (1967), to which he gave the name *k-means*. This is a partitioning method for a group of objects, and not a method of hierarchical classification. The user decides how many groups, *k*, she wants to obtain from the program. The *k*-means algorithm followed here is the one described on page 163 of Anderberg (1973). The present program computes the clustering with or without contiguity constraint (spatial or temporal), following the user's request. It complements the hierarchical clustering programs of the "R" package, which implement various algorithms of clustering without constraint (CLUSTER in the Macintosh version, INTERLNK and LANCE in the CMS and VMS versions) or with contiguity constraint (BIOGEO and CHRONO).

Input and output files



Dashed arrows indicate optional files.

(1) Raw data file

Contrary to the other clustering programs of this package which require a similarity matrix as their main input file, data are given to the K-MEANS program in the form of a rectangular ($p \times n$) raw data file, in which the rows correspond to the objects and the columns are the variables, without row or column identifiers. For example:

23.4	12.4	3.2	77	22.6
12.6	13.2	4.9	44.1	23.6
33.4	11.8	5.5	55.3	21
45.1	12	3.1	109	22.8
50.7	11.7	4.6	67.9	23.5

There are other *k-means* algorithms that use a distance matrix instead of raw data as their main input file. The present program cannot handle missing values; these must be filled before clustering, using one or another of the interpolation methods; another possibility consists of eliminating the row corresponding to the object(s) with missing values.

N.B. Actually, the program minimizes the sum of squares of the *Euclidean* distances to the centroid of the objects of their respective group. If the method is to be applied to data for which the Euclidean distance is considered inappropriate (as it is the case for species abundance data containing many zeros), one can proceed through the following steps instead (see also the example):
1) Compute a similarity or distance matrix of one's choice using program SIMIL.
2) Compute a principal coordinates analysis of that matrix.

2) Compute a principal coordinates analysis of that matrix.
3) Ask program PCOORD to save some or all of the principal coordinates on a file (generally, 10 or 15 principal coordinates are enough to account for most of the variability).
4) This file of principal coordinates may now be given to the K-MEANS program as the new raw input data file.

(2) File of link edges (optional)

If the user decides to compute the partitioning with spatial contiguity constraint, a file of link edges has to be provided besides the raw data file, as in program BIOGEO. See the example in the description of that program. That file may have been produced either by AUTOCOR (in the CMS and VMS versions) or by LINKS (in the Macintosh version). The following example corresponds to the rook's connection scheme among 12 localities forming a regular grid of 3 rows and 4 columns; each link edge is represented by a pair of numbers:

```
 1  2     2  3     3  4     5  6     6  7     7  8     9 10    10 11
11 12     5  1     6  2     7  3     8  4     9  5    10  6    11  7
12  8
```

(3) File of spatial coordinates (X, Y) (optional)

If one wants to ask the program to draw maps for each clustering level, which is an option of the program, one must provide a list of the coordinates of the point locations. That list of coordinates is used solely for the purpose of drawing the maps. The coordinates are provided as an ASCII file of integer or real numbers; the coordinates must not be in the degree-minute-second form. The number of coordinates in that file must correspond to the number of objects in the study. When using the CMS or VMS versions, don't forget to put a zero before the decimal mark (for instance, write "0.376" instead of ".376").

In certain cases, a more didactic representation may be obtained by imposing coordinates that do not exactly correspond to the geographic positions. For instance, to analyze in a single run repeated observations (through time) of a set of localities, the output maps may be organized in such a way that the various time slices come out as a mosaic of separate parts in the final picture. This may be done because the coordinates provided in the file of spatial coordinates are used solely to draw the maps; the spatial or spatio-temporal relationships that are taken into account during the clustering process are those described in the file of link edges only.

(4) Initial configuration (optional)

Only the raw data file is necessary to compute a partitioning without contiguity constraint. In many cases, however, the user wants to provide a file containing one or several initial configurations of the objects to increase the performance of the algorithm and prevent it from falling in a local minimum of the objective function (see options 1b and 2b); that option is also used when the K-MEANS program is run to optimize a partition obtained from an agglomerative program. The initial configuration file presents itself in the form of a list of objects attributed to each group. NOTICE: each group list must end with a zero. If several initial configurations are provided in the file, these are treated one after the other by the program. For instance, to test two initial configurations for a problem containing four groups and a total of 13 objects, the initial configuration file could be the following:

```
1 7  3  12  0
8  2  0
10  13  4  5  0
6  9  1  0                                   [end of the first initial configuration]

2  4  10  0
```

```
9  1  3  13  0
12  5  6  0
7  8  1  0                    [end of the second initial configuration]
```

The program checks whether all objects have been assigned to a group; if this is not the case, the user is asked, conversationally, to assign the missing objects to a group. If an object has erroneously been assigned to more than one group, the last assignment is retained.

(5) File of results

The results are presented as a list of objects for each group. In the CMS and VMS versions, results of constrained partitionings are not presented in the form of maps, while they are by the Macintosh version. The program may be asked to list only the initial and final configurations, or else the intermediate steps as well. Besides the list of group members, the value of the sum of squared distances to the centroid (E) is shown for each group, as well as the value of statistic D (which is the sum of the E values) for the whole solution. See the section describing the contents of the results file for more details.

Options of the program

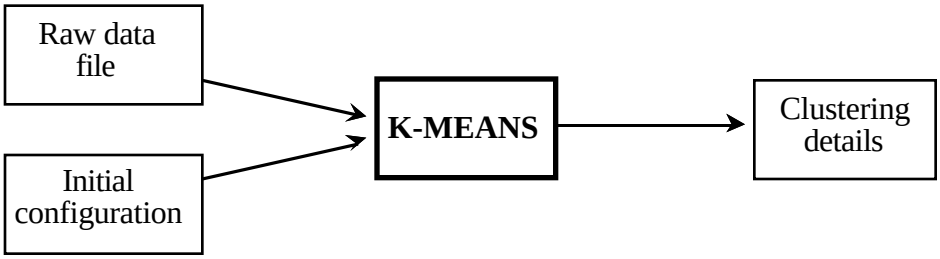
The main difficulty with this method is to establish an initial configuration of the objects, or in other words, an initial division of the objects into *k* groups, choosing a configuration which will be sufficiently close to the optimal solution (*i.e.*, the solution minimizing the sum of within-group sums of squares) so that the algorithm has a fair chance of converging towards it. Solutions to this problem which are available in the program are the following:

(1) Clustering without constraint

For clustering without contiguity constraint, three options are available.

1a) The so-called “Stony Brook” method is the one privileged by R. R. Sokal at that university. In that method, the program performs *N* iterations, each one starting from a different random allocation of the objects to the *k* groups. Statistic D is computed for each iteration, and the solution that minimizes D is to be kept as the final solution. D is the sum, over all groups, of the sums of squared distances to the group centroids (Späth, 1980, p. 73).

1b) An initial configuration of the user’s choice may be provided. The way to do it is described above. An initial configuration presumably close to the optimal one may have been obtained from another clustering or ordination program; this is certainly the quickest and most efficient method to prevent getting locked in a local minimum of the D function. In other cases, the initial solution to be tested may be known by hypothesis.

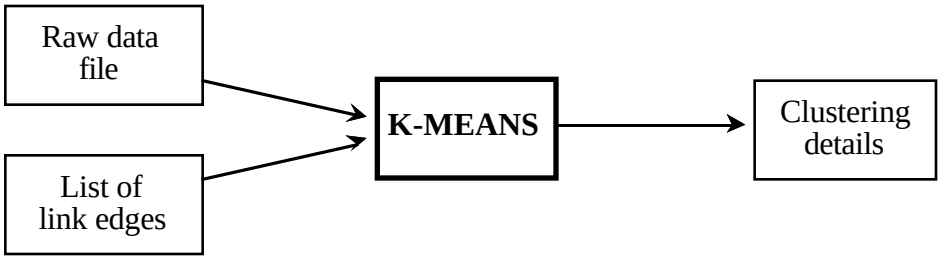


1c) The MODULUS method (Späth, 1980, p. 67), where the program creates an initial configuration by putting object 1 in group 1, object 2 in group 2, … , object *k* in group *k*, object *k* + 1 in group 1, etc.

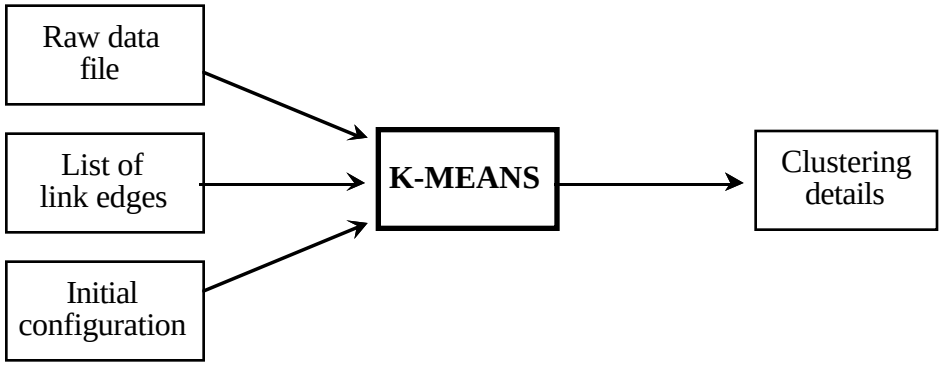
(2) Clustering with constraint

For clustering with contiguity constraint, if the constraint is one-dimensional (time series, or spatial transect), the user only has to say so as answer to one of the questions of the program; the algorithm will then assume that successive objects in the input file are adjacent in space or time. If on the contrary the objects are spread in a space with two geographic dimensions or more, a list of constraints is provided in the form of a file of link edges, as in program BIOGEO. The two following solutions are available to establish the initial configuration.

2a) The Stony Brook method, as in (1a) above.



2b) Your own initial configuration, known by hypothesis or obtained from another clustering or ordination program, as in (1b) above. Because of the very nature of its algorithm (minimization of the sum of within-group variances), this program may be useful to determine with greater accuracy the position of the group borders obtained by the constrained agglomerative clustering program BIOGEO.



Examples

The two examples below illustrate the use of the program to compute a partitioning with spatial contiguity constraint. In the first case, the clustering is computed from 10 random initial configurations.; see (1) [numbers refer to the numbers in the left-hand margin of the examples below.] In the second example (2), a file containing two initial configurations is provided to the program. The calling program, whose dialog is reproduced below (examples computed under CMS), requests the names of the various files; the answers of the user are underscored and in bold. The questions of the Macintosh version are essentially the same, although their precise formulation may sometimes differ slightly.

The data analyzed below are the same that served to illustrate the use of program BIOGEO. Since the K-MEANS program requires a rectangular raw data file (objects x variables), the similarity matrix analyzed by BIOGEO has been run through the principal coordinates analysis program PCOORD, which computed the position of each object in Euclidean space (see note at the end of the section about the raw input data file); the first two principal coordinates only were kept, because they were the only ones to correspond to positive eigenvalues. Since the principal coordinates analysis has produced a Euclidean representation of the cloud of points, and since we want K-MEANS to minimize the sums of squares of these very Euclidean distances to the group centroids, the program is requested (3) to perform no transformation of the data. On the contrary, if we had been dealing with a series of variables possessing different physical dimensions, we would have requested the program to standardize the variables, following the same logic as in principal components analysis.

Example 1: from 10 random configurations

```
Kmeans
WARNING! This program cannot handle missing values.
What is the name of the main data file (rows = objects,
cols = variables)? (defaults are "... data a")
file  pcoord  a

What is the name of the file of LINKS among localities (if any) ?
(defaults are "... data a")
file  links  a

What is the name of the file containing the starting configuration '
(if any) ?  (defaults are "... data a")

What is the name of the file where the results are to be stored?'
(optional; defaults are "CONSTRKM OUT a")
file  res1  a

Execution begins...
 P R O G R A M   K - M E A N S  with constraints

 Author: Alain Vaudor

 Number of objects
57
 Number of variables
2
 Number of groups
4
 Type of clustering:
   0: Clustering without constraint
```

```
       0: Clustering without constraint
       1: Clustering with contiguity constraint in 1 dimension
       2: Clustering with general contiguity constraints (LINK file requested)
     2
      Options:
       1: At random (Stony Brook method)
       2: Your own file of initial configuration(s)
(1)  1
      Number of iterations ?
(1)  10
      Options:
       1: Print all intermediate results
       2: Print results only for initial and final configurations
     2
      Options:
       0: No data transformation
       1: Transformation to standardized variables
(3)  0
     Type a small integer for the random number generator
     5
      End of the program.
```

Example 2: from a file containing two initial configurations

```
     Kmeans
     WARNING! This program cannot handle missing values.
     What is the name of the main data file (rows = objects,
     cols = variables)? (defaults are "... data a")
     file  pcoord  a

     What is the name of the file of LINKS among localities (if any) ?
     (defaults are "... data a")
     file  links  a

     What is the name of the file containing the starting configuration '
     (if any) ?  (defaults are "... data a")
(2)  file  init  a

     What is the name of the file where the results are to be stored?'
     (optional; defaults are "CONSTRKM OUT a")
     file  res2  a

     Execution begins...
      P R O G R A M   K - M E A N S  with constraints

      Author: Alain Vaudor

      Number of objects
     57
      Number of variables
     2
      Number of groups
     4
      Type of clustering:
       0: Clustering without constraint
```

```
        1: Clustering with contiguity constraint in 1 dimension
        2: Clustering with general contiguity constraints (LINK file requested)
     2
      Options:
        1: At random (Stony Brook method)
        2: Your own file of initial configuration(s)
(2)  2
      Number of iterations ?
     2
      Options:
        1: Print all intermediate results
        2: Print results only for initial and final configurations
     2
      Options:
        0: No data transformation
        1: Transformation to standardized variables
(3)  0
      End of the program.
```

Contents of the file of results, example 2

The outputs presented below have been produced by the Macintosh version of the program. Versions CMS and VMS produce an identical ASCII output file, but they are unable to draw the maps.

First initial configuration:

The first initial configuration proposed to the program divided the objects in four groups, as follows; the end of each group is marked by a zero:

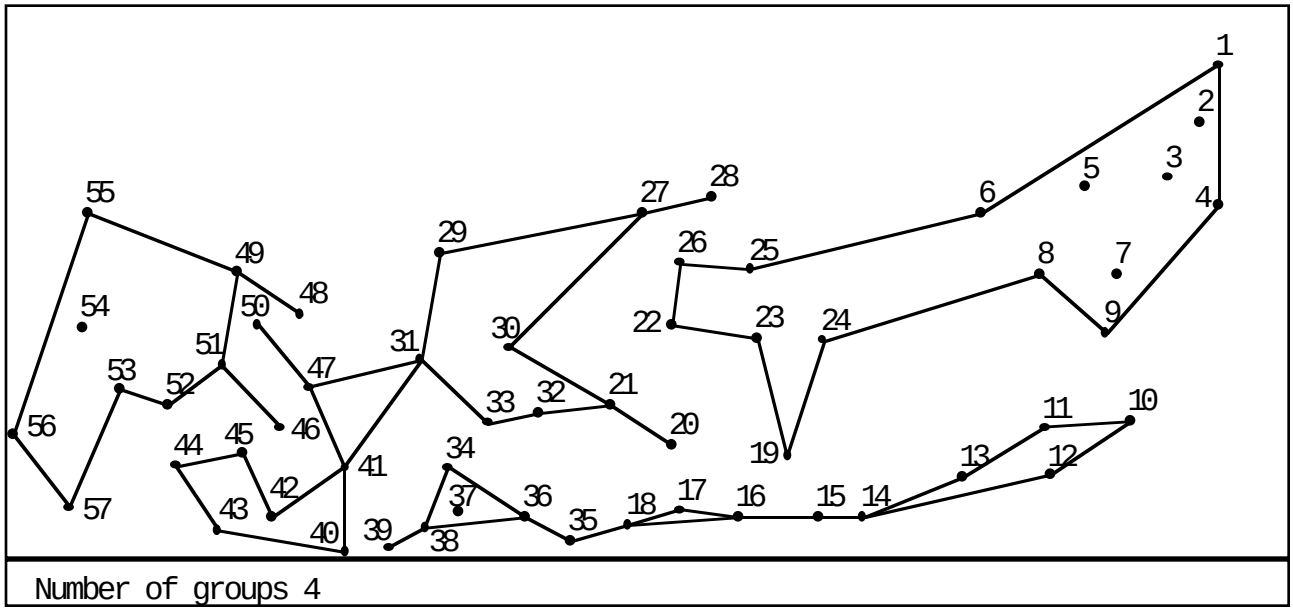
```
42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57  0
22 23 24 25 26 27 28 29 30 31  0
14 15 16 17 18 19 20 21 32 33 34 35 36 37 38 39 40 41  0
 1  2  3  4  5  6  7  8  9 10 11 12 13  0
```

Referring to the map below, one sees that according to this hypothesis, the localities would be divided in four blocs of about the same size: the first one to the left, the next two in the center (upper and lower parts), and the last one to the right. The D statistic, which represents the sum, for the various groups, of the sums (E) of squared distances to the group centroids, has the value 11.72596 for the initial configuration; by moving the objects among groups, the algorithm has been able to reduce the value of D (or: Sum of the E statistics) to 8.35474. We will see in the next example that this result is still quite remote from the optimal value; the present example has been presented in order to show that *k-mean*-type algorithms may often fail to converge towards the overall minimum value of statistic D, depending on the initial configuration they have to start from.

```
 Iteration no. 1

   Configuration:  initial
      Group   1: 42  43  44  45  46  47  48  49  50  51  52  53  54  55  56
                 57
              E =   3.68762
      Group   2: 22  23  24  25  26  27  28  29  30  31
              E =   1.75314
      Group   3: 14  15  16  17  18  19  20  21  32  33  34  35  36  37  38
                 39  40  41
```

```
                39  40  41
             E =   3.84957
   Group   4:  1   2   3   4   5   6   7   8   9  10  11  12  13
             E =   2.43562
   Sum E   =       11.72596
Configuration:  1
   Group   1: 40  43  44  46  48  49  51  52  53  54  55  56  57
             E =   2.75096
   Group   2: 19  20  21  22  23  27  28  29  30  31  32  33  41  42  45
                47  50
             E =   2.35024
   Group   3: 12  14  15  16  17  18  34  35  36  37  38  39
             E =   2.48375
   Group   4:  1   2   3   4   5   6   7   8   9  10  11  13  24  25  26
             E =   2.44078
   Sum E   =       10.02573
Configuration:  2
   Group   1: 46  48  49  51  52  53  54  55  56  57
             E =   1.48061
   Group   2: 20  21  27  28  29  30  31  32  33  40  41  42  43  44  45
                47  50
             E =   2.21594
   Group   3: 10  11  12  13  14  15  16  17  18  34  35  36  37  38  39
             E =   2.82830
   Group   4:  1   2   3   4   5   6   7   8   9  19  22  23  24  25  26
             E =   1.82989
   Sum E   =        8.35474
```

The map produced by the program is shown hereunder. Each group is surrounded by an envelope; the individual points shown within an envelope, for instance 2, 3, 5 and 7, are members of the same group as 1, 4, 6, etc.



Second initial configuration:

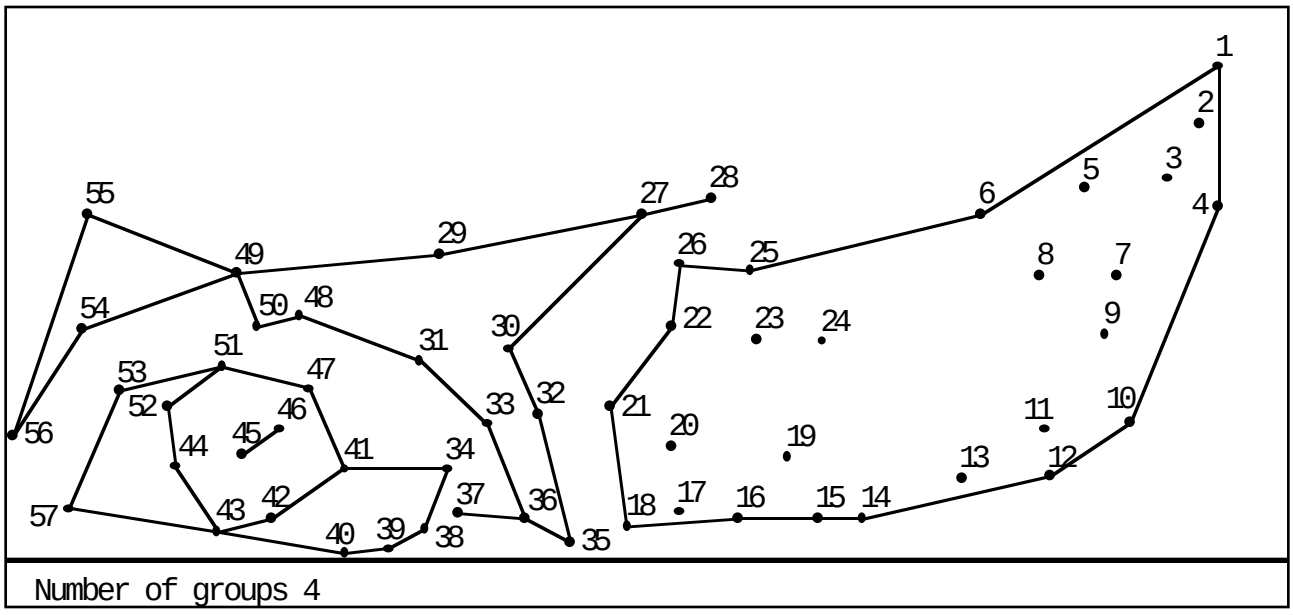
The second initial configuration given to K-MEANS is the four-group solution produced by program BIOGEO; see the last map of section “Contents of the file of results”, in chapter BIOGEO:

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26  0
27 28 29 30 31 32 33 35 36 37 48 49 50 54 55 56  0
45 46  0
34 38 39 40 41 42 43 44 47 51 52 53 57  0
```

This initial configuration produces a D statistic (or: Sum of E) = 7.71485. Program K-MEANS has not succeeded in finding a lower value by object interchange among the groups.

```
Iteration no. 2

  Configuration:  initial
     Group   1:  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15
                16  17  18  19  20  21  22  23  24  25  26
             E =   4.48868
     Group   2: 27  28  29  30  31  32  33  35  36  37  48  49  50  54  55
                56
             E =   0.87511
     Group   3: 45  46
             E =   0.17951
     Group   4: 34  38  39  40  41  42  43  44  47  51  52  53  57
             E =   2.17156
     Sum E   =        7.71485
  The initial solution could not be improved.
```

The map produced by the program is the following. It shows the same assignments to groups as the last map (“4 groups”) presented in the example of program BIOGEO. Notice that group (45, 46) lies inside the “doughnut” formed by the two envelopes (outside and inside) drawn around group (34 to 57). **N.B.** — To increase the resolution, simply use the mouse to draw a box around any part of the picture you wish to blow up.



*LANCE*CMS/VMS

What does LANCE do ?

Program LANCE computes agglomerative clusterings following the Lance & Williams general algorithm (1966, 1967). Methods available through this algorithm include single and complete linkage, unweighted arithmetic average (UPGMA), weighted arithmetic average (WPGMA), unweighted centroid (UPGMC), weighted centroid (WPGMC), as well as the family of methods known as flexible clustering. Ward's (1963) method of minimum-variance agglomerative clustering has recently been added to the program, following the recommendation of Everitt (1980). For any method other than single or complete linkage, or UPGMA, UPGMC, WPGMA, WPGMC and Ward, the program asks the user to provide the four parameters $alpha_j$, $alpha_m$, *beta* and *gamma* which determine the clustering strategy to be used by the Lance & Williams algorithm. Table 2 provides the values of these parameters in the various cases. References in the table heading may be consulted for more information on the role of these parameters in the clustering strategy.

Table 2 — Values of parameters $alpha_j$, $alpha_m$, *beta* and *gamma* of the general Lance & Williams (1966) equation, for the various combinatorial types of sequential agglomerative clustering. Inspired from Sneath & Sokal (1973), Legendre & Legendre (1984a), and Jain & Dubes (1988).

Clustering method	Parameters of the combinatorial model α_j	α_m	β	γ
Single linkage	0.5	0.5	0	-0.5
Complete linkage	0.5	0.5	0	0.5
Average clustering:				
Unweighted arith. average (UPGMA)	$n_j/(n_j+n_m)$	$n_m/(n_j+n_m)$	0	0
Weighted arithmetic average (WPGMA)	0.5	0.5	0	0
Unweighted centroid (UPGMC)	$n_j/(n_j+n_m)$	$n_m/(n_j+n_m)$	$-\alpha_j\alpha_m$	0
Weighted centroid (WPGMC)	0.5	0.5	-0.25	0
Flexible clustering	$[\alpha_j + \alpha_m + \beta = 1;\quad \alpha_j = \alpha_m;\quad -1 \le \beta \le 1]$			0
Ward's method	$(n_j+n_g)/(n_j+n_m+n_g)$	$(n_m+n_g)/(n_j+n_m+n_g)$	$-n_g/(n_j+n_m+n_g)$	0

The LANCE calling program launches three different programs in turn: (1) a sorting program, which rewrites the similarity in order of decreasing similarity values (necessary for the *a posteriori* clustering statistics); (2) the clustering program itself; and finally (3) the program that plots the dendrogram. The user may request this third program to compute various clustering statistics (chain of primary connections, cophenetic correlations, Gower's distance, efficiency coefficients) which have been described in the chapter dealing with program CLUSTER. Program LANCE exists only in the CMS and VMS versions of "R".

Input and output files

(1) Input file

The input file may contain either a similarity or a distance matrix, written by programs SIMIL or IMPORT (since program LANCE exists only in CMS and VMS versions).

The maximum number of objects that may be analyzed by this program is determined by parameter MAXNOBJ, at the beginning of the program. MAXNOBJ may be changed to accommodate larger problems; this requires recompiling the program, though.

(2) Output file

The output file contains the dendrogram describing the agglomerative clustering results, as well as the clustering statistics. These have been described in detail in the chapter dealing with program CLUSTER. If identifiers have been provided for the objects in the raw data file submitted to SIMIL (10 first characters), the dendrogram shows these identifiers, instead of the order numbers which are otherwise attributed to the objects by the program.

Questions of the program

After the calling program has requested the name of the input and output files, the clustering program itself only asks which clustering method should be used. If the user chooses option 6, the values of parameters $alpha_j$, $alpha_m$, *beta* and *gamma* needed by the program will have to be provided. See table 2.

```
DO YOU WANT
     1 - Unweighted arithmetic average clust. (UPGMA)
     2 - Weighted arithmetic average clust.   (WPGMA)
     3 - Unweighted centroid clustering       (UPGMC)
     4 - Weighted centroid clustering         (WPGMC)
     5 - Ward's minimum variance clustering
     6 - Other combinatorial clustering methods
```

This is the only clustering program in the “R” package that allows using either a similarity or a distance matrix as the input file. The next question allows to determine which type of matrix it is:

```
IS THE INPUT FILE A SIMILARITY OR A DISTANCE FILE? (Write S or D)
```

To this question, one must answer by a letter: *S* or *s* for a similarity matrix, *D* or *d* for a distance matrix. Notice however that the normal functioning of this program requires a similarity matrix; with a distance matrix, the adjustment to the computations does not extend to the statistics associated with the dendrogram, which is computed by another program. The following warning is given to the user who has chosen to work from a distance matrix:

```
Reading the DISTANCE matrix.
```

```
Reading the DISTANCE matrix.
The a posteriori tests are all correct only for
SIMILARITY matrices. If you request them nevertheless,
 - the minimum spanning tree will then be incorrect;
 - cophenetic correlations will have the wrong sign;
 - the Gower distance will be incorrect;
 - entropy measures will be correct.
```

The next questions are asked by program DENDRO, which draws the dendrogram and computes the clustering statistics; see the description of these statistics in the chapter on program CLUSTER. The width of the dendrogram to be drawn is determined by the user, who must tell the program how many printing characters should be used to draw the dendrogram. To the question

```
WIDTH OF THE DENDROGRAM IN CHARACTERS (MINIMUM 10, MAXIMUM 279)
```

the answer given must be an integer between 10 and 279, depending on the width of the screen or of the page available for printing. Notice that the width provided here only concerns the dendrogram itself; to this one must add 12 characters on the left for the object names and the dendrogram margin, and 10 characters on the right for the fusion levels (see example below).

Note on Ward's method

Ward's (1963) variance minimization method clusters objects or groups in such a way as to minimize the sum of squared distances to the centroids of the groups. The computations in Lance & Williams' general agglomerative algorithm are done on **squared distances** D^2. Dendrograms may be presented in different ways, depending on the authors. Jain & Dubes (1988) directly use the fusion levels obtained from the clustering algorithm, expressed as squared distances, as the horizontal scale of their dendrograms. Everitt (1980) uses instead a statistic of sum, over the various groups, of the sums of squares of the distances to the group centroids, called E.S.S. in Everitt's book; this statistic may also be computed as the sum, over the k groups, of the values $e_k^2 = \Sigma(D^2)/n_k$. Jain & Dubes' scale is simply a linear transformation of that used by Everitt. Finally, the SAS (1985) manual recommends using one of the following statistics as the horizontal scale of the dendrogram: either Everitt's E.S.S. statistic divided by the total sum of squares, which produces proportions of variance (SAS manual, 1985, p. 267); or, the "semipartial R-squared", which is the between-cluster sum of squares divided by the total sum of squares (SAS manual, 1985, p. 272 and 281); these are again linear transformations of Everitt's or of Jain & Dubes' scales.

Notice that all the above-mentioned measures are essentially squared distances. In the LANCE program, we use instead the **square root** of the squared fusion distances given by the combinatorial Lance & Williams algorithm and used by Jain & Dubes. There are two advantages to this. On the one hand, this produces a better-looking dendrogram (better balanced) than the above-mentioned methods. On the other, this is the most appropriate distance when one wants to compare the cophenetic matrix to the original distance or similarity matrix, using matrix correlation or Gower's distance.

Example

The example below, computed under CMS, is the result of Ward's clustering applied to the 5 pools, already used in the chapter on program CLUSTER to illustrate the clustering statistics. Four new variables were obtained by principal coordinates analysis of the similarity matrix (coefficient S20) computed from the original data; Euclidean distances were then computed among the objects (pools) for these new variables, in order to illustrate the answers given by the program when the input file contains a distance matrix.

```
What is the name of the SIMILARITY matrix file? (defaults are "... data a")
pools  d1  a

Name of the OUTPUT file for the dendrogram and the tests?
(defaults are "RESULT listing a")
pools  dendr-d1  a

Execution begins...                    Execution of the sorting program begins
Execution begins...                    Execution of the clustering program begins

 P R O G R A M   L A N C E -- General agglomerative clustering model.

 Version 2.2b (Modified for SIMIL 3.0 / Includes Ward)
 Author: A. VAUDOR

 DO YOU WANT ');
      1 - Unweighted arithmetic average clust. (UPGMA)
      2 - Weighted arithmetic average clust.   (WPGMA)
      3 - Unweighted centroid clustering       (UPGMC)
      4 - Weighted centroid clustering         (WPGMC)
      5 - Ward's minimum variance clustering
      6 - Other combinatorial clustering methods
5

 IS THE INPUT FILE A SIMILARITY OR A DISTANCE FILE? (Write S or D)
d
 Reading the DISTANCE matrix.
 The a posteriori tests are all correct only for
 SIMILARITY matrices. If you request them nevertheless,
  - the minimum spanning tree will then be incorrect;
  - cophenetic correlations will have the wrong sign;
  - the Gower distance will be incorrect;
  - entropy measures will be correct.

 End of the clustering.

Execution begins...        Execution of the program that draws the dendrogram begins

 P R O G R A M   D E N D R O

 Dendrogram, minimum spanning tree, tests during clustering
 Version 3.0b
 AUTHOR: A. VAUDOR

 DO YOU WANT THE MINIMUM SPANNING TREE? ( y or n)
y
 DO YOU WANT THE A POSTERIORI TESTS:
 COPHENETIC CORRELATIONS, GOWER'S DISTANCE AND ENTROPY ?
n
 WIDTH OF THE DENDROGRAM IN CHARACTERS (MINIMUM  10, MAXIMUM 279)
50
 End of the program.
```

<u>Contents of the file of results</u>

On the left of the dendrogram are found the object identifiers. If identifiers had not been provided when the similarity matrix was computed, the clustering program would have attributed sequential numbers 1 to *n* to the objects. Each fusion level (expressed as <u>distances</u>), shown on the right, corresponds to the vertical line that **begins** immediately on its left and goes downwards. For instance, the vertical line identified by the arrow has the value D = 0.50000 shown on the right.

```
P R O G R A M   D E N D R O  to plot dendrograms.

Version 3.0b

NUMBER OF OBJECTS :    5
NUMBER OF VARIABLES:    4
TITLE: 5 pools of Legendre & Chodorowski (1977)
DATE 03/03/91
FUNCTION  d01

D E N D R O G R A M
                                                                 LEVEL(D)
-------------------------------------------------------------------------

POOL 431   |<------------------                                  0.50000
                              |<------------                     0.81854
POOL 432   |<------------------            |
                                           |<-----------------   1.29596
POOL 233   |<-------------------------------                 |
                                                             |<
POOL 212   |<--------------                                  |   0.40000
                          |<----------------------------------
POOL 214   |<--------------
```

[Block of information concerning the input resemblance matrix]

LINKS Macintosh

What does LINKS do ?

Program LINKS computes a variety of connecting schemes among localities that are neighbors in space (in 1 or 2 dimensions) and writes the resulting link edges into a file. Several spatial analysis programs use these files as their source of information about the neighboring relationships that exist among localities; this is the case of program AUTOCOR for spatial autocorrelation analysis, the spatially constrained clustering programs BIOGEO and K-MEANS, as well as the COCOPAN analysis of variance program. Program LINKS exists only in Macintosh version; most of its functions are available, for the CMS and VMS versions, in program AUTOCOR.

When the points form a regular grid on the map, it is easy to link the first neighbors using simple connecting schemes whose names are derived from the game of chess (Cliff & Ord, 1981): rook's (square), bishop's (diagonal), or king's connections (also called queen's: both square and diagonal).

When the localities are positioned in an irregular manner, geometric connecting schemes may be used, such as the Gabriel's connection (Gabriel & Sokal, 1969), the Delaunay triangulation (Dirichlet, 1850; Upton & Fingleton, 1985), or the relative neighborhood graph. There exists an inclusion relationship among those connecting schemes: all links that obey the relative neighborhood graph definition are also members of the Gabriel graph, which in turn are all included in the Delaunay triangulation.

Relative neighborhood graph ⊃ Gabriel graph ⊃ Delaunay triangulation

Input and output files

(1) File of coordinates

To obtain a Delaunay triangulation, a Gabriel graph or a relative neighborhood graph, one must provide the program with a list of the geographic coordinates of the localities. Each row of that file must contain two pieces of information, as follows:

X coordinate Y coordinate

The coordinates must be written as integer or real numbers (*i.e.*, decimal numbers) and not as degrees-minutes-seconds. These data are read by the program in free format; in other words, the number of blank spaces before or after each number does not matter. In the case of a regular grid of localities, no file of coordinates is needed.

(2) Output: File of link edges

This ASCII file contains a list of link edges among pairs of neighbors, as permitted by the connecting scheme (option) that has been used to run the program. Each connecting edge is represented by the numbers of the two localities linked by it. An example follows, corresponding to a regular grid of 4 rows and 5 columns (20 localities), king's connection:

```
 1   2     2   3     3   4     4   5     6   7     7   8     8   9     9  10
11  12    12  13    13  14    14  15    16  17    17  18    18  19    19  20
 6   1     7   2     8   3     9   4    10   5    11   6    12   7    13   8
14   9    15  10    16  11    17  12    18  13    19  14    20  15     6   2
 7   3     8   4     9   5    11   7    12   8    13   9    14  10    16  12
17  13    18  14    19  15     7   1     8   2     9   3    10   4    12   6
13   7    14   8    15   9    17  11    18  12    19  13    20  14
```

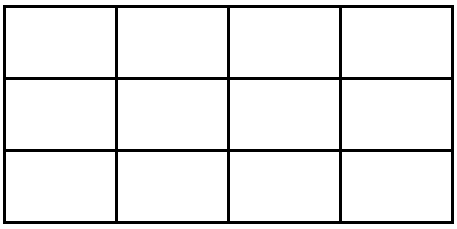
Notice that the user may edit that ASCII file; link edges may be added or removed, as requested by the study.

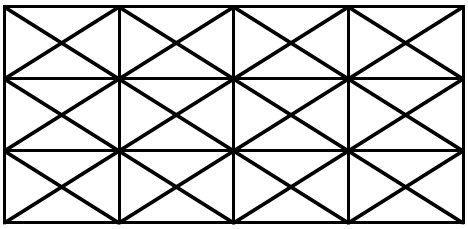
Options

The presentation of the program's options also contains examples.

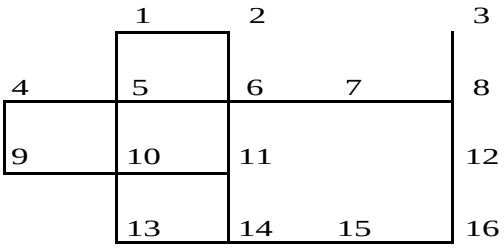
(1) Regular grid

When the localities are located at the vertices (intersections) of a regular grid, it is not necessary to provide a list of coordinates. The program first asks what is the size of the grid (how many columns and rows), and then what connections are requested: horizontal, vertical, positive slope, negative slope. The picture on the left shows a rook's connection (horizontal and vertical edges only), while the picture on the right presents a king's connection scheme (link edges in all four directions). For the right-hand example, the list of link edges is shown above, the localities being numbered by rows, from 1 to 20, as one reads a book.





Given a regular grid of a certain size, the program offers the possibility of removing some points from the grid. One must first indicate how many points must be eliminated, and then identify them, assuming that the points are numbered from left to right in each row, and the rows are read from top to bottom. For instance, points no. 1, 4, 14 and 16 could be eliminated from the 20-point grids above in order to obtain the following scheme (rook's connection) containing only 16 points:

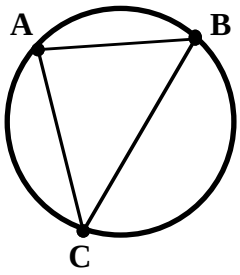


(2) Transposing the axes

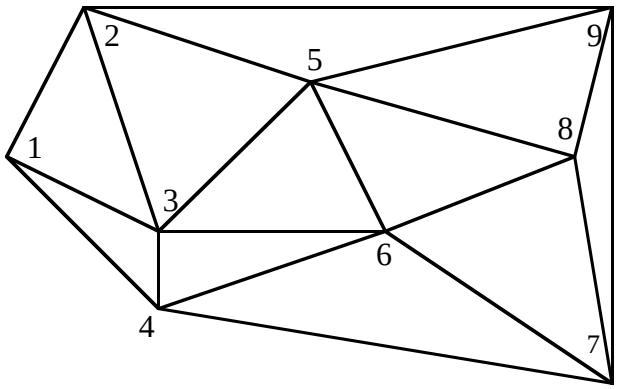
After reading in the file of coordinates of the localities, the program plots the points on the screen. The user may then transpose the abscissa (ordering the points from right to left instead of left to right) as well as the ordinate (ordering the points from top to bottom instead of bottom to top). From there on, the new ordering will be kept.

(3) Delaunay triangulation

The Delaunay triangulation criterion (Dirichlet, 1850; Upton & Fingleton, 1985) is as follows. Given any triplet of points A, B and C, the triangle uniting these points will be included in the triangulation if and only if the circle (shown left) passing through the three points includes no other point in the set under study. For example, the file of coordinates shown in the center will give rise to the triangulation on the right (N.B. do not include the point identifiers in your file):



Point identifiers	Coordinates	
1	0	3
2	1	5
3	2	2
4	2	1
5	4	4
6	5	2
7	8	0
8	7.5	3
9	8	5



This triangulation includes the following 19 link edges:

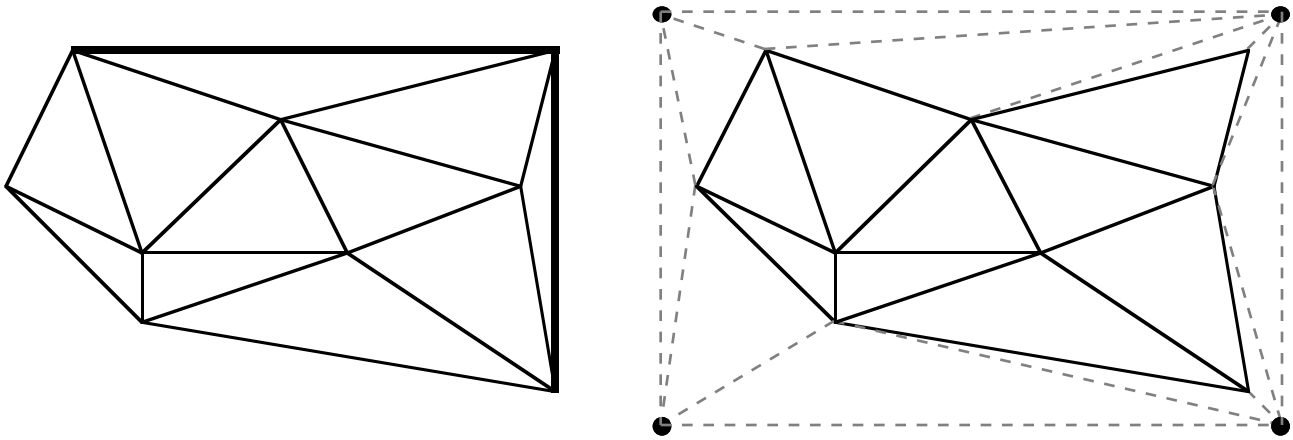
```
1   4     1   2     1   3     2   3     2   9     3   4     4   7     5   3
5   6     3   6     2   5     5   9     4   6     5   8     6   8     6   7
7   9     7   8     8   9
```

Long edges may be created in the periphery of a cloud of points, simply because the sampling design includes no other points located farther (border effect); for example, edges 2 - 9 and 7 - 9 above could well not have been kept, had the cloud of points been larger. One could always edit the file of link edges and eliminate the edges (pairs of numbers) linking peripheral objects located too far from one another. Another possibility is to ask the program to take care of that operation for us. To do so, “constraints” are imposed onto the cloud of points. These constraints are supplementary points, located judiciously and included in the analysis, whose presence prevents the formation of the long unwanted peripheral edges; the link edges between these supplementary points and the real points are not written down in the file of link edges.

Supplementary points may be provided to the program in two different ways. The manner chosen is indicated in answer to the following question of the program:

```
Constraints
- Rectangular constraints
- No constraint
- Constraints in the input file
```

Rectangular constraints — Program LINKS contains an algorithm that automatically generates rectangular “constraints”. For the points of the example above, whose Delaunay solution is repeated below (left), the two bold lines are those that are eliminated by the constraints. The algorithm first includes four supplementary objects at the corners of an imaginary rectangular frame, slightly larger than the set of points under study; these supplementary objects are represented by dark points in the graph on the right. Computing the triangulation, the supplementary points form connections with the real object-points: the presence of these edges (dashed lines) prevents the formation of the two bold edges on the left. Edges connected to supplementary points are not included in the list of link edges.



Constraints in the input file — The user may also add "constraints" into the input file; these additional points, judiciously positioned, are described in the input file by their X and Y coordinates, just like the real object-points of the analysis. If, for instance, 6 supplementary "constraint" points had been included in the input file *after* the 9 real object-points, one would have told the program that there are 9 real points in the analysis; then, following an additional question of the program after specifying that the constraint points are in the input file, one would have told the program that there are also 6 "constraint" points in the file.

No constraint — No additional point is included in the computation of the triangulation. It remains possible to edit the file of link edges, using an ASCII editor or a word processor, and eliminate the edges (pairs of numbers) among distant peripheral objects, if need be.

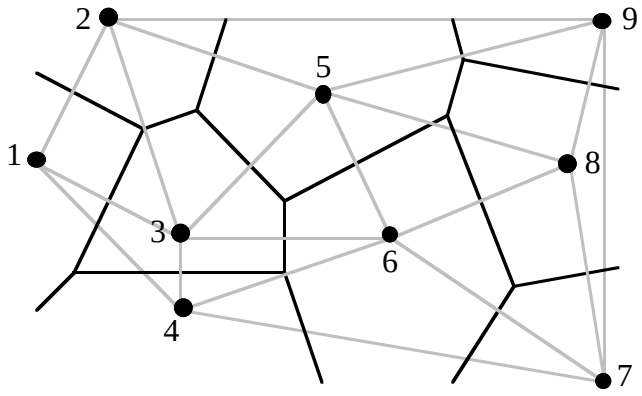
(4) Influence polygons

It may be interesting to determine the geometric zone of influence of each point. The zone of influence of an object-point **A** includes all other points of the surface that are closer to **A** than to any other object-point in the study. The zones of influence so defined have the shape of polygons, also called *tiles*, *tessellae* or *tesserae* (singular: *tessella* or *tessera*). The resulting picture is called a *mosaic* or *tessellation* (adjective: *tessellated*); it is often referred to as a *Dirichlet tessellation* (1850), or as *Voronoï polygons* (1909) or *Thiessen polygons* (1911), from the names of the authors who first described these mathematical structures.

These polygons may easily be constructed from the Delaunay triangulation, since they form its logical complement (dual). One only has to find the perpendicular bisector of each segment in the triangulation; the intersection of the perpendicular bisectors delimits the polygons (tiles). Upton & Fingleton (1985) as well as Isaaks & Srivastava (1989) propose various applications of these tessellations in spatial analysis. The program offers the following choices to the user:

```
Choice of lines
- Triangulation only
- Polygons only
- Triangulation and polygons
```

For the following picture, the choice was "Triangulation and polygons". The Delaunay triangulation is in gray while the Dirichlet tessellation is in black. The object-points, with numbers, are near the middle of each tile, but not necessarily at their centroids (mass centers). The reason is that the position of the line separating two neighboring tiles depends on how far the two nearest neighbor points are in that direction.



Different options are offered to the user in the pull-down menu “Graphs”:

```
Graphs
- Write number of links
- Print the graph
- Draw on PICT file
- Write surfaces
- Finish
```

That menu is accessible with all the connecting scheme options; option “Write surfaces”, however, is available only after a Dirichlet tessellation has been computed. This is the reason why that menu is presented here.

Write number of links — The number of link edges that have been written in the file is shown on the screen. Since some of the programs that use the file of link edges will ask how many edges there are, it is good practice to include that number in the name of the file of link edges.

Print the graph — The graph is sent to the available printer. In particular, and since laser printers as well as photocopying machines can print on transparencies, the picture may be reproduced on a transparency, which can be overlaid on an existing map.

Draw on PICT file — The picture is saved in PICT format in a file, whose name is provided by the user. That file may be read back by anyone of the Macintosh graphics programs, such as MacDraw, SuperPaint, etc. In this way, the picture may be edited before printing, or it may be incorporated into a word processor’s file (MacWrite, Word, LaserWriter, etc.). Most of the pictures presented in that section have been produced in that way.

Write surfaces — It may be useful to know the area of each polygon. Measures of area may be written in a file (example below), whose name is provided by the user. Surface areas are given in the same (squared) units as the original coordinates. In some cases, the peripheral tiles are closed, even if their limits lie outside the area reproduced in the graph (which depends on the size of the screen). In other cases, the peripheral tiles are open; a note to that effect is printed in the file below.

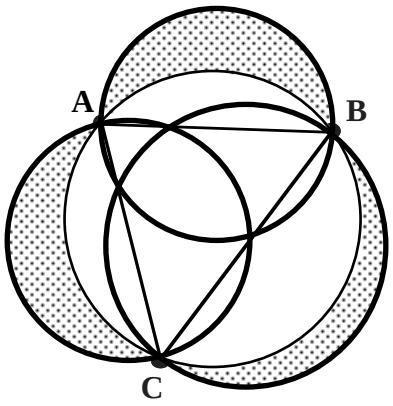
```
1 Open
2 Open
3   4.62500
4 Open
5  15.49844
6  10.06532
7 Open
```

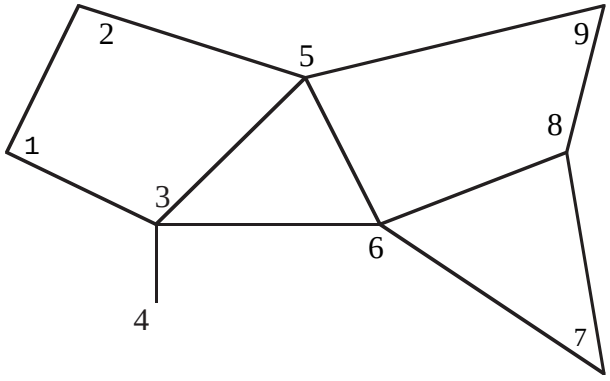
```
8  11.95391
9 Open
```

(5) Gabriel graph

The criterion of the Gabriel graph (Gabriel & Sokal, 1969) differs from that of Delaunay in the following way. Let us connect two points **A** and **B**. That link edge will be part of the Gabriel graph if and only if no other point **C** lies inside the circle whose diameter is that line. In other words, the link edge between **A** and **B** is kept as part of the Gabriel graph if $D^2_{\mathbf{A,B}} < D^2_{\mathbf{A,C}} + D^2_{\mathbf{B,C}}$ for any other point **C** in the study ($D^2_{\mathbf{A,B}}$ representing the square of the geographic distance between points **A** and **B**). Another way to express that criterion is the following: if **Center** represents the middle point between **A** and **B**, then the line connecting **A** to **B** is kept as part of the Gabriel graph if $D_{\mathbf{A,B}}/2 < D_{\mathbf{Center,C}}$ for any other point **C** in the study.

The example below (right) represents the Gabriel graph for the same points as above. It is easy to see that the 12 link edges forming the Gabriel graph are a subset of the 19 lines kept in the Delaunay triangulation (above). Indeed, as can be seen in the picture on the left, the circles (bold) corresponding to the Gabriel criterion may contain, in the shadowed zones outside the Delaunay circle (light), some object-points that the Delaunay criterion circle does not encompass; this is why some link edges authorized by the Delaunay criterion are excluded by Gabriel's.

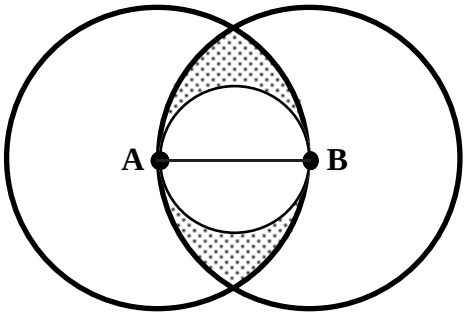


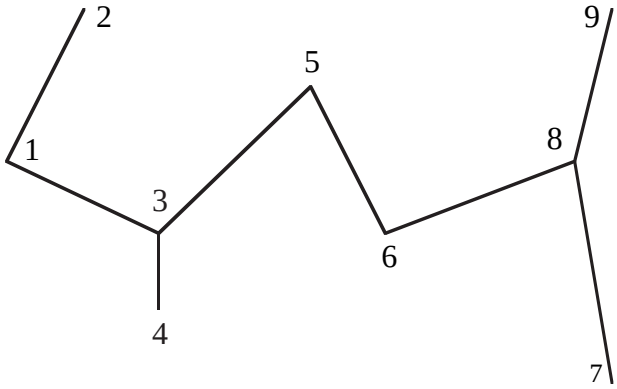


In this option, the problem of "constraints" (see above) does not exist, because the long connecting lines that could form at the periphery of the cloud of points are automatically eliminated by Gabriel's criterion.

(6) Relative neighborhood graph

The relative neighborhood graph criterion differs from Gabriel's criterion in the following way. Let us connect two points **A** and **B**; draw a first circle centered over **A** and a second one centered over **B**, each having the line from **A** to **B** as its radius. That line will be part of the graph if no other point **C** in the study is contained in the intersection of the two circles. In other words, the line from **A** to **B** is kept as part of the relative neighborhood graph if and only if $D_{\mathbf{A,B}} \leq \max(D_{\mathbf{A,C}}, D_{\mathbf{B,C}})$ for any other point **C** in the study. The example below (right) represents the relative neighborhood graph for the same points as in the examples above. One can easily see that the 8 connecting lines (8 = number of objects - 1) forming the relative neighborhood graph are a subset of the 12 lines forming the Gabriel graph (above). Indeed, as can be seen in the picture on the left, the intersection of the two circles (bold) corresponding to the relative neighborhood criterion may contain, in the shadowed zone, some object-points that the Gabriel criterion circle (small circle, light) does not encompass; this is why some link edges authorized by the Gabriel criterion are excluded by that of the relative neighborhood graph.


A
B

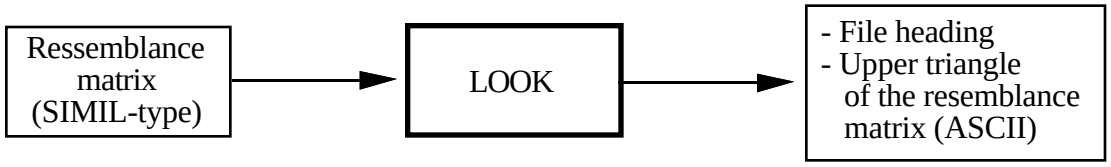

1
2
3
4
5
6
7
8
9

LOOK

What does LOOK do ?

Program LOOK allows to look at the binary files written by SIMIL, IMPORT (CMS and VMS versions) or IMPORT-EXPORT (Macintosh version); these matrices cannot be read directly because they are written in binary and not in ASCII. Since the user is often interested only to look at the block of general information at the beginning of the file (title, date, function, number of objects, number of descriptors), a first window, in the Macintosh version, presents only that information. After that, the user is invited to request that the binary file be copied into an ASCII file, if he so desires. In the output file, after the block of information of the file, only the upper triangle of the resemblance matrix is printed, together with the object names identifying the rows and columns.

Input and output files



(1) Input file

The input file is a binary file of similarities, distances, or dependence measures among descriptors, written by programs SIMIL, IMPORT (CMS and VMS versions) or IMPORT-EXPORT (Macintosh version). In the Macintosh version, these files are represented by a triangular icon with the word SIMIL.

(2) Output file

The output file contains two types of information. First, the block of general information at the beginning of the file (title, date, function, number of objects, number of descriptors); this is followed by the upper triangular matrix of resemblance measures. The diagonal is not written; depending on whether the measures of resemblance are similarities, distances, or measures of dependence among descriptors, the diagonal implicitly assumes values 0 or 1. The object identifiers are written on the left and above the half-matrix; if no identifier has been provided when the file was computed by SIMIL, object sequence numbers are printed instead.

If the user wants to obtain a full square, ASCII representation of the matrix, and not only the upper triangle, in order to use it in subsequent analyses, program EXPORT (CMS and VMS versions) or IMPORT-EXPORT (Macintosh version) should be used instead of LOOK.

Questions of the program

In the CMS and VMS versions, the questions of the program concern only the names of the input and output files.

After presenting on the screen the block of information concerning the input file, the Macintosh version asks the user if the resemblance matrix should be copied onto the output file. In many cases, indeed, one only wishes to read the heading information. When the program is done with the file, it presents the question: “Another file to process ?” If this is the case, the user may choose the file from the menu. One clicks “Cancel” to indicate that there are no more files to process.

Example

Here is an example of how to use the program on mainframes; that example has been run under CMS.

```
Enter name of the SIMIL file (fn ft fm; defaults are ... DATA A)
locality  d1  a

Enter name of the output file (fn ft fm; defaults are MAT DATA A)
locality  output  a

Execution begins...
```

Contents of the file of results

Here is an example of the output file. First, the block of information concerning the input file is copied (title, date, function, number of objects, number of descriptors), followed by the half-matrix of resemblance.

```
P R O G R A M   L O O K
to look at a SIMIL binary matrix

VERSION Mac 3.0
Author: A. Vaudor

INPUT FILE : 7 localities/D1
    TITLE: 7 localities of Québec / Distances in km
    DATE: 7/9/91
    FUNCTION: D01
    Number of objects : 6
    Number of descriptors : 2

                    i
                   m                                                  e
                  i                a        l                 i        m
                 t                h        a                 k        ô
                u                s        é        c        s        r
               o        é        a        r        e        u        é
              c        p        u        t        b        o        J
             i        s        g        n        é        m        -
            h        a        i        o        u        i        t
           C        G        M        M        Q        R        S
Chicoutimi          485.60595 349.75490 378.53672 180.15916 185.63113 368.84501
Gaspé                       161.00831 781.74651 551.23544 302.05421 792.79199
Miguasha                              621.47124 392.58719 165.79426 633.74548
Montréal                                        233.77626 502.99525  44.29501
Québec                                                    269.67142 241.57762
Rimouski                                                            506.65426
```

MANTEL

What does MANTEL do ?

Program MANTEL computes Mantel's (1967) *Z* statistic between two similarity or distance matrices, as well as the derived forms described below: partial Mantel tests, Mantel correlogram. The significance of Mantel's statistic may be evaluated either by permutations, or through the normal approximation described by Mantel (who called that statistic *t*; its distribution is asymptotically normal). Since the probability obtained from the normal approximation rapidly converges towards the permutational probability, it becomes useless to use the permutational test when it would be more costly in computer time, that is, with problems containing many objects. Legendre & Fortin (1989) present a brief account of the Mantel permutational test.

Notice that the value of the *Z* statistic computed by this program is half that of Mantel (1967), because the computations are done on half-matrices of similarities or distances; Mantel's *t* statistic, however, as well as Hubert's standardization of *Z*, are computed as if the matrices were square. The standardized Mantel statistic (*r*) is not affected by computations on half-matrices.

Besides its applications in spatial analysis, the Mantel test is useful in a number of other statistical situations. Hubert *et al.* (1982), as well as McCune & Allen (1985), Burgman (1987), Hudon & Lamarche (1989) and Legendre & Fortin (1989), have used it to test the goodness-of-fit of models to data. Legendre & Troussellier (1988) as well as Legendre & Fortin (1989) have used the Smouse-Long-Sokal partial Mantel tests in a causal modeling framework. Sokal *et al.* (1987) have proposed to limit the permutations during the Mantel test in such a way as to evaluate which one among two rival alternative hypotheses (H_1) fits the data better; an example is provided in the section on limited permutations (Options of the program, section 8, below).

Input and output files

The calling program asks several questions about the input files; this reflects the multiplicity of options offered by this program. Read the questions carefully before answering them.

Simple Mantel tests require two matrices, referred to as **A** and **B**. Partial tests require the presence of a third matrix, **C**, besides matrices **A** and **B**. Finally, correlograms require file **B** as well as an ASCII file describing the distance classes.

(1) Input file B

Matrix **B** must always be present, and it always has to be a SIMIL-type binary matrix produced by SIMIL, IMPORT-EXPORT (Macintosh version) or IMPORT (CMS and VMS versions). It may be a distance or a similarity matrix.

(2) Input file A

Matrix **A** may take several forms, described below.

(2.1) Binary files of similarities

Matrix **A** may be a SIMIL-type binary matrix produced by SIMIL, IMPORT-EXPORT (Macintosh version) or IMPORT (CMS and VMS versions). Like matrix **B**, **A** may be a distance or a similarity matrix. It is better for matrices **A** and **B** to pertain to the same type, however; the interpretation of the sign of the Mantel statistic is easier when this is the case.

(2.2) File of distance classes

To compute a correlogram, a series of matrices **A** will be computed by the program from the information given in the file of distance classes, described below; these will be used in turn to compute Mantel tests for each distance class.

The file of distance classes, used to compute the Mantel correlogram, is called CLASSEF in the list of file names of the program, as well as in the EXEC or COM files of the mainframe versions. That file is in ASCII (readable) characters, not in binary. It contains an upper triangular matrix of distance classes among objects, without the diagonal. In the case of small data sets, that file may have been written by hand by the user, with the help of an ASCII editor. For larger problems, that file may have been prepared using program AUTOCORRELATION or AUTOCOR (see that program), or with the help of any other specifically designed user's program.

In that file, integers 1, 2, 3, etc. represent the various distance classes. A Mantel test is computed for each distance class present in the file; for each distance class in turn, the program builds a matrix **A** containing 1's for all pairs of objects pertaining to that distance class, and 0's for all the other pairs of objects. No Mantel test is performed for class "0" or smaller, if present in the file. The following file would represent an acceptable CLASSEF matrix for a set of 6 objects:

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

(2.3) Regular grid of objects

Matrix **A** may be computed by the program itself if the user declares that the points form a regular grid. The program reads the total number of points in the block of information of the file containing matrix **B**, and asks what is the width of the grid, from which the height of the regular grid can be computed.

(2.4) File of geographic coordinates (DMS, or decimal degrees)

The file of geographic coordinates used to compute matrix **A** is an ASCII file. The coordinates are written in free format. They may be presented on a Cartesian plane, or as terrestrial coordinates, either in degrees, minutes and seconds (DMS) or as decimal degrees. For instance, one may write 45 15 36 (in DMS), which is equivalent to 45.26 (in decimal degrees). The latitude is written first, followed by the longitude. The program offers the choice of computing the distance using either the Euclidean distance formula (flat coordinates), or the great circle distance following earth's curvature (coordinates on a sphere); if this is the case, the distances are expressed as nautical miles.

(3) Input file C

Matrix **C** used in the partial Mantel tests is always a SIMIL-type binary matrix produced by SIMIL, IMPORT-EXPORT (Macintosh version) or IMPORT (CMS and VMS versions), just like matrix **B**.

(4) Output file

On mainframes, the results of the Mantel tests are presented on the screen, instead of an output file. CMS users may have the results copied into a console memory file, using the procedure described on page 2. In the Macintosh version, the results are not printed on the screen; they are written on an output medium instead. The printer may be designated as the output medium; the results may also be

output medium instead. The printer may be designated as the output medium; the results may also be preserved in a file whose name is decided by the user. See the Results section for more information on how to interpret the results.

Limitations of the program

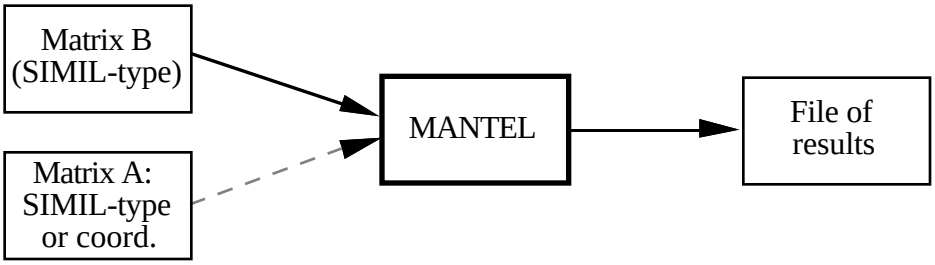
The mainframe versions are limited by two parameters found at the beginning of the program. These are the maximum number of objects that can be analyzed (ex. MAXNOBJ = 1000), and the largest number of objects for which permutation tests are authorized (ex. PETITNOBJ = 200). The Macintosh version contains no such limitations; the program dynamically allocates the available memory of the machine. Under FINDER, a message should be produced if the machines falls short of RAM to complete the computations. Notice that computing time increases approximately as the square of the number of objects.

Options of the program

The options available in the program allow to compare two matrices, to perform partial Mantel tests, or to compute a Mantel correlogram.

(1) Option 0: Mantel between two matrices

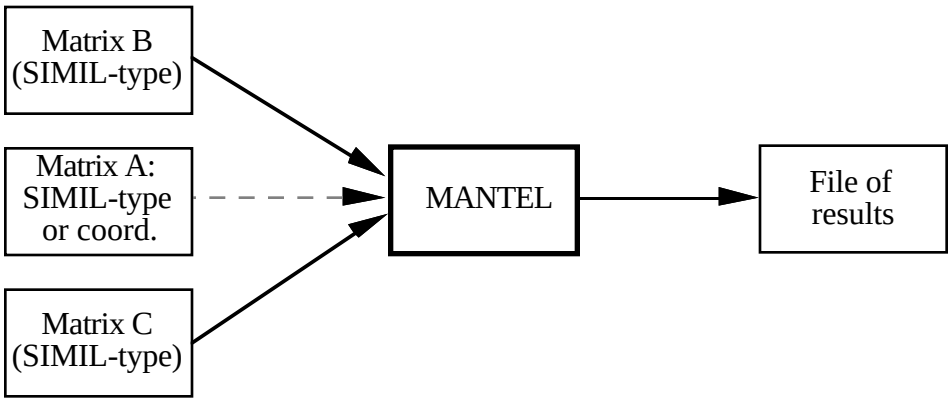
The program requests the name of the two matrices, **A** and **B**. It also asks to what type matrix **A** belongs.



With options (2.3) and (2.4) for matrix **A** (regular grid of points, or file of coordinates in degrees-minutes-seconds or in decimal degrees: see above), the user may ask to have the distances in matrix **A** transformed into 1/D or $1/D^2$.

(2) Options 1 to 3: Partial Mantel tests

The program proposes several types of partial Mantel tests. These methods all require three matrices to be presents (**A, B** and **C**). Here again, matrices **B** and **C** have to be SIMIL-type binary matrices, while matrix **A** may take one or another of the forms described above.



Option 1: Dow & Cheverud (1985) method. Statistic: **(A * (B - C))** where ***** represents the Mantel sum of products, and separates the two blocks to be permuted. This statistic may be expressed as follows: $\Sigma\ [a_{ij} * (b'_{ij} - c'_{ij})]$ (unstandardized statistic) or $\Sigma\ [a'_{ij} * (b'_{ij} - c'_{ij})']/(n-1)$ (standardized statistic), where the sign *prime* (') represents a standardized value.

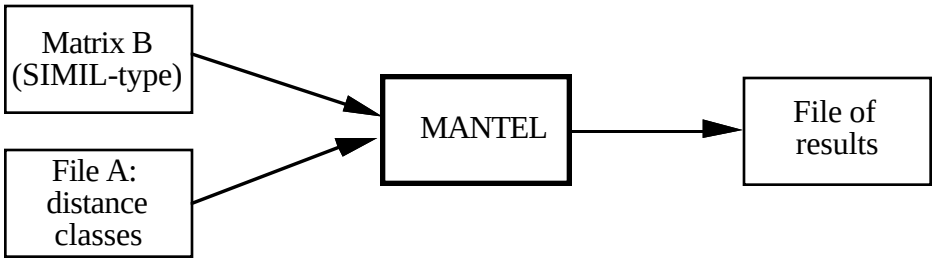
Option 2: Smouse, Long & Sokal (1986) method. Statistic: **(A*B.C)**. This statistic is actually the partial correlation between the values in **A** and **B** conditional on the values in **C**. It is computed by first calculating **A '** which is the matrix of residuals of the regression of the values in **A** against those in **C**, and then **B '** which is the matrix of residuals of the regression of the values in **B** against those in **C**, after standardizing the values within each of those matrices; a Mantel test is then performed between **A '** and **B '**. This is only another way of computing the partial correlation.

Option 3: Hubert (1985) method. Statistic: **(A * (BC))**. This statistic may be expressed as follows: $\Sigma\ [a_{ij} * (b_{ij} * c_{ij})]$ (unstandardized statistic) or $\Sigma\ [a'_{ij} * (b_{ij} * c_{ij})']/(n-1)$ (standardized statistic), where the sign *prime* (') represents a standardized value.

The partial test most currently used in our lab is the second one; the statistic takes the same value as the parametric partial correlation between the values in matrices **A** and **B**, conditional on the values in matrix **C**. At the present moment, option (2) appears to be the only acceptable one in spatial analysis (Oden & Sokal, submitted).

(3) Option 4: Mantel correlogram

This program is capable of computing a Mantel correlogram (Sokal, 1986; Oden & Sokal, 1986). Compared to the correlogram produced by AUTOCOR, this program presents the advantage of producing a correlogram from multivariate data, since the correlogram is computed from a similarity or distance file produced by SIMIL, which uses multivariate data in most cases; see Legendre & Fortin (1989) for an example. The Mantel correlogram is obtained by requesting computation's option "0", as well as option "0" for matrix **A**; a Mantel test is then computed for each distance class of the correlogram.



(4) Mantel statistics

(4) Mantel statistics

The program can compute either the Mantel statistic *Z*, which is simply the sum of cross-products of the corresponding values in the two matrices, diagonal excluded:

$Z = \sum \sum x_{ij} y_{ij}$ for all pairs of values (i, j) of the two matrices,

or a standardized form of that statistic, *r*, as proposed by Smouse, Long & Sokal (1986). To compute that statistic, the values within each of the distance matrices (diagonal excluded) are standardized, before computing the sum of cross-products; the result is divided by $(n - 1)$ where *n* is the number of pairs of distances considered in the computations. That statistic is then equivalent to the computation of a Pearson correlation coefficient between the values of the two matrices (diagonal excluded), so that the values so obtained are between -1 and +1. That the statistic be standardized or not does not change the associated probabilities.

(5) Probabilities

The probabilities may be computed in two different ways: either by permutations, or by transforming the *Z* or *r* statistic into another statistic, called *t* by Mantel (1967), which is asymptotically distributed like a standard normal deviate. That test gives a good approximation of the probability when the number of objects is sufficiently large, provided that some other conditions are also fulfilled (see Mielke, 1978). When the number of objects is large, the permutational test requires a lot of computing time. The user may then decide to ask the program to compute the approximate test only; this is accomplished by requesting zero permutation. For the permutation tests, a limit of 200 objects is set in the program **[CMS and VMS versions only ?]**. If the number of objects in the problem exceeds that value, the program computes only the approximate test in the case of option 0 (Mantel test for two matrices, or simple correlogram); it stops immediately with the partial Mantel tests (options 1 to 3). That limit is set by a parameter (PETITNOBJ = 200) in the block of constants at the beginning of the program, which may be modified by the user, depending on need, in the mainframe versions of the program.

Mantel test probabilities have often been presented as the proportion of values, under the probability distribution, which are located *left* of the observed value of the statistic; a negative significant Mantel statistic had a probability close to 0, while a positive significant value had a probability near 1. Our program provides instead the estimated probability (one-tailed test) of the null hypothesis (H_0: no linear relation between the two matrices), as it is customary in statistical testing. In this way, significant Mantel statistics have a probability near zero for both positive and negative values of the statistic.

(6) Permutation tests

For permutation tests, the user must indicate how many permutations are to be run. Asking for zero permutation means that the permutation test is not requested. For large matrices, the permutation test is useless because the approximate test statistic becomes asymptotically distributed like a standard normal deviate, so that it can be tested against the standard normal distribution.

Probabilities obtained by permutations are computed following Hope (1968). That method, which is also recommended by Edgington (1987), consists of including the observed value of the statistic among those of the reference distribution, so that it is never possible to obtain 0% of the values “as extreme as or more extreme than the observed value”. According to Edgington, this way of computing the probability is biased, but it has the merit of being valid. In any case, the probabilities so obtained have to be interpreted in terms of “strictly smaller” or “strictly larger” than the stated value; for instance, if the probability obtained by permutation is 0.05, then the probability of the null hypothesis to be true is strictly smaller than 0.05 for a one-tailed test. The precision of that probability value is the inverse of the number of permutations requested by the user.

(7) Hubert's standardization

The standardization proposed by Hubert (1985), which produces values between -1 and +1, consists of positioning the true value of *Z* or *r* between the extreme values (minimum and maximum) obtained from the permutations, and then attributing to that statistic the sign that *Z* or *r* had. A Hubert's standardized value equal to +1 essentially means that the observed value of the statistic is the largest one in the reference distribution, while a value equal to -1 means that the observed value is the smallest one in the reference distribution.

(8) Limited permutations

This program allows to run the test with permutations limited to exchanges among subgroups determined by the user (Sokal *et al.*, 1987). The program has to be told how many subgroups have to be recognized, and the list of objects which are members of each subgroup has to be given. Object numbers may be given one by one, or else in blocks using dashes (in the CMS or VMS versions only); for example: 1 4 7 9-32 38 67 would be a valid answer. To indicate that this option will not be used, the user answers "1" to the question "Number of groups to permute ? -- (Normal cases: answer "1")". The principle of that test is explained below.

Let us consider the case where the Mantel test is used as a test of goodness-of-fit of a model to data. The method consists in that case in formulating the alternative hypothesis (H_1) — for instance, the existence of distinguishable groups in the data — in the form of a model-matrix, containing values "1" (for instance) among the objects assumed to pertain to the same group, and values "0" elsewhere. A resemblance matrix is also computed for the data. The null hypothesis (H_0) of non-conformity of the model to the data is then tested by comparing the value of the Mantel statistic to a reference distribution obtained by repeatedly permuting one of the two matrices and recomputing the Mantel statistic.

When two different alternative hypotheses of group membership are both in agreement with the data (matrix **A**), one can proceed as follows to evaluate which one, if any, fits the data better:

1- Express each of the alternative hypotheses in the form of a model-matrix, called $\mathbf{B_1}$ and $\mathbf{B_2}$. The pairs of objects that belong to the same group according to alternative hypothesis 1 receive values "1" in matrix $\mathbf{B_1}$, while the pairs of objects that belong to the same group according to alternative hypothesis 2 receive values "1" in matrix $\mathbf{B_2}$.

2- A Mantel test is run between **A** corresponding to the data and model-matrix $\mathbf{B_1}$, permuting only within the groups recognized by the second alternative hypothesis; $\mathbf{B_2}$ actually becomes the null hypothesis of this test.

3- In the same way, a Mantel test is run between **A** corresponding to the data and model-matrix $\mathbf{B_2}$, permuting only within the groups recognized by the first alternative hypothesis; $\mathbf{B_1}$ actually becomes the null hypothesis of this test.

4- If only one test remains significant, the alternative hypothesis corresponding to it is retained.

The following example has been analyzed by Legendre & Lessard (in prep.). The question is whether gill nets of different mesh sizes catch essentially the same fish species at a series of sampling stations. The null hypothesis is that differences among nets are independent of the stations or net types. The first alternative hypothesis is that both net types catch fish from the same community at each station; if that hypothesis can be shown to be supported by the data, it becomes possible to put together the fishing results obtained from these two net types when studying fish communities. The second alternative hypothesis claims on the contrary that the first net type catches fish from a first community (small species) at all stations, while the second net type, with larger mesh size, catches

fish from a second community (larger species). These three hypotheses may be represented by the following data vectors; the numbers represent community types, assuming that there are 5 sampling stations:

	Station 1		Station 2		Station 3		Station 4		Station 5	
	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2	Net 1	Net 2
Observation no.	1	2	3	4	5	6	7	8	9	10
Null hypothesis	1	1	1	1	1	1	1	1	1	1
Alternative hypothesis 1	1	1	2	2	3	3	4	4	5	5
Alternative hypothesis 2	1	2	1	2	1	2	1	2	1	2

Each of these vectors of numbers may easily be transformed into a model-matrix by computing a Jaccard similarity coefficient for multistate nominal data, using coefficients S15 or S16 of the SIMIL program. In the first limited permutation test, the test is made between matrix **A** (similarity based upon the real data) and model-matrix $\mathbf{B_1}$, permuting only within the two net-type groups (1-3-5-7-9) and (2-4-6-8-10). Then, a second Mantel test is computed between **A** and model-matrix $\mathbf{B_2}$, permuting only within the five station-groups (1-2), (3-4), (5-6), (7-8) and (9-10).

Questions of the program

The first question of the program concerns the type of test requested. The choice is between the simple Mantel test and one of the partial tests. Mantel correlograms are most of the time computed from option 0 (simple Mantel test); the program also allows to compute partial correlograms, that is, correlograms consisting of partial Mantel tests. See how to specify the distance classes in input file **A**, in (2.2) above.

The second question concerns the options available for matrix **A**. These options are described in (2) above. If one tells the program that the points form a regular grid, the next question concerns the width of that grid (number of columns); since the total number of points is known, the program having read it in the block of information of file **B** (which is a SIMIL-type file), the program can automatically compute the number of rows of points. If on the contrary the user has chosen to prepare a file of geographic coordinates (option 3 or 4), the next question of the program allows to determine how to compute the distances: either from the Euclidean distance formula (flat coordinates), or following earth's curvature (great circle distance, for coordinates on a sphere); in this last case, the distances are expressed as nautical miles.

The next question concerns the Mantel statistic to be computed: either the original Mantel statistic *Z*, or the standardized statistic *r* which takes values between -1 and +1. This question does not appear when the Smouse, Long & Sokal test is requested, because the statistic in that case has to be a partial correlation.

The user must now determine how many permutations are to be run. If "zero" is requested, then only the approximate test is computed. Users of the Mantel test often request 999 permutations (for a total of 1000 with the real value); it is recommended, however, to substantially increase that number when coming close to the predetermined significance level α, because of the instability of the probabilities obtained from the permutational method (Jackson & Somers, 1988).

The last question concerns the number of groups to be permuted; see section (8) above. If there is more than one group, the program will ask how many objects pertain to each group, as well as the identification numbers of these objects; the information to be given is the sequential number of the objects in the input matrix, and not the identifiers that may have been given to the objects in the first 10

columns of the raw data matrix.

Example

The example below illustrates the use of the program to compute a partial Smouse, Long & Sokal (1986) relationship on mainframe (CMS or VMS systems; this example has been computed on CMS). The calling program first asks the user to identify the files that will be used; the answers are underscored. Then, after the header, come the questions asked by the program itself to determine the computation options. This example is one of the results reported by Legendre & Troussellier (1989): it is the partial Mantel test between variables MA and CHLA in the Thau lagoon, controlling for the effect of the matrix of geographic distances (XY).

```
Program MANTEL3, December 1989.

This program uses 2 or 3 dissimilarity matrices; two for Mantel tests
(they are then called A and B), and three for partial Mantel tests
(in which case they are called A, B and C). Similarity matrices can be
used instead of dissimilarity matrices, but mixing types would unduly
complicate the interpretation.

Matrix A : It can be a SIMIL dissimilarity matrix. If this is the
case, what is the name of the file that contains this matrix?
(defaults are "... data a")
MA  D01  B

If the points form a regular, rectangular grid, no file is required
to compute matrix A of geographic distances among points. If not,
and if you have not provided (above) a file for matrix A, the
program will need a file containing the coordinates of the objects.
What is the name of this file, if any? (defaults are "... data a")


B is a SIMIL dissimilarity matrix. What is the name of the file that
contains it? (defaults are "... data a")
CHLA  D01  B

When matrix C is needed, it is also a SIMIL dissimilarity matrix.
What is the name of the file that contains it, if any?
(defaults are "... data a")
XY  D01  B

To compute a Mantel correlogram, a file should be attached to the
run, that contains an upper triangular matrix of distance classes,
without diagonal. What is the name of this file, if any?
(defaults are "... data a")


P R O G R A M   M A N T E L   with permutation test

Author: A. Vaudor
Departement de Sciences biologiques, Universite de Montreal,
C.P. 6128, Succursale a, Montreal, Quebec H3C 3J7.

Computation requested:
```

```
Computation requested:
   (0) Mantel test for two matrices
   (1) Dow & Cheverud (A.(B-C))
   (2) Smouse, Long & Sokal (AB.C)
   (3) Hubert (A.(BC))
2

Options for matrix "A":
   (0) Input file in classes (for a Mantel correlogram)
   (1) Regular lattice (no file is required)
   (2) Distance (or similarity) file from simil
   (3) File of coordinates in degrees, minutes and seconds
   (4) File of coordinates in decimal degrees
2

Number of iterations ? -- (Recommended >= 250)
999

Number of groups to be permuted ? (General case: 1)
1

One-tail test on the left or on the right:
Probabilities near zero are significant.
ST stands for Smaller Than, EQ EQual & GT Greater Than the original statistic.
The original value is added to the EQuals, following Hope (1968).


Computation:                                    Permutations     Approximation
                r      r stand.  ST   EQ   GT     Prob(r)       t     Prob(t)
                     **Hubert**                 (Hope,1968)

  AB.C       0.25210   0.96420  997    1    2     0.00300   4.19588   0.00001

End of the program.
```

Contents of the file of results (Macintosh version)

The Macintosh version writes the results in a file, while the CMS and VMS versions present them on the screen instead, as we have seen in the example above. The file first recalls what resemblance matrices have been used during the computations, reproducing the block of binary information attached to each SIMIL-type file. In the case of a simple Mantel test, the method used is not identified on the left of the line of results.

The example that follows has been computed on Macintosh.The results tell us that the Mantel relation (r = 0.25210) is positive and significant at level α = 5% ($p_{1000\ \text{permutations}}$ = 0.003, $p_{\text{approximation}}$ = 0.00001). This is the simple Mantel test between matrices MA and XY, that are also part of the example above. Details about the permutation results are reported: 997 permutations have produced statistics with values smaller than (ST) the value obtained for the two original matrices; no value obtained by permutation was equal to the real value, since the number reported under “EQ” first contains the real value itself, following Hope. Finally, 2 results obtained by permutations were higher than (GT) the real value. The probability estimated from the permutation results is obtained by (EQ + GT)/(number of permutations + 1) = 3/1000 in this example. For a one-tailed test in the left tail, this probability would be obtained by (ST + EQ)/(number of permutations + 1). Notice that a problem of this size (63 observations) would not have required to proceed by permutations, the *t* test results being

probably sufficiently close to the permutational results for a 5% significance level. If we were interested in a 0.001 significance level instead, one would have to substantially increase the number of permutations, first to minimize the effect of Hope's correction, and then to verify on what side of the level the result actually falls.

```
Number of iterations:  999
One-tail test on the left or on the right:
Probabilities near zero are significant.
ST stands for Smaller Than, EQ EQual & GT Greater Than the original statistic.
The original value is added to the EQuals, following Hope (1968).

Computation:                                    Permutations     Approximation

               r      r stand.  ST   EQ   GT     Prob(r)       t     Prob(t)
                    --Hubert--                (Hope,1968)

            0.22338   1.00000  999    1    0     0.00100   4.69498   0.00000
```

In a correlogram, a new line of results is presented for each distance class. The following example has been computed on Macintosh.

```
**** Notice that in this correlogram, positive autocorrelation
     produces negative values of Z in the low distance classes.

Number of iterations:  249
One-tail test on the left or on the right:
Probabilities near zero are significant.
ST stands for Smaller Than, EQ EQual & GT Greater Than the original statistic.
The original value is added to the EQuals, following Hope (1968).

Computation:                                    Permutations     Approximation

               r      r stand.  ST   EQ   GT     Prob(r)       t     Prob(t)
                    --Hubert--                (Hope,1968)

class    1
           -0.19512  -0.50163   30    1  219     0.12400  -1.26150   0.10356
class    2
           -0.23068  -0.58478   13    1  236     0.05600  -1.60066   0.05473
class    3
           -0.22218  -0.64604   17    1  232     0.07200  -1.60212   0.05457
class    4
            0.07324   0.17663  178    1   71     0.28800   0.48162   0.31504
class    5
            0.12409   0.28565  203    1   46     0.18800   0.87896   0.18971
class    6
            0.12082   0.32244  191    1   58     0.23600   0.73562   0.23098
class   7
            0.24780   0.59178  230    1   19     0.08000   1.50877   0.06568
class   8
            0.38124   1.00000  245    5    0     0.02000   2.25543   0.01205
```

Notice that in this type of correlogram, if matrix **B** is a distance matrix, the Mantel statistic has a negative sign in cases of positive autocorrelation; the reverse is true if matrix **B** is a similarity matrix, a positive sign indicating the presence of positive autocorrelation. This is the meaning of the note printed

positive sign indicating the presence of positive autocorrelation. This is the meaning of the note printed before the correlogram. To draw the correlogram, the values of r (ordinate) are plotted against the distance classes (abscissa). The significance values may be taken either from the column of the permutational or from that of the approximate tests; a Bonferroni correction should be used to evaluate the overall significance of the correlogram, as recommended also in programs SPATIAL AUTOCORRELATION and PERIODOGRAPH. In the example above, where matrix **B** is a distance matrix, all signs of the Mantel statistics should be changed before plotting the correlogram.

PCOORD

What does PCOORD do ?

Program PCOORD produces a reduced-space ordination following the method of principal coordinates analysis (Gower, 1966). Like principal components analysis, this is a *metric multidimensional scaling* method. The computations, however, are made on a *similarity* or *distance* matrix instead of a raw data table; this is also the case with the methods of *nonmetric multidimensional scaling.*

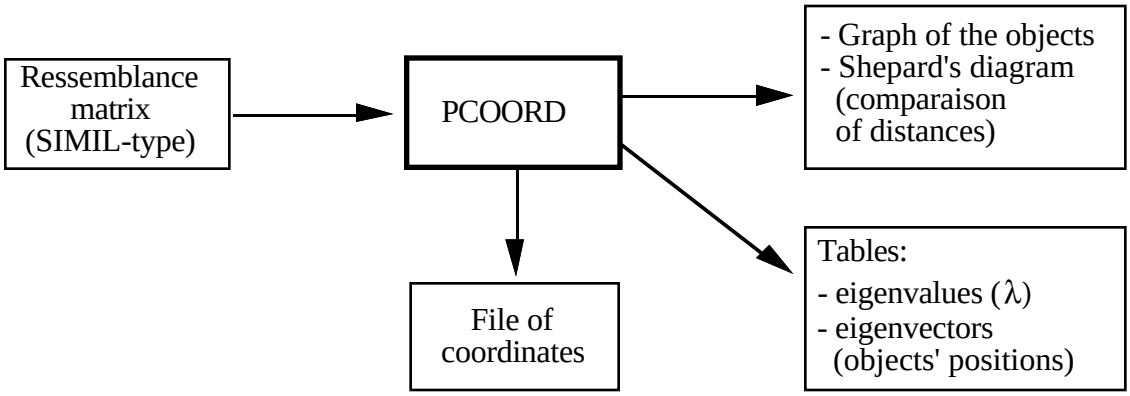
Each distance d is first transformed into a new distance $d' = -d^2/2$ before centering the matrix using the formula

$$alpha = d' - d'bar_i - d'bar_j + d'bar$$

where $d'bar_i$ and $d'bar_j$ are respectively the mean of row i and of column j in distance matrix d', while $d'bar$ is the mean of all the values in the matrix. The new coordinates of the objects in reduced space are the eigenvectors of that centered matrix, after normalization to the square root of their eigenvalues.

The size of the distance matrices that can be analyzed by this program is limited, in the CMS and VMS versions, by a parameter DIMENSION at the beginning of the program. If that constant is too small, one should change it and recompile the program. There are in principle no limits to the size of the matrices that can be analyzed by the Macintosh version; the program occupies all the available RAM space in the computer, so that the size of the matrices that can be analyzed is, in practice, a function not only of the size of the memory available in the machine, but also of the simultaneous use of MultiFinder, of a RAM cache, or of other programs. Under system 6.04, matrices of size **???** can be analyzed with 1 Meg of RAM memory.

Input and output files



(1) Input file

The input file is a binary SIMIL-type ($p \times p$) similarity or distance matrix produced by programs SIMIL, IMPORT (CMS and VMS versions) or IMPORT-EXPORT (Macintosh version) describing the resemblance among p objects, computed from n variables.

In principle, the matrix used in this analysis should correspond to a *metric* distance, allowing a Euclidean representation of the objects. If this is the case, p objects will produce a maximum of $(p - 1)$ positive eigenvalues and a single null eigenvalue, since $(p - 1)$ dimensions are sufficient to represent

positive eigenvalues and a single null eigenvalue, since (p - 1) dimensions are sufficient to represent the position of p objects relative to one another in Euclidean space. Negative eigenvalues are produced when the distances among objects cannot entirely be represented in the Euclidean way. Gower (1982) has shown that in some cases, a metric distance function may lead to a non-Euclidean representation of the objects, while Gower & Legendre (1986) have described the conditions allowing a fully Euclidean representation of a resemblance matrix. In any case, the non-Euclidean fraction of the reduced-space ordination does not matter much as long as it is clearly smaller, in absolute value, than the variability expressed by the first principal coordinates.

(2) File of results

The file of results contains the eigenvalues as well as the positions of the objects with respect to these first three eigenvectors, if they correspond to positive eigenvalues. When there are negative eigenvalues, the percentage of variance corresponding to each eigenvalue λ_i is corrected by the absolute value of the largest negative eigenvalue (Cailliez & Pagès, 1976), using formula

$$\lambda_i' = (\lambda_i + |\lambda_p|) / [\sum_{j=1}^{p-1} \lambda_j + (p - 1) |\lambda_p|]$$

where $|\lambda_p|$ is the absolute value of the largest negative eigenvalue and p is the number of objects; p also represents the number of eigenvalues that have been computed.

Three graphs of the objects are printed next: a first graph for the projection on axes I and II, a second one for axes I and III, and a third one for axes II and III. In the CMS and VMS versions, these graphs, which are printed laterally, present the first axis vertically and the second one pointing left. This representation is justified by the fact that axis I is always more variable than axis II, so that it is likely to be longer. Rotate these graphs by 90° if you reproduce them.

(3) File of coordinates

A second file, written in ASCII and called "COORD data a" by default in the CMS version, contains the positions of the objects with respect to as many of the principal axes as requested by the user; that number cannot be larger than the number of positive eigenvalues, however. The user who wishes to produce graphs of more than the first three eigenvectors may transfer that file to a microcomputer where the graphs can be produced using any statistical program. That file may also be used as input to the non-hierarchical clustering method *k-means*, as explained in the chapter dealing with program K-MEANS.

Files of type (2) and (3) described above may also be produced by the Macintosh version. The graphs are of publication quality, as can be seen in the examples below; the user may plot any combination of axes. The graphs are first presented on the screen; they can then be printed, or saved in a PICT file for future use. The Macintosh version may also produce a Shepard diagram (dispersion diagram comparing the original distances to the distances in the reduced space: see example).

Questions of the program

The questions presented by the program on the Macintosh screen are described in the following paragraphs. The questions of the CMS and VMS versions are essentially the same, as can be seen in the example below, except with respect to the Shepard diagram which is not available in the mainframe versions. To start the program on the Macintosh, click on the PCOORD icon and then on “Open” in the “File” menu.

(1) “Title ...” — The user gives a title, which will be used as header in the graphs sent to the printer.

(2) “Is this a matrix of distances instead of similarities?” [Yes, No] — The answer is *Yes* if the input file contains a distance matrix.

(3) “Input file” — A menu is presented showing the available SIMIL-type binary files, since the input matrix to PCOORD must have been produced by SIMIL, IMPORT-EXPORT (Macintosh version) or IMPORT (CMS and VMS versions).

(4) “How many eigenvalues should be computed?” — The algorithm used in the Macintosh version to compute the eigenvalues is a stepwise **[name?]** algorithm, which computes the largest eigenvalues first. The user may limit the computations to the first few eigenvalues (usually 2 to 5) which usually contain most of the variance; this may represent an appreciable saving in time with large problems (many objects).

(5) “How many dimensions are to be drawn?” — Successive graphs will be produced for all pairs of principal axes requested by the user. For instance, if one requires 3 dimensions, three graphs will be produced, respectively corresponding to axes I and II, I and III, II and III. To increase the resolution of the picture, simply use the mouse to draw a box around any part of the picture you wish to blow up.

(6) “Write object numbers on graph?” [Yes, No] — The answer is *Yes* if the user wants the object sequential numbers to be printed as identifiers on the graphs.

The list of eigenvalues is now available in the “Computation details” menu. One may go up or down the list by pointing the mouse cursor at the top or the bottom of the table. The results may be sent directly to the printer, or copied onto a file of results for future reference. In the same way, from the “Graphs” menu, graphs may be sent to the printer, or they may be preserved in PICT files, which allows to edit them using a graphics program or to include them in a manuscript using a word processor. It is necessary to “Finish” a graph before going to the next one, or to the next question.

(7) “How many coordinates will be written?” — The user indicates how many coordinates (integer number) are to be written onto an output file. A file name will be requested by the program. The answer is “0” (zero) if that file is not needed.

(8) “Distance comparison?” [Yes, No] — This question is not presented by the CMS and VMS versions of the program. If the answer is *Yes*, the next questions allow to determine how the comparison will be made (Shepard diagram) between the distances in the input matrix and those in the space reduced to 2, 3, … dimensions. In that graph, a narrow cloud of points, located under the diagonal but close to it, indicates a good representation of the original distances in the reduced space. If a distance has been used that cannot entirely be represented in Euclidean space, points may appear above the diagonal of the graph.

(8.1) “How many eigenvectors are to be compared?” — One indicates how many dimensions of the reduced space will be included in that comparison of distances (generally 2 or 3).

(8.2) “XX distances to compute; would you rather sample them?” [Yes, No] — There are XX = $p(p-1)/2$ distances among p objects. When that number becomes too large (more than a few hundreds, which would cause the calculations to be too long), the user may ask the computer to randomly choose a given number of these distances. How many will be chosen is determined by question (8.3), the selection being done by a pseudo-random number generator initialized at question (8.4).

(8.3) “Number of distances to sample?” — The user writes how many distances should be selected.

(8.4) “Random number generator: type a (small) integer” — A small positive integer is given, for instance 2, 5 or 10.

instance 2, 5 or 10.
(8.5) “Another comparison of distances?” [Yes, No] — Answering *Yes* to this question brings you back to question (8.1). Answering *No* terminates the execution of the program.

Example

The example that follows uses a file of Mahalanobis distances, previously computed among groups of observations. That SIMIL-type file is called **mahal d5 a**; it will be used as input by the principal coordinate analysis program. The example has been run under CMS. The CMS or VMS calling file asks the first three questions, which are followed by the questions of the program itself.

Notice the question about the width of the graph (flagged 1 in the left margin). If for instance the user wants the graph to be 8 inches wide (20 cm), the answer to the question is “8”; the algorithm asks for answers that are multiples of 4. Since the input file contains a distance matrix, the answer is **d** to question (2). Finally (3), the user has asked in this example that the position of the objects with respect to the first **5** principal coordinates be written in a file.

```
     Pcoord
     What is the name of the SIMIL input file?
     (defaults are "... data a")
     mahal  d5  a

     What is the name of the output file (Eigenvalues and Graphs)?
     (defaults are "... listing a")
     mahal  out  a

     What is the name of the COORD output file, if any?
     (defaults are "COORD data a")

     Execution begins...

(1)   WIDTH OF THE GRAPH? (in inches: multiples of 4)
     8
      IS THE INPUT FILE A SIMILARITY OR A DISTANCE FILE? (Write S or D)
(2)  d
      RE-WRITING THE COORDINATES OF THE OBJECTS (File "COORD"):
      HOW MANY AXES DO YOU WANT? (Type 0 if you don't want any)
(3)  5
      TITLE OF THE RUN ?
     Mahalanobis  distances,  9  groups

      End of the program.
```

Graphics and contents of the file of results

The first file contains the eigenvalues, as well as the percentage of variance explained by each one. Since there are negative eigenvalues, the correction described in the section about the file of results has been used here.

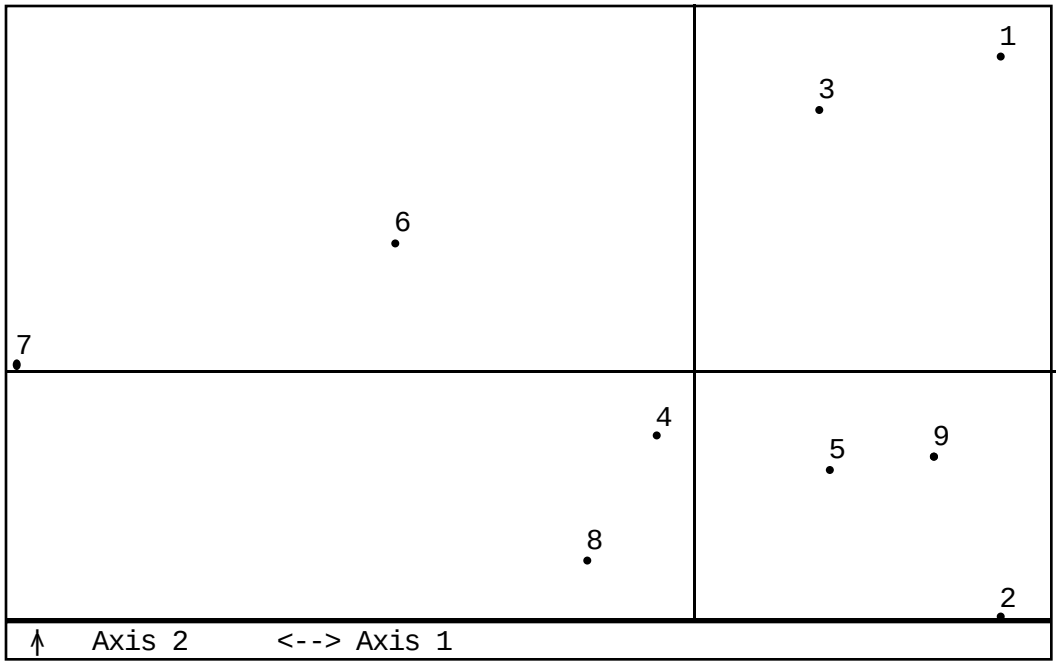
```
        Eigenvalues       % of variance
            9.43558            50.40291
            3.55587            18.99487
            3.06849            16.39137
```

```
 1.62149            8.66186
 0.70898            3.78740
 0.30302            1.61886
 0.02664            0.14252
-0.00000            0.00022
-0.00004            0.00000
```

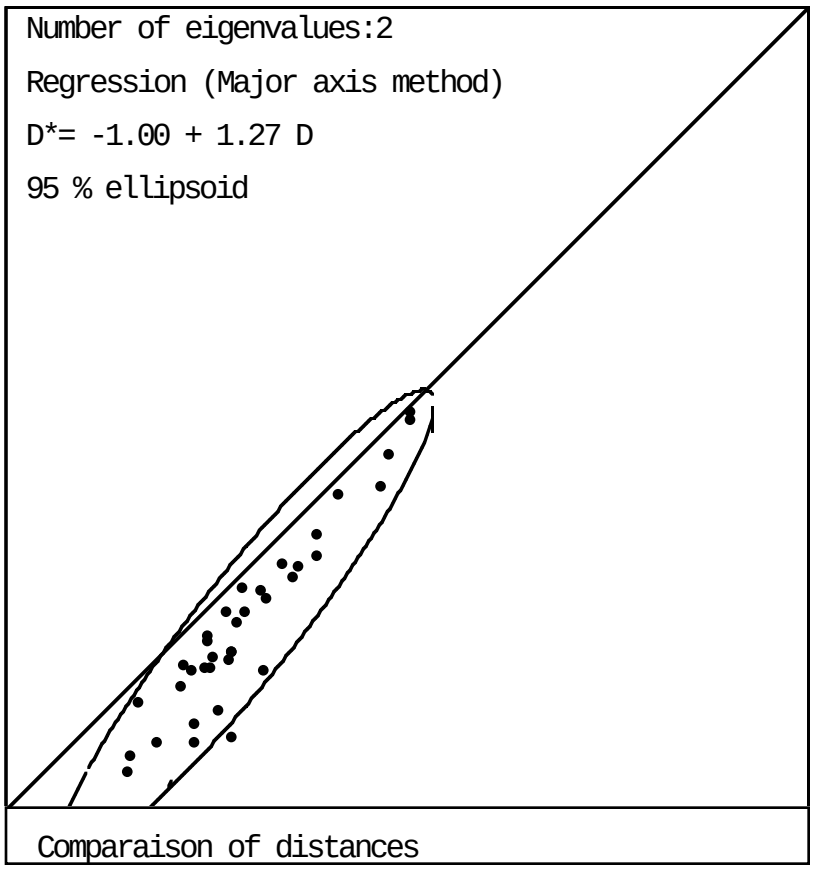
Another file contains as many principal coordinates (columns) as were requested by the user; here, 5 coordinates. Each line of that file then represents the coordinates of an object with respect to 5 dimensions.

```
 1.03108   1.06888   0.80571   0.12374  -0.27798
 1.03064  -0.85473   0.17086   0.84875   0.15406
 0.42006   0.88857  -0.96345   0.28882   0.08743
-0.12718  -0.23230  -0.08304  -0.38978  -0.45709
 0.45717  -0.34516   0.69916  -0.44539   0.31628
-0.99884   0.43152  -0.34929  -0.14860   0.41268
-2.26578   0.01133   0.59580   0.27563  -0.03772
-0.35435  -0.66460  -0.66777   0.04047  -0.32255
 0.80720  -0.30352  -0.20799  -0.59363   0.12490
```

The graph of the position of the objects in the first two dimensions is presented hereafter (Macintosh version).



The Shepard diagram reproduced below compares the original distances (abscissa) to the distances in the space formed by the first two principal coordinates (ordinate). It displays a narrow cloud of points, near the diagonal; this indicates that the original distances are well represented by two principal coordinate dimensions only.



*PERIODOGRAPH*Macintosh *or PERIOD*CMS/VMS

What does PERIODOGRAPH do ?

Program PERIODOGRAPH computes and plots a contingency periodogram (Legendre *et al.*, 1981) for a univariate space or time series. The data may be qualitative (nominal), semi-quantitative (ordinal), or quantitative. Quantitative and semi-quantitative data must first be divided into classes before computing this periodogram; the program takes care of this division, following an optimality criterion. In the periodogram itself, the contingency statistic is computed for all periods in the observation window, that is, periods $T = 2$ to $T = n/2$ where n is the length of the data series; in the Macintosh version, the user may choose a narrower computation window. Legendre & Legendre (in 1984a, vol. 2, and more briefly in their 1983 book), as well as the above-mentioned paper, provide more details on the method. Besides its capacity to analyze semi-quantitative or qualitative data series, the method also allows to analyze short series, which is not the case with the Schuster periodogram or spectral analysis, for example. The method also allows to analyze multivariate time series, by first computing a multivariate partitioning of the data series (through clustering), followed by periodogram analysis of the resulting data partition, as proposed in the 1981 paper; however, one should prefer to compute a Mantel correlogram (see program MANTEL) instead of a contingency periodogram in this case. Another advantage of the Mantel correlogram method is that it is not restricted to data with a constant sampling interval.

Dividing a quantitative or semi-quantitative variable into classes is accomplished by a procedure which optimizes the following two criteria, in such a way as to take tied values (*ex aequo*) of the data series into account:

1- For a given number of classes, one minimizes the sum of within-class sums of squares; this part of the computations is done either on the raw data, or on ranks.
2- The program selects the number of classes that maximizes the amount of entropy per class.

A stepwise algorithm, transcribed into procedure APPROX of the program, is described in the Legendre *et al.* (1981: 969-973) paper. That procedure first looks for the two-class partition which minimizes the first criterion; then, keeping the first division fixed, one looks for a second cutting point which would create three classes minimizing again the minimum variance criterion; and so on until the second criterion is maximized. A second algorithm has recently been created by A. Vaudor. This method, translated into procedure EXACT of the program, finds at each step the optimal partition of the observations into k classes, independently of the class limits found during the previous step; the partition that maximizes the amount of information per class is retained. The program uses procedure EXACT whenever possible. One should notice that with this algorithm, the second criterion is often optimized for three classes. The user can always impose another number of classes if she so wishes.

Input and output files

(1) Input file

The input file is an ASCII file (readable using an ASCII editor or a word processor) which may contain data either in the form of classes (categories) or as integer or real values. The Macintosh version imposes the following limits: no more than 2 000 real values, or 10 000 integer values, or else 60 classes. In versions CMS and VMS, the user decides, before compiling the program, what the

values are for the program parameters that fix these limits; parameter LIMITE fixes the maximum number of values that can be analyzed in a data series, while parameter LIMCLASSES fixes the maximum number of classes in each qualitative data series.

The observations are entered following either their temporal order, or the transect in the case of spatial data, without identifier of any kind, one series after the other. There are three points to check concerning the input file:

1- All data must be strictly positive. This restriction comes from the fact that the method has been developed first for nominal data, coded as *k* classes which are usually numbered from 1 to *k*. If one wishes to analyze quantitative data containing values that are null or negative, they have to be transformed before entering the PERIODOGRAPH; data can easily be made strictly positive using program VERNORM in this package, or else one of the many statistical packages available on microcomputers.

2- To simultaneously analyze several data series, each one must be written as a line of the input data file, or on two or more consecutive lines. All series analyzed in a single run must be of the same length. If necessary, it is easy to transpose a data file using program VERNORM in this package.

3- As it is also the case with other methods of time series analysis, this program assumes that the data are stationary (i.e., same mean and same variance for the various portions of the series), and that the sampling interval (lag) is constant. If this is not the case, the sampling interval can be made constant by interpolation. This program cannot handle missing values; these must be filled either by interpolation or using some other form of estimation.

The following file, which contains two series of 16 observations, would be an acceptable file as input to the PERIODOGRAPH:

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

(2) Output file

The output file first contains the information about the division of quantitative or semi-quantitative variables into classes, and then the details of the periodogram. See the example below. This output comes out on the screen only in the CMS and VMS versions; it is possible to have it written to a “screen memory” file using the procedure described on page 2.

Besides that file, the Macintosh version also produces graphs of the periodogram on the screen; examples are displayed below. This graphical option is not available in the CMS and VMS versions.

Questions of the program

The questions of the program in the CMS and VMS versions are presented in the next section (Example). These questions are essentially the same as those flashing on the Macintosh screen, although their precise formulation may slightly differ in a few cases. To start the program on the Macintosh, click on the icon, then choose “Open” in the “File” menu.

(1) “File of results” — A menu is presented allowing the user to give a name to the file of results. A name is suggested by default.

(2) “Input file” — The program presents the available ASCII file names in a menu.

(3) “Number of observations” — Type the number of observations in each of the data series.

(4) “Number of variables” — Type the number of data series to be analyzed.

(5) “Is the data file already in classes (nominal data)? [Yes, No] — The answer is *Yes* if the data are qualitative (nominal). If the answer is *No*, the following questions are presented on the screen:

(5.1) “Number of classes? (0 for computation by the program)” — The user may decide how many classes he wants to obtain; in this case, only the first criterion of the partitioning algorithm will be used (minimizing the sum of within-class sums of squares), as explained in the introduction section above. If the answer is “0”, the program determines what is the number of classes that optimizes the second criterion of the algorithm (maximizing the amount of entropy per class).

(5.2) “Computation on ranks rather than original raw data?” [Yes, No] — If the answer is *Yes*, the computations are carried out on ranks rather than raw data values. Semi-quantitative data are not modified by this procedure, except for tied values which are treated as in non-parametric statistics.

(6) “Output file. Confidence interval - Significance level:” The answer is given by clicking on one of the four answers [•0.005 •0.01 •0.05 •0.10] — The significance level set here is used to compute the critical value of the periodogram statistic. In the file of results, the critical value appears as a numerical value, and also as a “+” in the graph.

The first computation steps (reading the data in, division into classes) are carried out at this point.

(7) “Interval of analysis: from $\mathbf{x_1}$ to $\mathbf{x_2}$” [OK] — The periods included in the observation window of a periodogram are from $T = 2$ to $T = n/2$ where n is the total number of observations in the series; the valued displayed for $\mathbf{x_1}$ is then 2 while the value for $\mathbf{x_2}$ is $n/2$. If the series is long, the user may wish to restrict the computations to a narrower window; the values of $\mathbf{x_1}$ and $\mathbf{x_2}$ can be changed before clicking on the *OK* button. This question is presented only by the Macintosh version of the program. It determines the number of classes that will be displayed in the periodogram and, eventually, included in the computation of the Bonferroni correction.

The periodogram is computed and displayed on the screen. The significant periods are evidenced by a scale of grays, corresponding to the following probability levels:

Significance level:	p≤0.001	p≤0.01	p≤0.05	p≤0.10	p>0.10
Symbol:	****	***	**	*	

Menu “Pictures” allows to print the graph or to save it as a PICT file, which makes it possible to edit it using a graphics program or to include it in a page of text using a word processor. A “Bonferroni correction” may be requested from the same menu in order to correct for the effect of multiple testing on the significance level of the test. That correction consists of using a more restrictive significance level $\alpha' = \alpha$ / (number of simultaneous tests); see Cooper (1968) or Miller (1977). For example, if 7 tests are computed simultaneously (for 7 periods), the Bonferroni correction modifies the significance level α to $\alpha' = \alpha / 7$, which may change the significance of some of the periods of the periodogram (see example). For the same reason, and following Oden (1984), it is recommended to use a Bonferroni correction in correlograms (programs SPATIAL AUTOCORRELATION and MANTEL).

Clicking on “Finish” in the “Pictures” menu allows to go back to the “File” menu, if the user wishes to analyze another data file immediately. Command “Interrupt” in menu “R: Period” allows to quit the program.

Example

The following example illustrates the use of the program on mainframes. The data series contains 16 semi-quantitative values:

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

The calling file, whose dialog makes up the first part of the example, asks for the name of the input data file; this example has been run under CMS.

```
*** Have you checked that all data values are strictly positive?
*** ... that the variables are the LINES of the data file?
*** ... that you have a constant time lag?
*** ... that there are no missing values?

What is the name of the DATA file? (Defaults are "... data a")
semiq  16  a

Execution begins...

 C O N T I N G E N C Y   P E R I O D O G R A M

 VERSION 2.0b

 UNIVERSITE DE MONTREAL
 DEPARTEMENT DE SCIENCES BIOLOGIQUES
 CASE POSTALE 6128, SUCC. "A"
 MONTREAL, P.Q. H3C 3J7

 AUTHOR:  A. VAUDOR

 NUMBER OF OBSERVATIONS
16
 NUMBER OF VARIABLES
1
 ARE THE DATA ALREADY DIVIDED INTO CLASSES ? (Y or N)
n

 Note that this program  uses procedure EXACT,  whenever
 possible, to partition a variable into classes.  It may
 produce a better partitioning  than the stepwise proce-
 dure  described in  Legendre et al. ( 1981:  969-973 ),
 called APPROX in the source program.

 NUMBER OF CLASSES ? (0 IF THE PROGRAM IS TO DECIDE)
0
 DO YOU WANT THE VALUES TRANSFORMED INTO RANKS ?
o
 CHOOSE CONFIDENCE INTERVAL :
 TYPE  1 for  0.005 , 2 for  0.01, 3 for  0.05, 4 for  0.10
3

 CONTINGENCY TABLE:
```

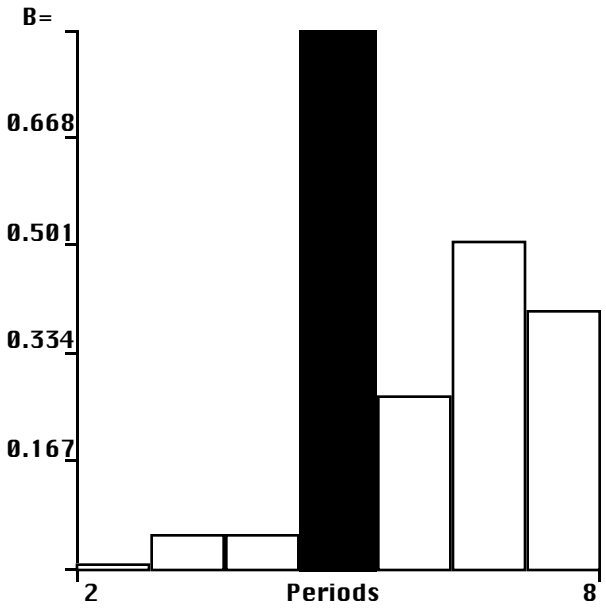
```
 NUMBER OF CLASSES:    3
        CLASS      UPPER LIMIT
           1         3.00000
           2         6.00000
           3        10.00000

 H(S)/S :    0.52043    IN BASE 2 LOGS

             C O N T I N G E N C Y   P E R I O D O G R A M

     (+=CONFIDENCE INTERVAL, *==>(+=B) )       THE SCALE OF B IS IN NATURAL LOG

 T=_ 0      0.27      0.54      0.81      1.08            CRITICAL
    .         .         .         .         .      B       VALUE    PROB(2NB)
   2B      +  .         .         .         .    0.00000   0.18719   1.00000
   3.  B      .+        .         .         .    0.07630   0.29656   0.65514
   4.    B    .    +    .         .         .    0.12912   0.39375   0.65885
   5.         .       + .         B         .    0.82227   0.48437   0.00093 ****
   6.         B         .+        .         .    0.25824   0.57187   0.60311
   7.         .         . B +     .         .    0.58357   0.65625   0.09670 *
   8.         .   B     .      +  .         .    0.38905   0.74062   0.57025
End of the program.
```

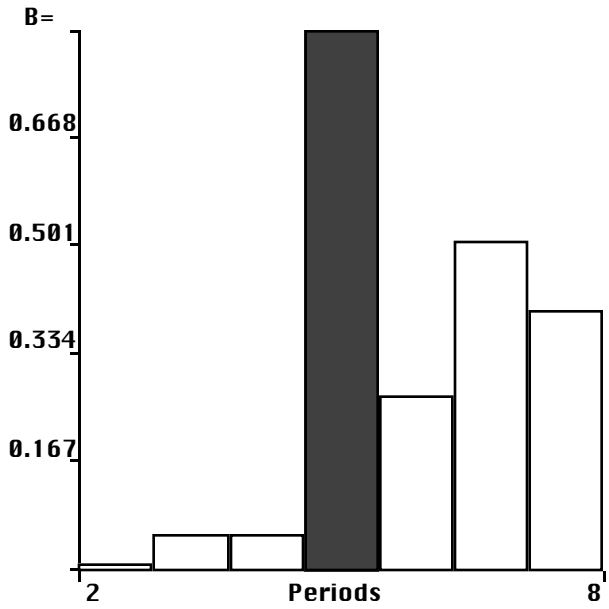
Graphics and contents of the file of results

The two graphs reproduced below are contingency periodograms as they appear on the screen of the Macintosh. The 16-value data series subjected to analysis (qualitative data) is the following:

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

This example has also been analyzed in the Legendre *et al.* (1981) paper. The left-hand graph is before the Bonferroni correction, the right-hand graph is after the correction:





Without correction for multiple testing *After a Bonferroni correction*

When applying a Bonferroni correction, the significance level changes. In the present case, there are 16 values, so that the program analyzes periods 2 to 8, or 7 periods; since 7 tests are carried out simultaneously, the Bonferroni correction modifies the significance level α to $\alpha' = \alpha / 7$, which changes the significance of the probability value of period 5:

Significance level α before correction	After Bonferroni correction: $\alpha' = \alpha / 7$
0.10 *	0.01429 *
0.05 **	0.00714 **
0.01 ***	0.00143 ***
0.001 ****	0.00014 ****

Period	Prob.(H_0)	Significance before correct.	Significance after correct.
2	0.81762		
3	0.77290		
4	0.94024		
5	0.00079	****	***
6	0.56404		
7	0.17769		
8	0.53819		

The Macintosh output file first contains information about the division of quantitative or semi-quantitative variables into classes, followed by details on the contingency periodogram itself. The listing below results from the analysis of the following file of 16 semi-quantitative values (same file as in section “Example” above):

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

This example, which has also been analyzed in the Legendre *et al.* (1981) paper, is not the same as the one which led to the two graphs above.

```
PERIOD: Contingency periodogram
        (Version 3.0)

Author: A. Vaudor
Département de sciences biologiques, Université de Montreal,
C. P. 6128, succursale A, Montréal, Québec H3C 3J7.

DATA FILE :16 quant.

Contingency table


Number of classes:   3
Class    Limit
   1    3.00000
   2    6.00000
   3   10.00000

h(s)/s :    0.52043
```

This first part is only presented when the program has been asked to divide an ordered variable (quantitative or semi-quantitative) into classes. The **upper** bound of each class is given by the program, as well as the amount of entropy per class for that division [“h(s)/s”]. See the remark in the introduction section about algorithm EXACT used by the program, compared to the stepwise algorithm described by Legendre *et al.* (1981). Next, the output file contains the contingency

periodogram itself:

```
      Contingency periodogram

 (+= Confidence interval, *==>(+=b) )    Scale  in  nat. log.

  T=\ 0      0.27      0.54      0.81      1.08 Value
    .         .         .         .         .   B         critical  prob(2nb)
   2B    +    .         .         .         .   0.00000   0.14406   1.00000
   3.  B     +.         .         .         .   0.07630   0.24313   0.65514
   4.    B    . +       .         .         .   0.12912   0.33125   0.65885
   5.         .    +    .         B         .   0.82227   0.41875   0.00093 ****
   6.         B       + .         .         .   0.25824   0.50000   0.60311
   7.         .         .+B       .         .   0.58357   0.57812   0.09670 *
   8.         .   B     .   +     .         .   0.38905   0.65938   0.57025
```

This graph represents a periodogram whose abscissa (periods *T*) goes from top to bottom while the ordinate (common entropy *B*, computed in natural logarithms) goes from left to right. Symbol “B” is used in the graph to represent the value of statistic *B*. The critical value, for the probability given as answer to question (6) (without Bonferroni correction), is represented by “+” signs; the probability requested for the confidence interval is here 0.1. The three columns of numbers give the precise value of statistic *B*, the critical value at the predetermined confidence level, and the probability of the null hypothesis to be true (probability that the computed value of *B* is not different from zero). Finally, a last column highlights the values that are significant at levels 0.10 (*), 0.05 (**), 0.01 (***) or 0.001 (****), before a Bonferroni correction is applied. For long data series, one should not be surprised to find that the multiples of the basic significant periods are also significant.

PNCOMP Macintosh

What does PNCOMP do ?

Program PNCOMP produces a reduced-space ordination using the method of principal component analysis (PCA) described in all textbooks of multivariate statistics. That very general method of analysis includes several variants; the main ones are briefly discussed in Table 3.

Table 3 — Questions related to the results of principal component analyses (adapted from table 9.I of Legendre & Legendre, 1984a).

Before a principal component analysis:

1) Are the descriptors appropriate?

   → Quantitative descriptors; normality; a limited number of zeros; in principle, more objects than descriptors

2) Are the descriptors dimensionally homogeneous? Question 7

   → If so: PCA on the dispersion (variance-covariance) matrix
   → If not: PCA on the correlation matrix

3) Purpose of the reduced-space ordination: Question 8

   → To represent the relative position of the objects: normalize the eigenvalues to 1
   → To represent the correlations among descriptors: normalize the eigenvectors to $\sqrt{\lambda}$
   → To represent both the object and the descriptors: biplot (normalize to 1)

Examining the results of a principal component analysis:

1) Which eigenvalues are significant? See description of the file of results

   → Test: is λ_i larger than the mean of the λ's?
   → Test: is the % of variance of λ_i larger than expected under the broken stick model?

2) Which descriptors contribute the most to the formation of the reduced space?

   → See graph of the descriptors, in which the variables are represented by axes (arrows), or else the table of coordinates of the descriptors Question 14
   → Look at the descriptors which are longer than the circle of equilibrium contributions
   → Examine also the correlations between descriptors and principal axes

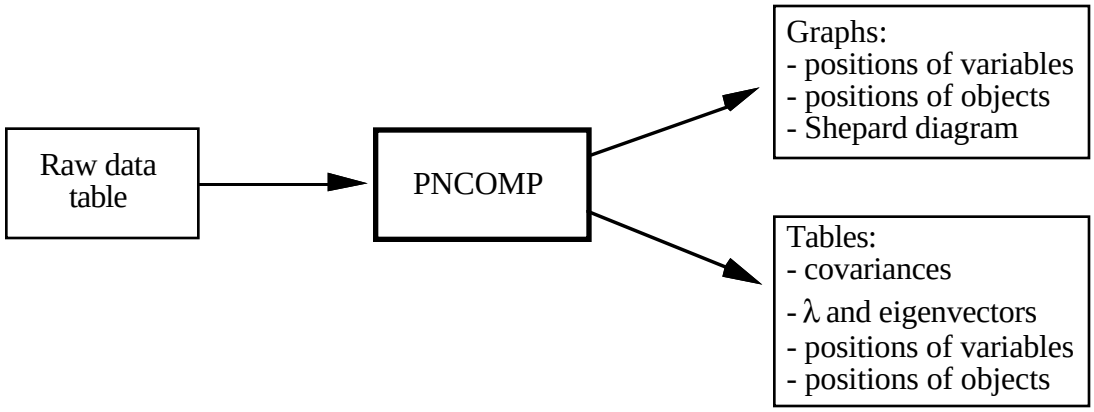
3) How to find the position of the objects in the reduced space?

   → Examine the graph and table of the object coordinates in the reduced space Question 15

4) Are the distances among objects well preserved in the reduced space?

   → Examine the Shepard diagram (comparison of distances) Question 16

We have written only a Macintosh version of this program, since many other packages offer this type of analysis on mainframes. The computations are made from a raw data table, which may contain missing values. The program produces graphs as well as a file of results, if the user request them. For the time being, the program can handle no more than 55 variables **[modify?]**.



Input and output files

(1) Raw data file

The raw input data file is a rectangular array (rows = objects, columns = descriptors) of quantitative data, written in ASCII, without any row or column identifier. That table is often extracted from a data base containing more rows and/or columns, where the information is stored; the spreadsheet program should asked to save the subfile with the option “text only”. The numbers may be separated from one another by spaces, tabs, etc. and do not have to follow a fixed format (well-aligned columns). The table may contain missing values, coded by a numerical value (for instance -9, or -999, etc.) chosen in such a way as to avoid confusion with other values present in the file. At the end of the list, supplementary objects (rows) may be present; the same applies to supplementary variables, that may be written in the columns that follow the active variables; both will be positioned in the reduced space without having taken part in the computation of the eigenvalues and eigenvectors. Finally, if the objects pertain to groups that have been previously identified, a variable (integer numbers) describing group membership may be included in the data table; this will allow the program to identify groups of objects by different symbols in the object graph; that variable may be located anywhere among the columns of the file.

(2) File of results

The file of results contains the tables that the user has requested to write from the “Computation details” menu. These are: the table of covariances or correlations, the eigenvalues and eigenvectors, the positions of the variables and of the objects with respect to the first principal components.

Options of the program

This program allows to compute eigenvalues and eigenvectors either from the matrix of covariances or the matrix of correlations (which are the covariances of the standardized variables). The eigenvectors may be scaled to a length of 1 (if one is primarily interested in expressing the Euclidean distance relations among objects, in the space of reduced dimensionality) or to the square root of their respective eigenvalues (if one is more interested in representing the correlations among descriptors). Two types of rotations are also available. Supplementary objects and/or variables may be projected in the reduced space, following in this the French tradition of data analysis.

If there are missing values in the data table, two strategies are available in this program. On the

If there are missing values in the data table, two strategies are available in this program. On the one hand, the objects with missing values can be eliminated from the study (*"listwise deletion of missing values"*). On the other hand, any pair of values containing missing data may be eliminated from the calculations of the covariances or correlations (*"pairwise deletion of missing values"*); this produces covariances with an uneven number of degrees of freedom and thus, possibly, small negative eigenvalues as well. These will have to be neglected when interpreting the results; see also the discussion of negative eigenvalues, in the chapter on program PCOORD. The solution which consists of estimating the missing values is not available in this program for the time being.

Questions of the program

The questions presented by the program on the Macintosh screen are described in the following paragraphs. To start the program, click on the icon, and then on "Open" in the "File" menu.

(1) "Number of objects (rows), besides the supplementary objects" — One types the number of objects that have to be included in the computation of the eigenvalues and eigenvectors, including the objects bearing missing values.

(2) "Number of variables (columns), besides the supplementary variables and the group identification variable" — One types the number of variables that have to be included in the computation of the eigenvalues and eigenvectors.

(3) "Number of supplementary objects" — The supplementary objects, which will be positioned in the object graph without having been included in the computation of the eigenvalues and eigenvectors, must occupy the last rows of the table.

(4) "Number of supplementary variables" — The supplementary variables must occupy the columns located to the right of the columns containing the variables that will be included in the computation of the eigenvalues and eigenvectors.

(5) "Are there missing values?" [Yes, No] — See the description of the methods available to deal with missing values, in the last paragraph of the previous section (Options of the program).

(5.1) "Suppress all objects containing missing values?" [Yes, No] — Answering *Yes* to this questions means that the first method should be used (listwise deletion of missing values). If the answer is *No*, the second method will be used (pairwise deletion of missing values during the computation of the covariances or correlations), producing coefficients based on uneven numbers of object pairs.

(5.2) "Code for missing values" — One types the **numerical value** which has been used in the input data file to identify missing values (often: -1, -9, -999, etc.)

(6) "Data file" — A menu is presented showing the available ASCII files.

(7) "Computations on correlation matrix instead of covariances?" [Yes, No] — The computations should be done from the correlation matrix (which are covariances of the standardized variables) only when the variables are of different natures, or are not dimensionally homogeneous (measured in different physical units); reducing the variables (which consists of dividing each value by the variable's standard deviation) eliminates the effect of the measurement scales and produces variables without physical dimensions. When the descriptors are of the same nature, and have been measured in the same physical units, the variance-covariance (dispersion) matrix should be used instead. The principal components computed from the correlation matrix are not the same as those extracted from the dispersion matrix.

(8) "Normalization by √(lambdas) ?" [Yes, No] — Normalizing the eigenvectors to 1 preserves the

Euclidean distances among objects, in the full-dimensional space; the original axes remain orthogonal under this normalization. The reduced-space representation thus produces a projection of the original cloud of points in a few dimensions. On the other hand, the effect of normalizing the eigenvectors to the square root of their respective eigenvalues ($\sqrt{\lambda}$) is that the descriptor-axes now form angles proportional to their covariances. Angles vary from 0° (maximum positive covariance) to 180° (maximum negative covariance), an angle of 90° meaning a null covariance. That normalization is to be used when the purpose of the analysis is to represent the relations among descriptors through projections of their angles. The relationships among points (distances) are stretched by this transformation.

(9) "Extract how many eigenvalues?" — The algorithm used to compute the eigenvalues is a stepwise **[name?]** algorithm, which computes the largest eigenvalues first. The user may limit the computations to the first few eigenvalues (usually 2 to 5) which usually contain most of the variance; this may represent an appreciable saving in time with large problems (many variables).

(10) "Input file contains group identifiers?" [Yes, No] — If the objects pertain to groups identified *a priori*, a variable (positive integers) describing group membership may be included in the data table; this will allow the program to identify the groups of objects in the graph, using different symbols. That variable may be located anywhere among the columns of the data file; its position will be given at question (10.1).

(10.1) "Number of the variable identifying groups of objects" — That variable, in which group membership is coded by **positive integers**, may be located anywhere among the columns of the input data file. One types here what column it occupies. If the column thus identified contains anything but positive integers, a message is produced and the program stops.

The input data file is read here.

(11) "Title ..." — The user gives a title, which will be used as header in the graphs sent to the printer.

The eigenvalues and eigenvectors are computed here.

(12) "Varimax rotation?" [Yes, No] — The normalized Varimax rotation (Kaiser, 1958) is an orthogonal rotation of the cloud of points that attempts to simplify the columns of the eigenvector table (previously normalized to 1) by maximizing the variance of the squared saturations in each column; when the variance of the saturations is large, they tend to be close to 0 or 1. The Varimax rotation maximizes the sum of these variances for all the factors included in the rotation. In this way, groups of descriptor-axes are more likely to be near (*i.e.*, small angle) the factorial axes after rotation; this simplifies the interpretation of the factors in terms of the original variables. The amount of variance explained by a factorial subspace remains unchanged after the rotation. The factors remain uncorrelated after this orthogonal rotation. The rotation is performed for the number of factorial axes stated by the user as answer to question (9).

(13) "Harris-Kaiser rotation?" [Yes, No] — The Harris & Kaiser (1964) rotation, also called *orthoblique*, introduces a deformation of the angles among descriptor-axes. The rotation proceeds in three steps: (1) stretching ("warp") of the eigenvectors, whose importance is determined by question (13.1); (2) Varimax rotation; (3) reverse of the stretching introduced in step 1. The factors are correlated after this oblique rotation.

(13.1) "Space warp coefficient" — The amount of space stretching is determined by specifying the exponent to be given to the square roots of the eigenvalues. Accepted values are between 0 and 1, value 1 corresponding to the Varimax solution. The space warp coefficient is the same as parameter HKPOWER in procedure FACTOR of SAS.

(14) "Graph of descriptors?" [Yes, No] — These graphs show the projections of the descriptor-axes

(14) “Graph of descriptors?” [Yes, No] — These graphs show the projections of the descriptor-axes in the reduced space; the descriptor-axes, which are the original axes of the cloud of objects, are then represented by axes, not by dots.

(14.1) “Write variable numbers on graph?” [Yes, No] — The answer is *Yes* if the user wants the descriptors’ sequential numbers to be printed as identifiers in the graphs.

(14.2) “Plot how many dimensions?” — Successive graphs will be produced for all pairs of principal axes requested by the user. For instance, if one requires 3 dimensions, three graphs will be produced, respectively corresponding to axes I and II, I and III, II and III.

The graphs of descriptors are produced at this point. To increase the resolution of the picture, simply use the mouse to draw a box around any part of the picture you wish to blow up. If the user has chosen the correlation matrix as computation basis, or else the matrix of covariances with a normalization of the eigenvectors to 1 (question 8), the circle of equilibrium contributions is also shown on these graphs. Legendre & Legendre (1983) have shown that if all n descriptors contribute equally to the formation of the space reduced to d dimensions (d being the number of dimensions chosen in question 14.2), then each one should have a length of $\sqrt{d/n}$. As a consequence, if a circle is drawn with a diameter of $\sqrt{d/n}$, then any descriptor-axis which is longer than that circle contributes more to the formation of the reduced space than predicted by the equilibrium contribution of descriptors model. When the calculations are made from the covariance matrix with the eigenvectors normalized to $\sqrt{\lambda}$, the equilibrium contribution of descriptors may still be computed, but the formula is a bit more complex and does not give rise to a circle (Legendre & Legendre, 1983); this is why the circle of equilibrium contributions is not drawn in that case.

The following tables are available from the “Computation details” menu: the covariances or correlations, the eigenvalues and eigenvectors, as well as the position of the variables with respect to the principal components selected in question (9). One may go up or down the list by pointing the mouse cursor at the top or the bottom of the table. The results may be sent directly to the printer, or copied onto a file of results for future reference. In the same way, from the “Graphs” menu, graphs may be sent to the printer, or they may be preserved in PICT files, which allows to edit them using a graphics program or to include them in a manuscript using a word processor. It is necessary to “Finish” a graph before going to the next one, or to the next question.

(15) “Plot of objects?” [Yes, No] — These graphs show the projections of the objects in the reduced space; the objects are represented by dots in these graphs.

(15.1) “Write object numbers on graph?” [Yes, No] — The answer is *Yes* if the user wants the object sequential numbers to be printed as identifiers on the graphs.

(15.2) “Plot how many dimensions?” — Successive graphs will be produced for all pairs of principal axes requested by the user. For instance, if one requires 3 dimensions, three graphs will be produced, respectively corresponding to axes I and II, I and III, II and III.

The graphs are produced at this point. To increase the resolution of the picture, simply use the mouse to draw a box around any part of the picture you wish to blow up. The “Position of objects” with respect to the principal components selected in question (9) now becomes available from the “Computation details” menu. It is necessary to “Finish” a graph before going to the next one, or to the next question.

(16) “Distance comparison (Shepard’s diagram)?” [Yes, No] — This question is produced only if the user chose to carry out the computations from the covariance matrix. If the answer is *Yes*, the next questions allow to determine how the comparison will be made (Shepard diagram) between the distances in the input matrix and those in the space reduced to 2, 3, … dimensions. In that graph, a narrow cloud of points, located under the diagonal but close to it, indicates a good representation of

the original distances in the reduced space. Occasionally, points may appear above the diagonal of the Shepard diagram; these points correspond to objects for which missing values have been filled by the program (see question 5.1).

(16.1) “Shepard’s diagram: how many eigenvectors?” — One indicates how many dimensions of the reduced space will be included in that comparison of distances (generally 2 or 3).

(16.2) “XX distances to compute; would you rather sample them?” [Yes, No] — There are XX = $p(p-1)/2$ distances among p objects. When that number becomes too large (more than a few hundreds, which would cause the calculations to be too long), the user may ask the computer to randomly choose a given number of these distances. How many will be chosen is determined by question (16.3), the selection being done by a pseudo-random number generator initialized at question (16.4).

(16.3) “Number of distances to sample” — The user writes how many distances should be selected.

(16.4) “Random number generator: type a (small) integer” — A small positive integer is given, for instance 2, 5 or 10.

(16.5) “Another comparison of distances?” [Yes, No] — Answering *Yes* to this question brings you back to question (16.1).

(17) “Finish computations?” [Yes, No] — If the user wishes to make another rotation, for instance, the answer is *No*, in which case questions (12) to (16) are presented again. Answering *Yes* terminates the execution of the program.

Example

The example below presents the results of a principal component analysis of a table of physical and chemical data from 71 sampling stations in an aquatic environment; 11 variables have been recorded. Since these are expressed in different physical units (mg/L, °C, etc.), the analysis has to be conducted on the matrix of correlations among descriptors. A twelfth variable describes the fact that the observations belong to 6 different groups, which will be represented by different symbols in the results of the analysis. Three eigenvalues have been requested.

Graphs and contents of the file of results

The file of results may contain anyone of the tables requested by the user: the covariances or correlations, the eigenvalues and eigenvectors, as well as the position of the variables with respect to the principal components selected in question (9). Since that file is written in ASCII, it may easily be edited if the need arises to transfer these results to some other program. An example of a Shepard diagram is presented at the end of the results, in the chapter on program PCOORD.

```
Matrix of correlations

          1         2         3         4         5         6         7
1    1.0000
2    0.2861    1.0000
3   -0.2737   -0.0784    1.0000
4   -0.4857   -0.8501    0.0283    1.0000
5   -0.4207   -0.6456   -0.1422    0.8441    1.0000
6   -0.0502   -0.0926    0.2120   -0.0019   -0.0187    1.0000
7    0.0115   -0.2906    0.1854    0.4446    0.3940    0.0088    1.0000
8   -0.6607   -0.4657    0.0515    0.7033    0.7368    0.0875    0.2165
```

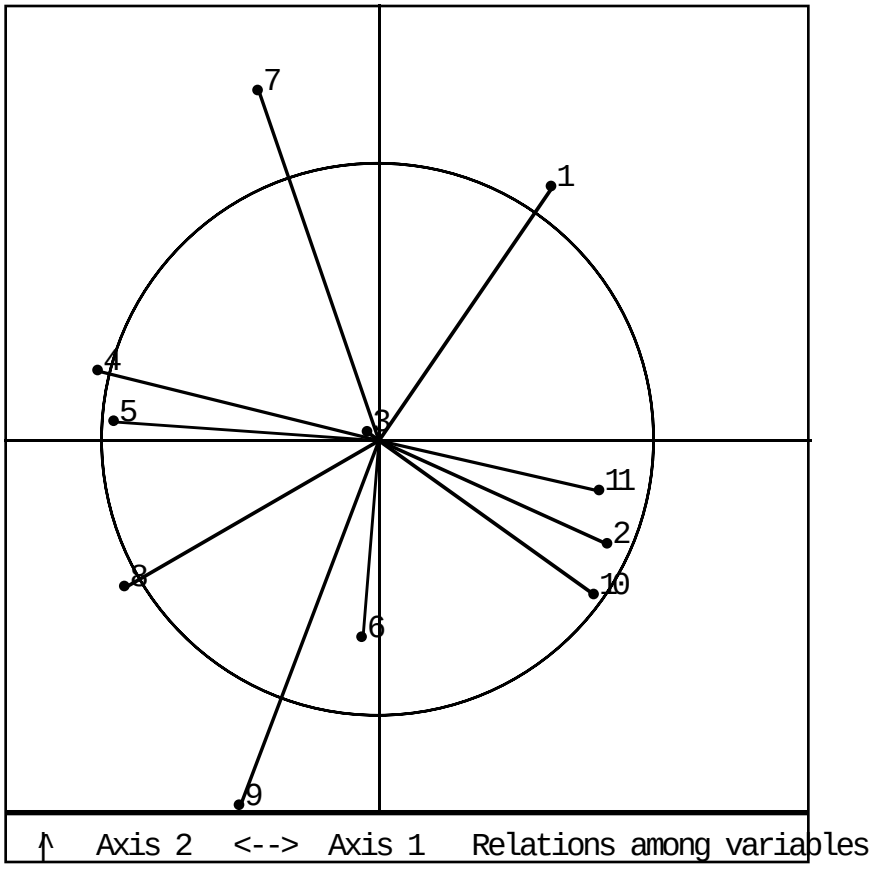
```
  8   -0.6607   -0.4657    0.0515    0.7033    0.7368    0.0875    0.2165
  9   -0.3879   -0.1814   -0.1466    0.3345    0.4184    0.2141   -0.1736
 10    0.2577    0.5017    0.0006   -0.6069   -0.5282    0.0195   -0.3897
 11    0.2701    0.4822    0.0529   -0.5764   -0.5810    0.0861   -0.1803

         8         9        10        11
  8   1.0000
  9   0.4824    1.0000
 10  -0.4594   -0.1870    1.0000
 11  -0.6025   -0.2748    0.6542    1.0000


Eigenvalues & eigenvectors

Mean of eigenvalues:   1.00000 ONE CAN ONLY INTERPRET LAMBDAS
THAT ARE LARGER THAN THIS VALUE (Ref.: Numerical ecology  p. 281)

          EIGENVALUES       % OF VARIANCE        % BROKEN STICK
            4.72621           42.96551            27.45343
            1.49324           13.57489            18.36252
            1.36044           12.36767            13.81707
```

The following criteria may help decide how many eigenvalues should be kept. On the one hand, one may decide to consider only the eigenvalues that are larger than the mean of the λ's, since one can demonstrate that a variable produced by a pseudo-random number generator would become important at or after that eigenvalue; notice that when the computations have been performed from the correlation matrix, as it is the case here, the mean of the eigenvalues is 1. On the other hand, the proportion of the total variance which is explained by the successive eigenvalues can be compared to the 'broken stick' random distribution (see Frontier, 1976, or Legendre & Legendre, 1983, p. 281). Any eigenvalue that explains more of the total variance than the corresponding fraction of the broken stick random model is worth examining.

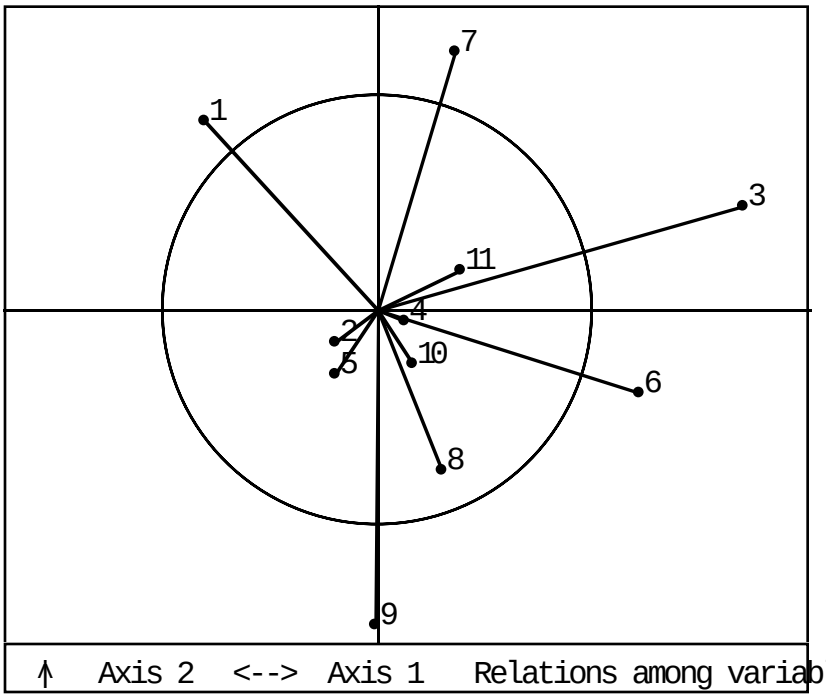
```
SELECTED EIGENVECTORS (BY COLUMNS, NORM = 1)

  1            0.26817            0.38985           -0.22411
  2            0.35148           -0.15842           -0.07806
  3           -0.01744            0.01350            0.75352
  4           -0.42825            0.10752            0.02143
  5           -0.40733            0.02935           -0.13672
  6           -0.02329           -0.30288            0.46169
  7           -0.18527            0.53790            0.26671
  8           -0.38760           -0.22506            0.02097
  9           -0.21302           -0.56205           -0.17144
 10            0.33065           -0.23811            0.05607
 11            0.33929           -0.07738            0.19866

POSITION OF OBJECTS IN THE NEW SPACE

        1         2         3
  1   2.2939   -0.7890    0.5123
  2   2.2939   -0.7890    0.5123
  3   2.5417   -0.6558    0.8178
[etc.]
 70  -0.2559    1.5296   -0.9209
 71  -0.2559    1.5296   -0.9209
```

```
POSITION OF VARIABLES IN THE NEW SPACE (Method (VARimax))

          1         2         3
 1   -0.3435    0.3767    0.1193
 2   -0.0835   -0.0635    0.3791
 3    0.7243    0.2081    0.0174
 4    0.0511   -0.0208   -0.4386
 5   -0.0852   -0.1286   -0.4021
 6    0.5194   -0.1619    0.0972
 7    0.1531    0.5109   -0.3321
 8    0.1244   -0.3145   -0.2949
 9   -0.0036   -0.6242   -0.0327
10    0.0673   -0.1070    0.3914
11    0.1652    0.0811    0.3560
```

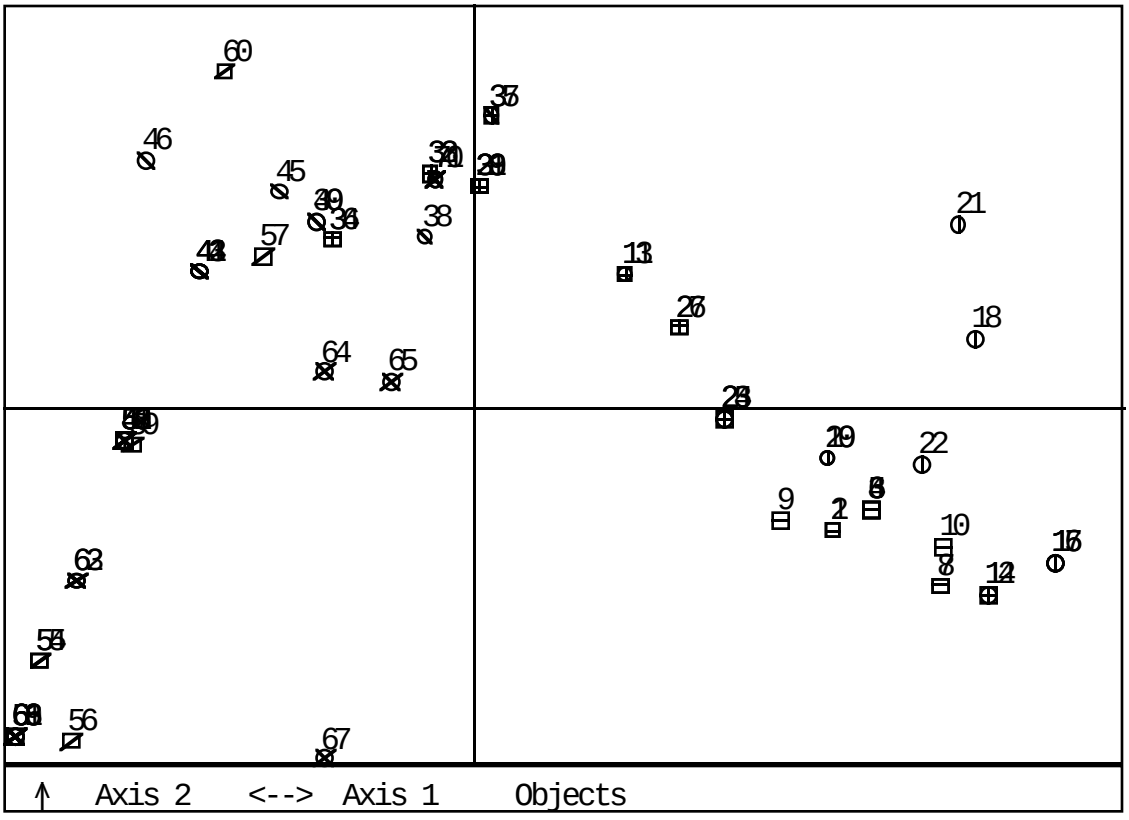
The following graphs are produced at the user's request. Here is first the graph of the projection of descriptors in the reduced space, without rotation:



The second graph represents the descriptors in the reduced space, after a Varimax rotation implying the three dimensions of the factorial space:



The third graph shows the position of the objects in the reduced space of the first two principal components. Note the symbols identifying object groups.



SIMIL

What does SIMIL do ?

SIMIL computes resemblance measures, either for binary (presence-absence) data, or for quantitative variables. The program allows the user to compute any of the measures described in Chapter 6 of the Legendre & Legendre (1983) textbook, except for the partial correlation coefficients. Table 4 lists the available coefficients, while Tables 5 to 7 summarize the criteria that may guide the user when choosing a coefficient. Four types of files, whose role is explained in detail in the next section, may be used in conjunction with this program; the dashed arrows identify files that are only used in connection with some of the coefficients.

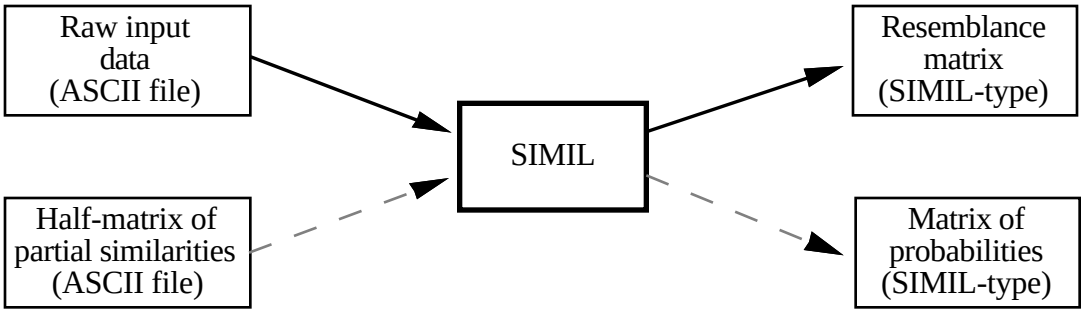


Table 4 - The association coefficients of the SIMIL program. The code recognized by the program for each coefficient is found in the left-hand column. Symmetrical coefficients include double absences in the measure of resemblance, while asymmetrical coefficients do not take them into account.

Binary coefficients including double-zeros (symmetrical)		
S01	(a+d)/(a+b+c+d)	Simple matching coefficient (Sokal & Michener)
S02	(a+d)/(a+2b+2c+d)	(Rogers & Tanimoto)
S03	(2a+2d)/(2a+b+c+2d)	
S04	(a+d)/(b+c)	
S05	(1/4) [a/(a+b) + a/(a+c) + d/(b+d) + d/(c+d)]	
S06	ad/√[(a+b)(a+c)(b+d)(c+d)]	
Binary coefficients excluding double-zeros (asymmetrical)		
S07	a/(a+b+c)	Coefficient of community (Jaccard)
S08	2a/(2a+b+c)	(Sørensen, Dice)
S09	3a/(3a+b+c)	
S10	a/(a+2b+2c)	
S11	a/(a+b+c+d)	(Russell & Rao)
S12	a/(b+c)	(Kulczynski)
S13	(1/2) [a/(a+b) + a/(a+c)]	(Kulczynski)
S14	a/√[(a+b)(a+c)]	(Ochiai)
S26	[a + (d/2)]/(a+b+c+d)	(Faith)

Table 4 ***(continued)***

Quantitative coefficients including double-zeros (symmetrical)		
S15	Σ(w[i] s[i]) / Σ(w[i])	(Gower, symmetrical)
S16	Σ(w[i] s'[i]) / Σ(w[i])	(Estabrook & Rogers)
Quantitative coefficients excluding double-zeros (asymmetrical)		
S17	2W/(A+B)	(Steinhaus)
S18	(1/2) [(W/A) + (W/B)]	(Kulczynski)
S19	Σ(w[i] s[i]) / Σ(w[i])	(Gower, asymmetrical)
S20	Σ(w[i] s'[i]) / Σ(w[i])	(Legendre & Chodorowski)
S21		Chi-square similarity (Roux & Reyssac)
Probabilistic coefficients		
S22		Chi-square probabilistic similarity
S23		Goodall's probabilistic coefficient
Binary coefficients for R-mode analysis (species associations, etc.)		
S24	[a/√((a+b)(a+c))] - 0.5√(a+c)	(Fager & McGowan)
S25	1 - *p*(chi square)	(Krylov)
Genetic similarity coefficient		
NEI		Nei's genetic similarity (bounded between 0 and 1)
Distance coefficients		
D01		Euclidean distance
D02		Taxonomic, or average distance
D03		Chord distance
D04		Geodesic metric
D05		Mahalanobis generalized distance (among groups)
D06		Minkowski metric (the user specifies the power)
D07		Manhattan metric
D08		Mean character difference (Czekanowski)
D09		Index of association (Whittaker)
D10		Canberra metric (Lance & Williams)
D11		Coefficient of divergence (Clark)
D12		Coefficient of racial likeness (among groups; Pearson)
D13		Nonmetric coefficient (Watson, Williams & Lance)
D14		Percentage difference (Odum; Bray & Curtis)
Coefficients of dependence between descriptors (R mode)		
RP		Pearson's *r*
RS		Spearman's *r*
TAU		Kendall's *tau*
CHI		*G* statistic (Wilks' chi-square)
HT		Tschuproff's contingency coefficient
HS0	B/(A+B+C)	Reciprocal information coefficient (Estabrook)
HS1	√[1 - (HD)**2]	Coherence coefficient (Rajski)
HS2	B/(A+2B+C)	Symmetric uncertainty coefficient
HD	(A+C)/(A+B+C)	Rajski's metric

Table 5 - Choice of an asymmetrical association measure among objects (Q mode) for species abundance data, or other descriptors for which double absences do not indicate resemblance. Modified from Legendre & Legendre (1983 and 1984a).

1) Presence-absence data, or relative abundance scale without partial similarities among classes	see 2
2) Metric coefficients: S07, S10, S11, S26	
2) Semimetric (or pseudometric) coefficients: S08, S09, S13, S14	
2) Nonmetric coefficient: S12	
1) Quantitative data	see 3
3) Raw data	see 4
4) Without probability level	see 5
5) Without standardization per object; a given difference between two objects, for abundant or for rare species, contributes the same to the similarity: S17, S18	
5) With standardization per object-vector; differences between objects for the most abundant species (in the whole file) contribute more to the similarity (less to the distance): S21	
4) Probabilistic coefficient: S22	
3) Normalized data (or, at least, non-skewed distribution), or relative abundance scale	see 6
6) Without probability level	see 7
7) Without standardization per object	see 8
8) A given difference between two objects, for abundant or for rare species, contributes the same to the similarity: S17, S18, D08, D14	
8) Differences between objects for the most abundant species (in the two objects under consideration) contribute more to the similarity (less to the distance): D10, D11	
8) Differences between objects for the most abundant species (in the whole data file) contribute more to the similarity (less to the distance): S19, S20	
7) With standardization per object-vector; for objects of equal *importance*, abundant and rare species contribute the same to these measures: D03, D04 (where the importance is measured by the length of the vector), D09 (where it is measured by the sum of the elements of the vector)	
6) Probabilistic coefficient: S23	

Input and output files

(1) Main input data file

In the input data file, the data are written in ASCII as real or integer, positive or negative numbers. The SIMIL program always computes resemblance coefficients among the rows of the input file; so, make sure that the rows represent the objects when computing a similarity or a distance coefficient (Q mode of analysis), or the descriptors when computing a coefficient of dependence (R mode). Program VERNORM allows to check the content of data files, to transpose matrices or to normalize descriptors, if needed:

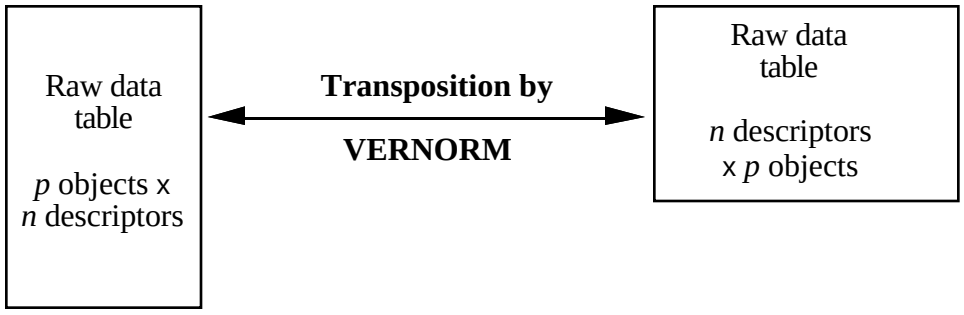


Table 6 - Choice of a symmetrical association measure among objects (Q mode) for physical, chemical, geological descriptors, etc. Modified from Legendre & Legendre (1983 and 1984a).

1) Comparison of individual objects	see 2
2) Binary descriptors, or multistate descriptors without partial similarities	see 3
3) Metric coefficients: S01, S02, S06	
3) Semimetric (or pseudometric) coefficients: S03, S05	
3) Nonmetric coefficient: S04	
2) Multistate descriptors	see 4
4) Quantitative, dimensionally homogeneous descriptors	see 5
5) Differences stressed by squaring: D01, D02	
5) Attenuated differences: D07, D08	
4) Descriptors not dimensionally homogeneous; equal (or different, depending on the imposed w_i values) weights are imposed to the various descriptors	see 6
6) Qualitative descriptors (without partial similarities), and quantitative descriptors with partial similarities based on the range of variation of each descriptor: S15	
6) Qualitative descriptors (partial similarity matrices among states are authorized), and quantitative or semi-quantitative descriptors with a partial similarity function for each descriptor: S16	
1) Comparison of groups of objects	see 7
7) Taking into account the correlations among descriptors: D05	
7) Without taking into account the correlations among descriptors: D12	

In the Q mode of analysis, the objects are the successive rows of that matrix; in a row, the various descriptors are written one after the other. An object, however, may take as many lines of the file as necessary to accommodate all its descriptors. Since reading the data is done in free format, the descriptors must be separated from one another by one or more blank spaces. The number of spaces does not matter; at the limit, a single data could be written per line of the file. A consequence of that input flexibility is that missing values cannot be represented by blank spaces, since those are ignored when the data are read in; missing values must be materialized by a numerical code (0, -1, -9 or -999 are often used), which will be declared in answer to one of the questions of the program. That code must differ in a non-ambiguous way from all the numerical values that may legitimately be found in the data file.

With distance coefficients D05 and D12, the measures are computed among groups of objects. It is necessary for the members of the same group to be located one after the other in the input file. Membership of the objects to the groups cannot be specified by a coded variable; the program will ask the user to state, for each group in turn, how many objects are members of the group.

In principle, there is no limit to the size of the matrices that can be handled by the Macintosh version of this program. The program occupies all the available memory space (RAM), so that in practice, the size of the matrices that can be handled is a function, not only of the amount of memory available in the machine, but also of the version of the System file in use, as well as the simultaneous use of MultiFinder, of a RAM cache, or of other programs. Versions 3.0 of SIMIL and above make all their computations in central memory in order to speed them up; if there is not enough memory available to handle a data table, the following message is produced:

```
Not enough memory! Try an older SIMIL version
```

The first solution consists of inactivating MultiFinder, if in use. If the problem is not solved, one may try using a copy of SIMIL with a version number smaller than 3; these older versions keep most of the data on disk, so that they may handle larger files, the cost being that they are much slower. Notice that

Table 7 - Choice of a dependence coefficients among descriptors (R mode). Modified from Legendre & Legendre (1983 and 1984a).

1) Descriptors: species abundances	see 2
2) Raw data: S21, RS, TAU	
2) Normalized data	see 3
3) Without probability level: RP (after elimination of the double zeros, as far as possible); RS, TAU	
3) Probabilistic coefficients: probability associated to RP, RS and TAU; S23	
2) Presence-absence data	see 4
4) Without probability level: S7, S8, S24	
4) Probabilistic coefficient: S25	
1) Other descriptors: physical, chemical, geological, etc.	see 5
5) Without probability level	see 6
6) Linearly related quantitative descriptors: RP	
6) Other ordered, monotonically related descriptors: RS, TAU	
6) Ordered descriptors in non-monotonic relation, and qualitative descriptors: CHI, HT, HS0, HS1, HS2, HD	
5) Probabilistic coefficients	see 7
7) Linearly related quantitative descriptors: probability associated to RP	
7) Other ordered, monotonically related descriptors: probability associated to RS, TAU	
7) Ordered descriptors in non-monotonic relation, and qualitative descriptors: probability associated to CHI	

the versions of SIMIL with number smaller than 3 are voracious in disk space required to run the program; this was the primary reason to write version 3. In the CMS and VMS versions, on the other hand, the size of the files that can be analyzed is limited by parameters (constants) at the beginning of the program; the user may change them according to need, after which the program must be recompiled.

The program allows the user to write identifiers at the beginning of each object-vector, but not at the head of the columns. If the user declares that there are identifiers in the data file, the program assumes that these occupy the first 10 characters of each object-vector (rows or blocks of rows); any alphanumerical character may be used in these identifiers, including blanks. If this is the case, the list of descriptors proper starts in column 11 or after. That convention is the same as in Prof. Joseph Felsenstein's PHYLIP phylogenetic analysis package. The file shown below, with identifiers, would be an acceptable file for SIMIL (fishing results: 6 objects, 4 descriptors; missing values are coded -9); the 10 spaces reserved for the identifiers are materialized by underscoring:

```
fish1____1 3.2 4 5
fish2____  1           2.9            3     4
tin can__
2 0.9 -9 -9
Glad bag_
2
15.0
-9 -9
fish3____
          1 3.5 4 20
scrap tire2 75.4 -9 -9
```

Notice that the files analyzed by SIMIL are often extracted from larger data bases managed by spreadsheet or statistical analysis programs; so they often look like the following file, if there are identifiers in columns 1 to 10:

```
Stat.100    2    4    3    1    1    4    1    1    1    2    2
Stat.200    2    4    3    1    1    4    1    1    1    2    2
Stat.320    2    4    3    1    1    4    1    1    1    1    4
Stat.330    2    4    3    1    1    4    1    1    1    1    4
Stat.340    2    4    3    1    1    4    1    1    1    1    4
```

or else, without identifiers:

```
-0.38566  -1.42712  37.1  8.24931  0.02627  0.85015
-0.01005   0.77932  37.5  7.34987  0.01033  0.77932
 0.10436   0.94391  37.1  7.09589  0.16279  0.49348
 0.33647   0.71295  37.4  6.79571  0.09373  0.57098
 0.30748   0.52473  37.3  6.57508  0.14691  1.39128
```

CAUTION: As with all the other programs of this package, which is written in PASCAL, one must write "0.376" for instance, instead of ".376", and “-0.42” instead of “-.42”, when using a version of SIMIL older than version 3, or else the CMS or VMS versions of the program; see notes on page 6.

(2) Partial similarity matrix input file

For coefficients S16 and S20, the program asks how many partial similarity matrices there are. If matrices of this type are used to quantify the relations among classes of semi-quantitative, qualitative, or circular variables (see Legendre & Legendre, 1983, chapter 6), one must create a second input file which must contain the following information for each partial similarity matrix:

1- The sequential number of the descriptor (column) to which the matrix applies.

2- The size of the partial matrix, which is equal to the maximum number of values (states) that may be taken by this descriptor.

3- The values of partial similarity themselves, expressed as real numbers, filling the lower triangular state-by-state matrix, excluding the diagonal. If the qualitative descriptor in question is made of *n* classes (states), there must be (*n**(*n*-1))/2 values in the partial similarity matrix.

Consider for example that descriptors 2 and 4 require the following partial similarity matrices, each one being here of order 5:

descriptor 2 (5 classes):

```
  1
0.4    1
0.5   0.6    1
0.5   0.4   0.45   1
0.46  0.47  0.5    0.5    1
```

descriptor 4 (5 classes):

```
  1
0.4    1
0.3   0.8    1
0.9   0.2   0.55   1
0.48  0.9   0.2    0.8    1
```

The input file of the partial similarity matrices would then be the following:

```
2  5
0.4
0.5  0.6
0.5  0.4  0.45
0.46  0.47  0.5  0.5
4  5
0.4  0.3  0.8  0.9  0.2  0.55  0.48  0.9  0.2  0.8
```

The first line gives the sequential number of the descriptor to which the first partial similarity matrix applies, as well as the number of classes; line 6 gives the same information for the second partial matrix. The successive rows of the first matrix are written on separate physical lines, while the successive rows of the second matrix are written on the same physical line; both formats are acceptable.

(3) Main output file

The output file, which is described below, contains the computation results; it is written in binary and is ready to be read back by the various data analysis programs of the "R" package (see pages 3 and 7 of the present manual). One may also read the content of the binary matrix using program LOOK, which can transcribe it into readable characters (ASCII), or using the procedure listed below. Note that the binary matrices written by SIMIL on one type of computer cannot be used by the "R" programs residing on another type of computer, because of the differences that exist in the lengths and structures of the machine words.

The structure of that binary file is the following:

- First word (integer): number of rows (lines, or blocks of lines) among which the resemblance function has been computed.

- Second word (integer): Number of columns in the raw input data file.

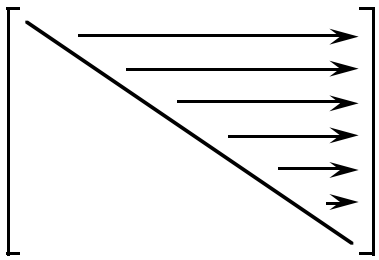
- Words 3 to 12 (characters): Title given to the file by the user.

- Word 13 (characters): Date of creation of the file.

- Words 14 and 15 (characters): Name of the resemblance coefficient used.

- Word 16 (integer): Number (k) of identifiers to be read; if there were no identifiers at the beginning of the rows, in the raw input data file, $k = 0$. If $k > 0$, that information is followed by a list of k machine words containing one row identifier each (10 characters).

- Next comes the list of resemblance measures, following the successive rows of the upper triangular matrix, diagonal excluded. One machine word (real number) is used for each resemblance value.



The PASCAL procedure that follows indicates how to read that binary file of similarities. Notice that this procedure is now written for CMS; to use it in order to read back SIMIL-type matrices computed on Macintosh, one has to know that it is necessary to write "READ(INFILE,VAL);" for instance, instead of "VAL:=INFILE@;GET(INFILE);". Also, while the CMS and VMS versions of SIMIL use 8-byte real numbers, the Macintosh version of SIMIL uses 10-byte real numbers (defined as "Extended" in MPW PASCAL). The user who is sufficiently familiar with PASCAL to decide to incorporate this procedure in her own program is likely to know also what other adaptations are necessary for her own computer and compiler.

```
PROCEDURE TypeSIMIL;
CONST NBWORDS=1;
      NBCAR=10;
(* ================================================================ *)
(* This procedure reads a SIMIL-type binary matrix and prints it out  *)
(* in readable form. The resemblance measures are also written out    *)
(* in binary form in a square matrix called MAT.                      *)
(* ================================================================ *)

TYPE MATRIXX = ARRAY[1..500, 1..500] OF REAL;
     VARIABLE= RECORD CASE INTEGER OF
               1:(INT:INTEGER);
               2:(RE:REAL);
               3:(CAR:PACKED ARRAY[1..NBCAR] OF CHAR);
               END;
VAR  I,J,NOBJ,NBDESC:INTEGER;
     INFILE: FILE OF VARIABLE;
     MAT: MATRIXX;
     VAL: VARIABLE;

   PROCEDURE WRITEDATA;
   VAR I,J,K:INTEGER;
   BEGIN
   J:=0;
   FOR I:=1 TO NBWORDS DO
      BEGIN
      VAL:=INFILE@;GET(INFILE);
      FOR K:=1 TO NBCAR DO
            BEGIN
            J:=J+1;
            IF J<=10 THEN WRITE(OUTFILE,VAL.CAR [K]);
            END;
      END;
   END;                                                (* End of WRITEDATA *)

BEGIN
RESET(INFILE);
WRITELN(OUTFILE,'INPUT FILE:');
VAL:=INFILE@;GET(INFILE);
NOBJ:=VAL.INT;                    (* NOBJ: number of rows in the input data matrix *)
WRITELN(OUTFILE,'    NUMBER OF OBJECTS: ',NOBJ:4);          (* Printing NOBJ *)
VAL:=INFILE@;GET(INFILE);
NBVAR:=VAL.INT;                                  (* NBVAR: number of columns *)
WRITELN(OUTFILE,'    NUMBER OF VARIABLES: ',NBVAR:4);     (* Printing NBVAR *)
WRITE  (OUTFILE,'    TITLE: ');
```

```
FOR I:=1 TO 10 DO WRITEDATA;                                    (* Printing the title *)
WRITELN(OUTFILE);
WRITE  (OUTFILE,'    DATE: ');
WRITEDATA;                                                      (* Printing the date *)
WRITELN(OUTFILE);
WRITE  (OUTFILE,'    RESEMBLANCE COEFFICIENT: ');
WRITEDATA;WRITEDATA;            (* Printing the name of the ressemblance coefficient *)
WRITELN(OUTFILE);
WRITELN(OUTFILE); WRITELN(OUTFILE,'LIST OF IDENTIFIERS:');
VAL:=INFILE@;GET(INFILE);             (* Reading how many identifiers are present *)
IF VAL.INT<> 0 THEN                                          (* Reading the identifiers *)
      BEGIN
      FOR I:=1 TO NOBJ DO
            BEGIN
            FOR J:=1 TO NBWORDS DO WRITEDATA;          (* Printing the identifiers *)
            WRITELN(OUTFILE);
            END;
      END;
WRITELN(OUTFILE); WRITELN(OUTFILE,'LIST OF RESEMBLANCE VALUES:');
FOR I:=1 TO NOBJ DO                                (*  Reading the resemblance values *)
      BEGIN
      MAT[I,I]:=1.0;
      FOR J:=I+1 TO NOBJ DO
            BEGIN
            VAL:=INFILE@;GET(INFILE);
            WRITELN(OUTFILE,VAL.RE:10:5);       (* Printing the resemblance values *)
            MAT[I,J]:=VAL.RE;    (* The same values are written in binary form in MAT *)
            MAT[J,I]:=MAT[I,J];
            END;
      END;
END;
```

(4) Output file for the probability matrix

When a test of statistical significance is associated with a coefficient (S23, S25, D05, RP, RS, TAU, CHI), a probability matrix written in binary form may be obtained in a second output file, if requested by the user. By default, that file is called "PROBAB DATA A" in the CMS version. Actually, it is the complement of the probability of the null hypothesis which is written in that file **[revise?]**. In this way, high values of probabilities are obtained for a significant coefficient, and a low value when there is no relation between the two objects or descriptors in question; so, these values of probability behave like similarity measures and may be used as such in clustering or ordination programs. Contrary to correlation coefficients, for example, in which the two ends of the scale correspond to a strong relation (positive or negative) between the two descriptors, the values of probabilities have a monotonic behavior on their scale of variation [0, 1]. That file may be read back using program LOOK.

Options of the program

The program offers as options the 50 measures of resemblance enumerated in Table 4. A detailed discussion of these functions is beyond the scope of the present document; one may refer to the text of Legendre & Legendre (1983) or to one of the following reviews of resemblance coefficients: Sokal & Sneath (1963), Williams & Dale (1965), Cheetham & Hazel (1969), Sneath & Sokal (1973), Clifford & Stephenson (1975), Daget (1976), Blanc *et al.* (1976), Orlóci (1978), Gower (1985). The criteria

that may guide users in the choice of a coefficient are summarized in Tables 5 to 7. The next section shown how the dialogue of the program with the user differs depending on the selected resemblance measure.

Questions of the program

The questions presented by the program on the Macintosh screen are described in the following paragraphs. The questions of the CMS and VMS versions are essentially the same, as can be seen in the example presented in the next section. To start the program, click on the icon, and then on "Open" in the "File" menu.

(1) "Input file" — The program presents a list of the available ASCII files.

(2) "Title:" — The title given here will be written in the block of information included with each SIMIL-type binary file; see program LOOK for details.

(3) "Number of rows (lines or blocks of lines)" — The answer to this question must be a positive integer number. In Q mode, the user gives here the number of objects, each object possibly occupying one or several lines in the data file; in the R mode, where the matrix has been transposed, it is the number of variables which has to be given here, each variable possibly occupying again one or several lines of the data file; refer to the description of the main input data file.

(4) "Number of columns" — In Q mode, the answer to this question is the number of variables describing each object, to the exclusion of the row identifiers if present in the file. In the R mode, where the file has been transposed, the number of objects composing each variable-vector is given, to the exclusion of the descriptor identifiers, if present in the file.

(5) "Code for missing values (default: 0)" — The numerical value is given which has been used in the file to indicate that an information is missing (-1, -9, -999, etc. are often used codes). That question has to be answered by a numerical value, even when there are no missing values in the file.

(6) "The 10 first characters of each row are identifiers" [Yes, No] — The answer is *Yes* if the first 10 columns of each object-vector or descriptor-vector contain a row identifier; see section on the main input data file.

(7) "Compute" [Similarities, Distances, Other; Information] — If the user chooses Similarities, a second menu offers the choice between similarities S1 to S26 of Table 4; if the choice is Distances, the menu gives the choice between distances D1 to D14; finally, Other leads to a new menu offering the choice between functions Tau, Pearson's R, Spearman's R, Chi, Ht, Hs0, Hs1, Hs2, Hd and Nei. Information gives access to a file containing Table 4. One may go up or down that file by pointing the mouse cursor at the top or the bottom of the screen; the table may also be sent to the printer if the user wants to obtain a hard copy.

In the CMS and VMS versions, the question is presented as follows; it contains the same choices:

```
WHAT RESEMBLANCE FUNCTION DO YOU WANT TO COMPUTE ?
Similarities  : s01 to s26, Nei
Distances     : d01 to d14
R-mode:    rp = Pearson's r
           rs = Spearman's r
           tau= Kendall's tau
           chi= Chi-square (G statistic)
           ht = Tschuproff's contingency coefficient
```

```
hs0= Reciprocal information        S=B/(A+B+C)
hs1= Rajski's coherence     S'=SQRT(1-(hd)**2)
hs2= Symmetric uncertainty coeff.S"=B/(A+2B+C)
hd = Rajski's metric          D =(A+C)/(A+B+C)
```

(8) “Output file” — The program presents a menu allowing to give a name to the binary file of resemblance values to be computed.

From this point and on, the questions of the program differ depending on the type of coefficient that the user wants to compute. These questions are followed by the computation of the coefficient values, after which the program goes back to the “File” menu, allowing to handle another input data file immediately. Command “Interrupt” in the “R: Simil” menu allows to stop the program.

Binary coefficients: S1 to S14, S24, S25, S26, D13

(9) “Threshold from which information will be coded 1 (smaller: 0)” — If the input data file contains only 0’s and 1’s, answer “1” to this question. If it contains quantitative data instead, it is possible to make the program handle these as if they were presence-absence data. We could have decided that all strictly positive values (larger than 0) would be recoded “1”; however, with species abundance data for instance, cases may occur where the user decides to consider as absent any species that is not represented, for instance, by at least 10 individuals at the sampling station; the answer to this question would be “10” in such a case. In general, the answer to this question indicates the threshold from which the user asks the program to consider a species as “present”. The smallest admissible value is “0”.

Simple quantitative coefficients: D1 to D4, D7 to D11, D14, NEI

No supplementary question is asked by the program before computing these coefficients.

The Minkowski metric: D6

(9) “Power for this function” — The answer is a positive integer, which provides the exponent “r” to the Minkowski metric whose formula is $D6(\mathbf{x}_1, \mathbf{x}_2) = [\ \Sigma\ |y_{i1} - y_{i2}|^r\ ]^{(1/r)}$. The Manhattan metric (D7) corresponds to the Minkowski metric to the power 1, while the Euclidean distance (D1) is the Minkowski metric to the power 2.

Distances among object groups: D5 and D12

(9) “File of associated probabilities” [Yes, No] — That question is presented only for coefficient D5 (Mahalanobis generalized distance), since coefficient D12 (coefficient of racial likeness) does not lead to a test of statistical significance of the differences computed among groups. If the answer is *Yes*, the program asks the name to be given to the binary SIMIL-type file of probabilities.

(10) “Cardinality of group 1” — One indicates here (positive integer) how many objects are found in the first group. The program then asks “Cardinality of group 2”, etc., until all objects, whose number is known from question 3, have been accounted for.

Gower’s coefficients: S15 and S19

(9) “Range on data rather than on population” [Yes, No] — The formula of Gower’s coefficients is $D(\mathbf{x}_1, \mathbf{x}_2) = \Sigma\ w_{i12}\ s_{i12} / \Sigma w_{i12}$. The weights w_i will be decided at question (10). We are interested here in the partial similarity function s_{i12} between objects $\mathbf{x}_1$ and $\mathbf{x}_2$ for quantitative descriptors. In that case, the difference between the values of the descriptor for these two objects, $|y_{i1} - y_{i2}|$, is divided by the maximum value of the range R_i between values of the descriptor; the question wants to

determine if that ranges R_i should be computed from the data available in the input data file (answer *Yes*), or if the user wants to provide other values for the ranges R_i which are known from a study of the reference population from which the sample under study has been extracted (answer *No*). The partial similarity s_{i12} between objects $\mathbf{x}_1$ and $\mathbf{x}_2$ is computed as $s_{i12} = 1 - [\,|y_{i1} - y_{i2}| / R_i\,]$.

(9.1) "Range of variable 1" — If the answer to the previous question was *No*, indicating that values for R_i will be provided by the user, the program now asks to give the value for variable 1 (positive real number). The program then asks: "Range of variable 2", etc., until all variables have been attributed a range. A dummy value must be given for the qualitative multiclass descriptors, if present in the file; they will only be identified at questions 11.

(10) "All weights (W[i]) are binary (0 or 1)" [Yes, No] — The w_i values have two different roles in the formula of these coefficients. On the one hand, they provide variable weights to the various descriptors, if the user so desires; to rule out that possibility, answer *Yes* to the question, which gives equal weights by default to all descriptors in the computation of the global similarity. The second role of these values is to provide a way of eliminating from the computation any descriptor for which one of the two objects possesses a missing value (whose code has been given in question 5); when there are missing values, the descriptor receives a weight $w_i = 0$. Finally, in the asymmetrical form of the coefficient (S19), $w_i = 0$ when the species is absent from both object-vectors ($y_{i1} + y_{i2} = 0$).

(10.1) "W[1]" — If the answer to question (10) was *No*, a weight must now be given for descriptor no. 1. The answer must be a real number ≥ 0; a weight of zero produces the elimination of the corresponding descriptor from the calculations. The program then asks: "W[2]", etc., until all descriptors have received a weight.

(11) "Number of multistate qualitative descriptors" — The multistate qualitative descriptors are handled by coefficient S15 in the manner of the simple matching coefficient for multistate data; a partial similarity $s_{i12} = 1$ is counted if the two objects agree on the state of that descriptor, and 0 if they disagree. If the file contains qualitative descriptors that must be handled in this way, the user must say here how many descriptors of that type there are. That question is presented only for the symmetrical form of the coefficient (S15); in the asymmetrical form (S19), reserved for frequency data (species abundances, in ecology), that question would not make sense.

(11.1) "Descriptor identifier: 1" — The order number of the first qualitative descriptor is given here, among the descriptors in the data file. The program then asks: "Descriptor identifier: 2", etc., until all the qualitative multistate descriptors have been identified. If the number of qualitative descriptors declared in question 11 is equal to the total number of descriptors in the file (question 4), that question does not appear.

Coefficients S16 and S20

(9) "Number of partial similarity matrices" — Coefficients S16 and S20 have the same general formula as Gower's coefficients, that is, $D(\mathbf{x}_1, \mathbf{x}_2) = \Sigma\, w_{i12}\, s'_{i12} / \Sigma w_{i12}$. The weights w_i will be decided at question (11). We are interested here in the partial similarity function s'_{i12} between objects $\mathbf{x}_1$ and $\mathbf{x}_2$; this is where S16 and S20 respectively differ from S15 and S19. The value of s'_i may be determined in two different ways: either by using the monotone decreasing function described at question (10), or by imposing predetermined values of partial similarities among the classes (states) of a qualitative, semi-quantitative or circular descriptor. Legendre & Legendre (1983, p. 184) provide an example for such a matrix of partial similarities. These matrices, one for each descriptor that has to be handled in that way, must be written one after the other in a separate file, following the instructions of the section "Partial similarity matrix input file". The answer to the present question of the program must be an integer ≥ 0, zero ("0") meaning that no partial similarity matrix is provided. if the answer given is a positive integer, the program presents a menu of the available ASCII files; the user indicates which one contains the partial similarity matrices. These matrices represent the only method available

to impose partial similarities among the states of a qualitative descriptor or of a circular variable. Coefficients S16 and S20 are quite useful to handle data tables containing mixtures of descriptor types (quantitative, semi-quantitative and qualitative).

(10) “Same value of K[i] for all descriptors” [Yes, No] — Estabrook & Rogers (1966) have proposed to estimate the partial similarity between the values of a quantitative descriptor through a partial similarity function between the values of that descriptor, which is a function both of the distance $d_{i12} = |y_{i1} - y_{i2}|$ between the values taken by two objects for that descriptor, and of a parameter k_i set by the user that limits the extension of the partial similarity to a maximum distance of k_i. The equation of that empirical function is $s'_{i12} = f(d_{i12}, k_i) = 2 (k_i + 1 - d_{i12}) / (2 k_i + 2 + d_{i12} k_i)$ if $d_{i12} \leq k_i$ and $s'_{i12} = 0$ when $d_{i12} > k_i$. Furthermore, with species abundance data (coefficient S20), $s'_{i12} = 0$ when y_{i1} or y_{i2} are zero (Legendre & Chodorowski, 1977). Examples of the use of that function are found in the Estabrook & Rogers (1966) paper, as well as in Legendre & Legendre (1983, p. 182-183). The present question of the program tries to determine whether different values of k_i will be attributed to the various descriptors.

(10.1) ”K[i]” — If the answer to question (10) was *Yes*, the single value of k_i to be used for all descriptors is given here. That value is a real number ≥ 0; $k_i = 0$ means that the descriptor should be treated in the manner of the multistate simple matching coefficient (with S16) or of the multistate Jaccard coefficient (with S20): a partial similarity $s_{i12} = 1$ is given when the two objects agree for that descriptor, and 0 if they disagree.

(10.2) ”K[1]” — If the answer to question (10) was *No*, the value of *k* to be used for descriptor no. 1 is given here. The program then asks: ”K[2]”, etc., until all descriptors have received a value of *k*. A dummy value must be given for the qualitative multistate descriptors that were handled in question (9).

(11) “All weights (W[i]) are binary (0 or 1)” [Yes, No] — The w_i values have two different roles in the formula of these coefficients. On the one hand, they provide variable weights to the various descriptors, if the user so desires; to rule out that possibility, answer *Yes* to the question, which gives equal weights by default to all descriptors in the computation of the global similarity. The second role of these values is to provide a way of eliminating from the computation any descriptor for which one of the two objects possesses a missing value (whose code has been given in question 5); when there are missing values, the descriptor receives a weight $w_i = 0$. Finally, in the asymmetrical form of the coefficient (S20), $w_i = 0$ also when the species is absent from both object-vectors ($y_{i1} + y_{i2} = 0$).

(11.1) “W[1]” — If the answer to question (11) was *No*, a weight must now be given for descriptor no. 1. The answer must be a real number ≥ 0; a weight of zero produces the elimination of the corresponding descriptor from the calculations. The program then asks: “W[2]”, etc., until all descriptors have received a weight.

The chi-square probabilistic similarity: S22

(9) “Wilks’ chi square rather than Pearson’s” [Yes, No] — Coefficient S22 is the one-complement of the probability associated with the chi-square statistic computed on the frequency table formed by two samples and *n* species, after having excluded double absences from the computations. The user may choose between Wilks’ chi-square statistic (also called the *G* statistic: answer *Yes*) or the Pearson chi-square statistic (answer *No*).

Goodall’s probabilistic coefficient: S23

(9) “Computation with Gower’s index rather than Steinhaus’ ” [Yes, No] — Coefficient S23 is the one-complement of the probability that two samples chosen at random be as similar as, or more similar than the pair of samples under study. The partial similarities, per species, on which the probability calculations are based, may be computed in the manner of the Gower S19 index (answer *Yes*) or in

that of the Steinhaus index S17 (answer *No*), as explained in Legendre & Legendre (1983, p. 190).

Dependence coefficients among descriptors (R mode): from RP to HD

(9) "File of associated probabilities" [Yes, No] — The answer is *Yes* when the user wants to obtain the file of probabilities associated with the selected dependence coefficient. The program then asks the name to be given to that SIMIL-type binary file; that file is described in the "Output file for the probability matrix" section, above. It may be examined using program LOOK.

(10) "Tau A and B rather than Tau A, B & C" [Yes, No] — There are three versions for Kendall's *tau* nonparametric correlation coefficient: tau_a is used when there are no ties in the data, tau_b when there are ties and the two variables possess the same number of semi-quantitative classes, and tau_c when there are ties but the number of classes is not the same in the two descriptors. Program SIMIL chooses the adequate version in each situation. However, there are authors who recommend not to use the tau_c, correction formula anymore, but to use tau_b in all cases where there are ties instead; users of the SIMIL program may decide to compute only tau_a and tau_b (answer *Yes*) if they wish. That question is only presented on the screen when the coefficient chosen is Kendall's *tau*.

Example

How to use SIMIL on mainframes is illustrated by the following example. The input data file contains 71 objects and 11 descriptors; missing values are coded "-9". Coefficient S15 will be computed among the rows of the input file; the questions that apply specifically to that function are discussed in the previous section. Coefficient S15 cannot use partial similarity matrices; as a consequence, no answer is given to the question of the calling program requesting a name for that file (see 1 in the left-hand margin). In the same way, no answer is given to the question requesting a name for the file of probabilities since coefficient S15 does not produce associated probabilities (see 2 in the left-hand margin). The dialogue of the CMS version is reproduced hereafter.

```
     What is the name of the DATA file? (defaults are "... data a")
     lakes data a
     What is the name of the OUTPUT MATRIX file? (defaults are "... data a")
     lakes s15 a
     What is the name of the PARTIAL SIMILARITY MATRICES, if any?
     (defaults are "... data a")
(1)
     What is the name of the output PROBABility matrix file, if any?
     (defaults are "PROBAB data a")
(2)

      S I M I L : A program for computing resemblance matrices

      VERSION 3.0b
      AUTHOR: A. VAUDOR
      REFERENCE: Chapter 6 of
      Legendre, L. and P. Legendre. 1983 -- Numerical ecology.
          Developments in Environmental Modelling, 3. Elsevier
          Scientific Publ. Co., Amsterdam. xvi + 419 p.

      TITLE:
     Physical and chemical data, 71 lakes.
      NUMBER OF OBJECTS (ROWS)?
     71
```

```
 NUMBER OF DESCRIPTORS (COLUMNS)?
11
 CODE FOR MISSING VALUES?
-9
 ARE THE FIRST 10 COLUMNS USED TO WRITE THE OBJECT NAME? (y or n)
y
 WHAT RESEMBLANCE FUNCTION DO YOU WANT TO COMPUTE ?
 Similarities  : s01 to s26, Nei
 Distances     : d01 to d14
 R-mode:    rp = Pearson's r
            rs = Spearman's r
            tau= Kendall's tau
            chi= Chi-square (G statistic)
            ht = Tschuproff's contingency coefficient
            hs0= Reciprocal information        S=B/(A+B+C)
            hs1= Rajski's coherence     S'=SQRT(1-(hd)**2)
            hs2= Symmetric uncertainty coeff.S"=B/(A+2B+C)
            hd = Rajski's metric          D =(A+C)/(A+B+C)

s15
 SHOULD THE VARIABLE RANGES BE COMPUTED FROM THE DATA SET? (y or n)
y
 SHOULD ALL WEIGHTS W[i] BE SIMPLY 0 OR 1 ? (y or n)
y
 How many multi-state QUALITATIVE descriptors are there?
0
 End of the program.
```

Contents of the file of results

The output file containing the results of the calculations is written in binary form; so, it is not possible to read it directly using an ASCII editor or a word processor. The same applies to the file of probabilities associated with some of the coefficients. These files may be examined through the LOOK program, that can transcribe them into readable (ASCII) characters.

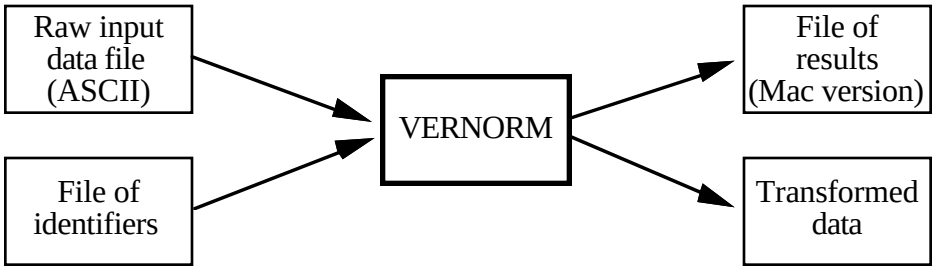
VERNORM

What does VERNORM do ?

VERNORM is a multiple-purpose program designed to be used at the beginning of the analysis of a data table. It has been created to answer general needs concerning input data files. Its name means VERify and NORMalize; it can transpose a matrix, reformat the data, add or remove object identifiers (although, on microcomputers, that function is more easily carried out by spreadsheet programs), draw histograms, divide variables into classes, standardize data, or transform them in various ways. VERNORM can handle missing values. Eliminating rows or columns from a data file can be done by spreadsheet programs, or through the editors of various types available on microcomputers; statistical analysis programs can be used to transform the data in ways not available in VERNORM.

To run an operation, simply choose the corresponding option in the menu, and answer the questions of the program. Leave it to the program to guide you.

Input and output files



(1) Raw input data file

That file, written in readable characters (EBCDIC on IBM mainframes, or ASCII), presents itself as a $p \times n$ data table where the rows are usually the p objects and the columns are the n variables. It may contain identifiers in the first 10 columns, if so desired; its general format is presented in section “Main input data file” of program SIMIL. The numbers may have been typed in using an editor, or extracted from a data base (“text only” format). It may also be the result of computations made by a statistical analysis program (SPSS, SAS, STATVIEW, etc.) or by some other program capable of writing results in “text” (EBCDIC or ASCII) form.

(2) File of identifiers

If the data table does not contain row identifiers, or else if it has been transposed, names can be added to the rows by providing VERNORM with a file containing a list of row identifiers. In the CMS and VMS versions, that file is called “TITLE” by default and presents itself as a list of object names written in a file in readable form (EBCDIC or ASCII).

Warning: In the Macintosh version, the identifiers must have at least 10 characters, including blanks and tabs; the first 10 characters of each line will be used. If there are fewer than 10 characters, unless one uses blanks to fill the line to 10 characters, the program will complete to 10 using characters from the next line of the file of identifiers, so that identifiers will be missing at the end of the list. In the CMS and VMS versions, one only has to write the identifiers as the successive lines of the file; the first 10 characters of each line will be used. Here is an example of a file of kangaroo species names:

```
Setonix b.
Thylog. c.
Petrog. g.
Wallab. v.
Macrop. w.
```

(3) File of transformed data

After transformation, the data can be written onto a new file if the user so requires. That file does not exceed 80 characters in width and is in a format appropriate to be used as input by the SIMIL program.

(4) File of statistical results

In the Macintosh version, results of the VERNORM computations are presented in a file containing statistical information. The content, which varies depending on the operations requested by the user, is described in more detail in the section "Contents of the file of results". In the mainframe version, this same information is presented on the screen (see example); it can be preserved in a console memory file, following the instructions on page 2 of the present manual.

Options of the program

This program offers a wide variety of options. The preliminary computation options allow to verify the data, to transpose the input matrix or to make all data positive. The main options, numbered 0 to 8 in the CMS and VMS versions, allow to test the data for normality, to transform them in various ways, and to plot histograms. Details on the use of these options, described below, are found in the next section ("Questions of the program"). Let us look at the options one by one.

(1) Verifying the input data file

This option allows to make sure that the data table is complete, and that it has the correct number of rows and columns. If there are fewer data than declared by the user, or if there are figures that are not separated by spaces, or containing "illegal" characters, an error message is produced and the program stops. The verification option also finds the global bounds of the values in the table (minimum, maximum), as well as the bounds and number of values per data line. All this information (see the example) may be useful to detect problems.

(2) Transposing the data matrix

This option offers the possibility of transposing a $p \times n$ input matrix into an $n \times p$ matrix, or vice versa. In this way, the same data file gives access to both Q- and R-mode analyses, by transposing the initial data matrix. By calling in a "File of identifiers" (see above), the rows of the transposed matrix can be labeled with identifiers (10 characters) when the data are rewritten.

(3) Making the data positive

This option allows to eliminate null or negative values from the file, by adding a constant to all data; the constant may be a different one for each column, or the same value across the whole file. That translation is necessary when the user wants to use either the Taylor, the Box-Cox, or the Box-Cox-Bartlett transformations, because they require computing logarithms of the data; that transformation may also be useful as a preliminary calculation before computing some of the similarity coefficients.

(4) OPTION 0: Taylor transformation

The first purpose of this transformation is to homogenize the variances of the variables in the input data matrix. It is necessary to have made all data strictly positive before using this transformation, because it requires computing logarithms of the data. When the data set contains several groups of objects (written down in the various columns), or when analyzing a group of dimensionally homogeneous quantitative descriptors (ex.: species abundances) to which one wants to apply a single transformation, Taylor's power law provides a general transformation that tends to homogenize the variances. The data are then more likely to conform to the conditions required by parametric statistical analyses, including normality. If one draws a graph of the variances against the means, Taylor's power law relates the means to the variances through the equation

$$\mathrm{Var}(y) = a\,(\mathrm{Moy}(y))^b$$

which allows to compute the values of parameters *a* and *b* through non-linear regression. An approximation may also be computed by linear regression (model I or model II) applied to the logarithmic form of the equation. VERNORM offers the following options to compute that regression:

- Model I: simple linear regression.
- Model II: reduced major axis regression.
- Model II: Bartlett's three-group method.
- Model II: major axis regression.
- Non-linear regression.

Another possibility is to ask the program to compute all the solutions listed above. Differences among these methods are explained in several textbooks of statistics, including Sokal & Rohlf (1981).

(5) OPTION 1: Box and Cox transformation

This option allows to individually normalize the variables in the input data matrix. The Box-Cox method empirically determines the best exponent of the following general transformation function, that produces the distribution which is closer to normality:

$y' = (y^{lambda} - 1)/lambda$ if $lambda \neq 0$
and $y' = \ln(y)$ if $lambda = 0$.

The value of *lambda* is found by iteratively maximizing a likelihood function (Sokal & Rohlf, 1981: 423). All values of *y* must be strictly positive because the likelihood function uses the logarithms of the data.

When *lambda* equals 1, the function would produce a simple linear transformation; in practice, no transformation is made in that case. If *lambda* is equal to 0.5, the function produces the square root transformation; when *lambda* is equal to 0, the transformation is log; finally, when *lambda* is equal to -1, the inverse transformation is obtained. This method, which is quite good to reduce the asymmetry of a data distribution, can never pretend to normalize a multi-modal distribution.

(6) OPTION 2: Box-Cox-Bartlett transformation

This option normalizes the variables and homogenizes their variances at the same time. In this variant, Bartlett's χ^2 statistic for homogeneity of the variances is used in the maximum likelihood equation of the Box-Cox method (Sokal & Rohlf, 1981: 425); it produces a single transformation for all the variables in the data file, which homogenizes the variances as much as possible, while normalizing the distributions. Like the Taylor transformation, this option may be used when a whole group of quantitative variables must be transformed using the same transformation. All the values of

y must be strictly positive because the likelihood function uses logarithms of the data.

(7) OPTION 3: Division into classes

This option offers the user with the possibility of dividing the variables of the input file into classes. One may choose to divide all the variables into classes, or some of them only. Each variable may be divided into the same or a different number of classes. VERNORM proposes to use a number of classes *k* which is a function of the number of observations *p*, following Sturge's rule: $k = 1 + (3.3 \log_{10} p)$ with rounding of *k* to the nearest integer value.

(8) OPTION 4: Your choice of transformation

This option allows to choose among four families of transformations:

1) $y' = a + by$
2) $y' = y^a$
3) $y' = \exp(y)$
4) $y' = \ln(a + by)$

The user has to provide the values of constants *a* and *b*, as applicable. All variables in the file may be subjected to the selected transformation, or some variables only. Notice that on microcomputers, it may often be easier to compute transformations of this kind using spreadsheet programs (ex. EXCEL) or statistical packages (ex. STATVIEW).

(9) OPTION 5: Histograms

This option draws frequency histograms for all variables. In this way, the frequency distributions of the descriptors can be visually examined before a transformation function is chosen. In the CMS and VMS versions, the histograms are represented laterally on the screen (see example), while in the Macintosh version, histograms are drawn in the usual way (see "Contents of the file of results"). The number of classes is determined by the user; VERNORM proposes a number of classes *k* which is a function of the number of observations *p*, following Sturge's rule: $k = 1 + (3.3 \log_{10} p)$ with rounding of *k* to the nearest integer value.

(10) OPTION 6: Standardization

This option allows to standardize (transformation into "*z*-scores") the variables chosen by the user. If that transformation is used after normalizing the data, standard normal distributions are obtained.

(11) OPTION 7: Tests of normality

This option computes the Kolmogorov-Smirnov test of normality, with reference to the table of critical values proposed by Lilliefors (1967); that table takes into account the fact that the mean and variance of the population remain unknown by hypothesis, and are estimated from the data themselves. The Kolmogorov-Smirnov test is preferable for example to the chi-square test, which does not take into account the ordered nature of the data. The test is computed for all variables in the input file. Results are displayed for the significance level chosen by the user; remember that a lower significance level (for instance 1%) is more permissive in terms of distribution of the data, because it is then more difficult to reject the null hypothesis of normality.

(12) OPTION 8: Rewriting the transformed data file

This option is used after the variables have been transformed, or after transposition of the

matrix, to rewrite the transformed data into a new file; the data include all the transformations performed up to that point. Questions are asked by the program to determine the output format of the new file. It is possible at this step to recode the missing values, to impose a scale to the data by fixing their minimum and maximum, and to include row identifiers provided in a file of identifiers.

Questions of the program

The questions displayed on the Macintosh screen are described in the following paragraphs. The questions of the CMS and VMS versions are essentially the same, as can be verified in the example of the next section. To start the program, click on the icon, then on "Open" in the "File" menu.

(1) "Data file" — The program presents a list of the available ASCII files.

(2) "File of statistical results" — The program presents a menu allowing to give a name to the file of statistical results computed by the program. That question is not produced by the CMS and VMS versions because the statistical results are displayed on the screen only.

(3) "Row identifiers present (10 first characters)? [Yes, No] — The answer is *Yes* if the first 10 columns of each object-vector or descriptor-vector contain a row identifier.

(4) "Number of rows (lines or blocks of lines)" — The answer to this question must be a positive integer number. In the case of a matrix of p rows (objects) x n columns (variables), the user gives here the number of objects, each object possibly occupying one or several lines in the data file; if the matrix has been transposed, it is the number of variables which has to be given here, each variable possibly occupying again one or several lines of the data file; refer to the description of the raw input data file.

(5) "Number of columns" — In the case of a matrix of p rows (objects) x n columns (variables), the answer to this question is the number of variables describing each object, to the exclusion of the row identifiers, if present in the file. If the matrix has been transposed, the number of objects composing each variable-vector is given, to the exclusion of the descriptor identifiers, if present in the file.

(6) "Code for missing values" — The numerical value is given which has been used in the file to indicate that an information is missing (-1, -9, -999, etc. are often used codes). That question has to be answered by a numerical value, even when there are no missing values in the file.

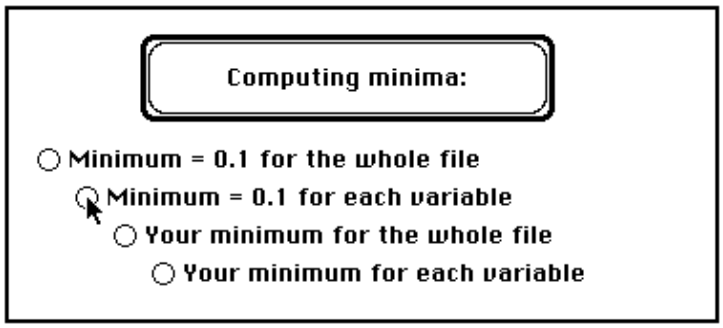
(7) "Verification of input file?" [Yes, No] — See the description of this function in paragraph (1) of the options. If the answer is *Yes*, the program asks for additional information about the input data file.

(7.1) "Input file containing only integer numbers?" [Yes, No] — Depending on the numerical nature of the data, integer or real, the program uses different procedures to read the data. After that question, the program (Macintosh as well as mainframe) lists on the screen how many values are found on each line of the input data file, as well as the minimum and maximum values per line; see the example. At the end of the list, the program indicates the minimum and maximum values in the data file; click the mouse to go to the next question.

(8) "Transposition of data matrix?" [Yes, No] — Answer *Yes* to indicate that the data matrix should be transposed. The row identifiers are lost during transposition, since the rows have become the columns. A new series of names, provided in a file of identifiers, may be added at the beginning of the new rows if the user so wishes.

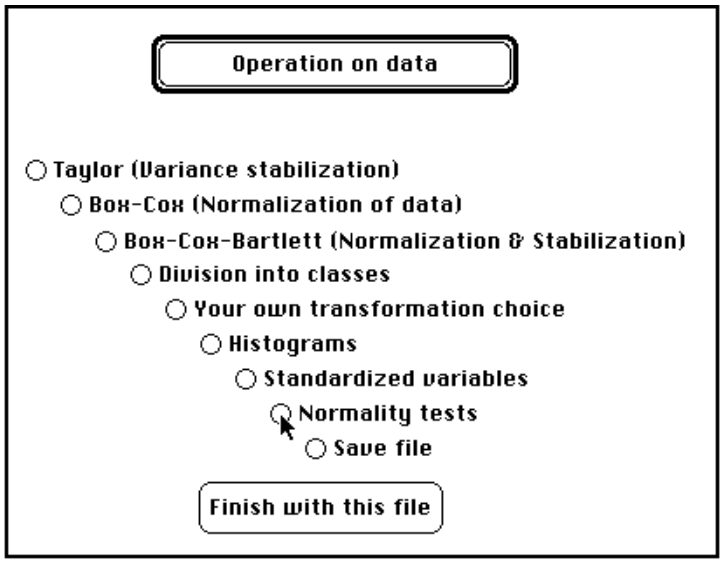
(9) "Make data positive (particularly for Taylor, Box-Cox, Box-Cox-Bartlett) ?" [Yes, No] — See the description of this function in paragraph (3) of the options. If the answer is *Yes*, the program offers the following options on the screen:

(9.1)



As can be seen, one may decide to impose the minimum value of one's choice (a different value for each variable, or else the same minimum value throughout the data matrix), or ask the program to impose 0.1 as the minimum (separately for each variable, or as if the whole data matrix contained a single variable).

(10) "Operation on data" — The screen allowing the user to choose an operation is the following:

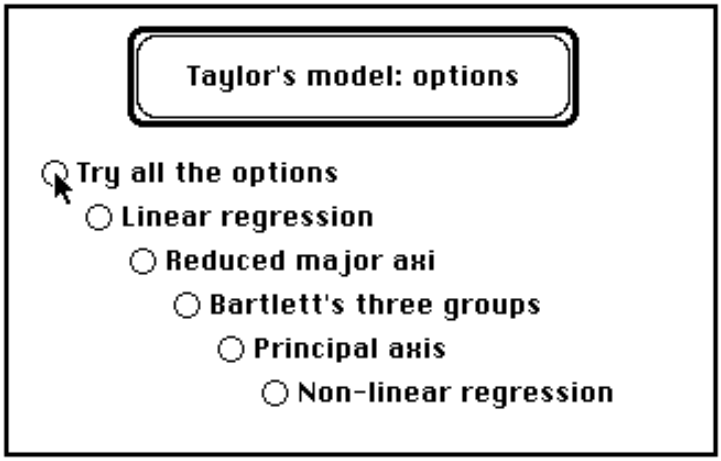


In the CMS and VMS versions, the options offered in this screen are divided in two questions. First, "Do you wish to do something with the data file? (Y or N)?". If the answer is *No*, the program stops immediately; if the answer is *Yes*, a menu with the 9 options shown above is presented; see the example.

From this point and on, the questions differ depending on the option. After each option is completed, the menu shown above is presented again. To stop the cycle, click on button "Finish with this file"; it is then possible to go back to the "File" menu and work on another input data matrix. Command "Interrupt" in the "R: Vernorm" menu stops the program.

Button: Taylor (Variance stabilization) — See paragraph (4) of the options

(11) "Taylor's model: options" — The program presents the user with the following screen:



Button: Box-Cox (Normalization of data) — See paragraph (5) of the options

(11) "How many variables to transform?" — Type the number of variables that will be subjected to the Box & Cox transformation. If the user asks to transform all variables, no supplementary question will be asked. When this is not the case, the program asks:

(11.1) "Number of variable [1]" — If for instance the fifth variable in the file is the first one to be transformed, type "5". Next the program asks: "Number of variable [2]", etc. until the number of variables to be transformed is accounted for.

Button: Box-Cox-Bartlett (Normalization & Stabilization) — Paragraph (6) of the options

No additional question is presented by the program. A single transformation is computed and applied to all variables. That transformation is not the same as would have been obtained if the user had pretended that the whole data file forms a single variable and had requested a Box-Cox transformation (above, and paragraph 5 of the options).

Button: Division into classes — See paragraph (7) of the options

(11) "How many variables to transform?" — Type the number of variables that will be divided into classes. If the user does not ask to transform all variables, the program asks:

(11.1) "Number of variable [1]" — If for instance the fifth variable in the file is the first one to be transformed, type "5". Next the program asks: "Number of variable [2]", etc. until the number of variables to be transformed is accounted for.

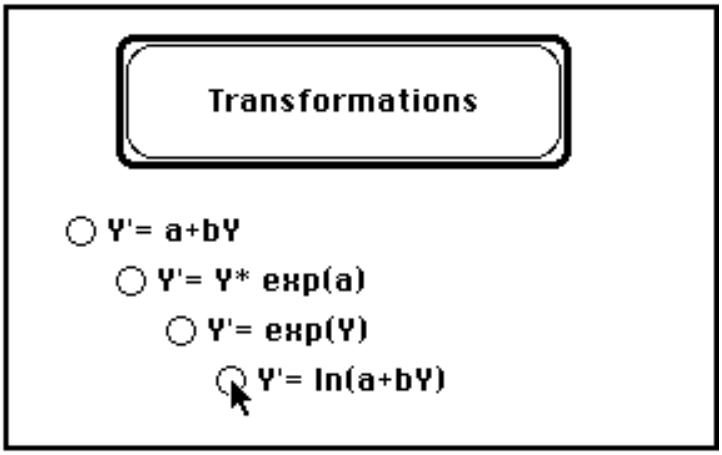
(12) "Same number of classes for all variables?" [Yes, No] — Answer *No* to indicate that each variable should be divided into a different number of classes.

(12.1) If the answer to (12) was *Yes*, the program asks the follow-up question: "Number of classes? (Sturge's number = *k*)" — Type the number of classes.

(12.2) If the answer to (12) was *No*, the program asks the follow-up question: "Number of classes for variable [1]? (Sturge's number = *k*)" — Type the number of classes in which to divide variable 1. The question is repeated for all the other variables identified in (11.1).

Button: Your own transformation choice — See paragraph (8) of the options

(11) "Transformations" — The program presents the user with the following screen:



(12) "Value of a" and/or "Value of b" — Depending on the answer to (11), the program needs values for the parameters of the selected transformation. To obtain the classical species abundance transformation $y' = \ln(y + 1)$, for instance, click the last button and give the following values for the parameters: $a = 1$, $b = 1$.

(13) "How many variables to transform?" — Type the number of variables that will be subjected to the selected transformation. If the user does not ask to transform all variables, the program asks:

(13.1) "Number of variable [1]" — If for instance the fifth variable in the file is the first one to be transformed, type "5". Next the program asks: "Number of variable [2]", etc. until the number of variables to be transformed is accounted for.

Button: Histograms — See paragraph (9) of the options

(11) "Number of classes? (Sturge's number = k) — Indicate the number of classes to be used. The histograms are presented on the screen; one must "Finish" a graph in menu "Histogram" to obtain the next one. The pictures may also be printed, or saved in a PICT file for future use.

Button: Standardized variables — See paragraph (10) of the options

(11) "How many variables to transform?" — Type the number of variables that will be subjected to that transformation. If the user does not ask to transform all variables, the program asks:

(11.1) "Number of variable [1]" — If for instance the fifth variable in the file is the first one to be transformed, type "5". Next the program asks: "Number of variable [2]", etc. until the number of variables to be transformed is accounted for.

Button: Normality tests — See paragraph (11) of the options

(11) "Tests of normality of Kolmogorov-Smirnov-Lilliefors. Significance level:" [1%, 5%, 10%, 15%, 20%] — Click on the button corresponding to the selected significance level.

Button: Save file — See paragraph (12) of the options

(11) "Number of characters in which numbers will be written" — Type a number, indicating how wide, in number of characters, the field devoted to each variable will be. See example.

(12) "How many digits after decimal point" — Type the number of decimal places each variable will have. See example.

(13) “Replace missing values by” — Type the code that will be used in the new file for missing values. The new code may be the same, or not, as the one used in the input file. An answer must be given, in the form of a numerical value, even if there are no missing values in the data file.

(14) “Do you wish to set min & max of output file?” [Yes, No] — *No* indicates that the user does not wish to impose a scale to the data, by fixing minimum and maximum values. If the answer is *Yes*, follow-up questions are presented:

(14.1) “Minimum value” — The value given here is used as the minimum over the whole data file.

(14.2) “Maximum value” — The value given here is used as the maximum over the whole data file.

(15) “Do you have a file of row identifiers?” [Yes, No] — Answer *Yes* if a file of row identifiers is available; in that case, the program presents the menu of the available ASCII files. Row identifiers are transcribed in the 10 first columns of each row of the data table (corresponding to a single line, or to a block of lines of the file).

Example

Following is an example of the use of VERNORM on mainframes. The input data file contains 60 objects and 3 descriptors. Even though there are no missing values in that file, an answer must be given to the question concerning missing values (the user’s answers are underscored and in boldface). The program is asked first to compute Kolmogorov-Smirnov tests of normality on the raw data, and to draw histograms; the next request is to find the best normalizing transformation in the sense of Box & Cox, followed by new tests of normality and histograms of the transformed variables. Finally, the user asks to rewrite the data into a format of 10 characters with 5 decimal places (Fortran format 3F10.5). The dialogue, produced under CMS, is reproduced below. The input data file is the same as in the “Contents of the file of results” section below, where it is analyzed on Macintosh.

```
Vernorm
What is the name of the DATA file ? (Defaults are "... data a")
60x3  data  a

What do you wish the TRANSFORMED DATA file to be called ?
(Defaults are '... data a')
60x3  transfor  a

What is the name of your OBJECT NAMES file, if any ?
(Defaults are "TITLE data a")


Execution begins...

 P R O G R A M   V E R N O R M  to VERify and NORMalize data

 VERSION 3.0b
 AUTHOR: A. VAUDOR.
 ARE THE FIRST 10 COLUMNS USED TO WRITE THE OBJECT NAME? (y or n)
n
 NUMBER OF OBJECTS (ROWS)?
60
 NUMBER OF DESCRIPTORS (COLUMNS)?
3
```

```
 CODE FOR MISSING VALUES?
-999
 DO YOU WISH TO CHECK THE DATA FILE?
y
 DOES THE DATA FILE CONTAIN ONLY INTEGER NUMBERS?
n
  LINE    N. OF VALUES       MIN         MAX
     1             3        3.08       48.70
     2             3        2.84       48.20
     3             3        3.12       49.00
     4             3        3.37       48.40
  [etc.]
    59             3        2.90       42.20
    60             3        0.86       42.20

 SMALLEST VALUE IN THE DATA FILE:    0.23
 LARGEST  VALUE IN THE DATA FILE:   50.10

 DO YOU WISH TO TRANSPOSE THE DATA FILE?
n
 DO YOU WISH TO MAKE ALL DATA POSITIVE?
(THIS IS REQUIRED FOR TAYLOR, BOX-COX AND BOX-COX-BARTLETT TRANSF.)
n
 DO YOU WISH TO DO SOMETHING WITH THE DATA FILE? (y or n)?
y
  OPTIONS
    0: TAYLOR (to homogenize variances)
    1: BOX-COX (to normalize data)
    2: BOX-COX-BARTLETT (to normalize data AND homogenize variances)
    3: DIVISION INTO CLASSES
    4: YOUR CHOICE OF A TRANSFORMATION
    5: HISTOGRAMS
    6: STANDARDIZE DATA (Z-scores)
    7: TESTING NORMALITY: Kolmogorov-Smirnov-Lilliefors
    8: RE-WRITE THE DATA FILE
7
 LEVEL OF SIGNIFICANCE REQUESTED: TYPE
     1 = 1 %, 2 = 5 %, 3 = 10 %, 4 = 15 %, 5 = 20 %
2

 KOLMOGOROV-SMIRNOV TESTS (LILLIEFORS TABLE)

 HYPOTHESIS: R=REJECTED, NR=NOT REJECTED, NC=NOT COMPUTABLE

 VARIABLE  :    1      2      3
 DISTANCE  : 0.1667 0.2629 0.0821
 CRIT. VAL.: 0.1144 0.1144 0.1144
 HYPOTHESIS:     R      R     NR

 DO YOU WISH TO DO SOMETHING WITH THE DATA FILE? (y or n)?
y
  OPTIONS
    0: TAYLOR (to homogenize variances)
    1: BOX-COX (to normalize data)
```

```
    2: BOX-COX-BARTLETT (to normalize data AND homogenize variances)
    3: DIVISION INTO CLASSES
    4: YOUR CHOICE OF A TRANSFORMATION
    5: HISTOGRAMS
    6: STANDARDIZE DATA (Z-scores)
    7: TESTING NORMALITY: Kolmogorov-Smirnov-Lilliefors
    8: RE-WRITE THE DATA FILE
5

 HISTOGRAMS

 HOW MANY CLASSES? (STURGE'S RULE SUGGESTS 7 CLASSES; MAX = 60 CLASSES)
7

 VARIABLE :  1
 |*************
 |************************
 |************
 |****
 |**
 |***
 |*
 ------------------------------------------------------------------------
                          25

 VARIABLE :  2
 |**
 |**************
 |****
 |
 |*
 |********************
 |*******************
 ------------------------------------------------------------------------
                    20

 VARIABLE :  3
 |***
 |*****
 |*************
 |******************
 |****************
 |**
 |**
 ------------------------------------------------------------------------
                   19

 DO YOU WISH TO DO SOMETHING WITH THE DATA FILE? (y or n)?
y
  OPTIONS
    0: TAYLOR (to homogenize variances)
    1: BOX-COX (to normalize data)
    2: BOX-COX-BARTLETT (to normalize data AND homogenize variances)
    3: DIVISION INTO CLASSES
```

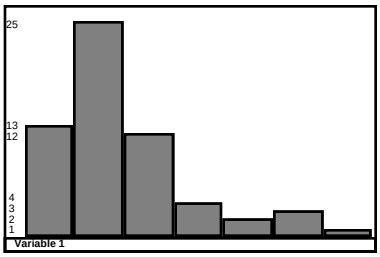
```
    4: YOUR CHOICE OF A TRANSFORMATION
    5: HISTOGRAMS
    6: STANDARDIZE DATA (Z-scores)
    7: TESTING NORMALITY: Kolmogorov-Smirnov-Lilliefors
    8: RE-WRITE THE DATA FILE
1
 HOW MANY VARIABLES DO YOU WISH TO TRANSFORM ?
3

 BOX-COX TRANSFORMATION

    VARIABLE    1
    LIMIT OF C.I. OF LAMBDA  -6.31388
                    LAMBDA  -3.96688
    LIMIT OF C.I. OF LAMBDA  -1.78488
    VARIABLE    2
    LIMIT OF C.I. OF LAMBDA   3.62280
                    LAMBDA   8.77780
    LIMIT OF C.I. OF LAMBDA  14.08280
    VARIABLE    3
    LIMIT OF C.I. OF LAMBDA   0.59872
                    LAMBDA   1.03672
    LIMIT OF C.I. OF LAMBDA   1.52372

 DO YOU WISH TO DO SOMETHING WITH THE DATA FILE? (y or n)?
y
  OPTIONS
    0: TAYLOR (to homogenize variances)
    1: BOX-COX (to normalize data)
    2: BOX-COX-BARTLETT (to normalize data AND homogenize variances)
    3: DIVISION INTO CLASSES
    4: YOUR CHOICE OF A TRANSFORMATION
    5: HISTOGRAMS
    6: STANDARDIZE DATA (Z-scores)
    7: TESTING NORMALITY: Kolmogorov-Smirnov-Lilliefors
    8: RE-WRITE THE DATA FILE
8
 HOW MANY SPACES DO YOU ALLOW TO WRITE EACH VALUE?
10
 HOW MANY DECIMAL PLACES?
5
 CODE FOR MISSING VALUES:
-999
 DO YOU WISH TO SCALE ALL VALUES BETWEEN IMPOSED MIN. AND MAX.?
n
 HAVE YOU PREPARED A FILE OF OBJECT NAMES? (File "TITLE")?
n
 TRANSFORMATIONS WRITTEN TO OUTPUT FILE:
 0=no transformation, A=Box-Cox, B=division into classes
 C=your choice of transformation, D=data standardized
 AA0                                                  [Explanation: next section]

 INSUFFICIENT FIELD; NEW FORMAT: 3F22.5               [Explanation: next section]
 End of the program.
```

Contents of the file of results

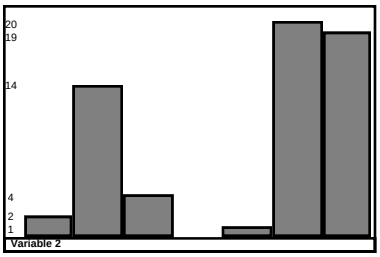
The file of statistical results produced by the Macintosh version contains different types of information, depending on the options of the program that have been selected. Here is an example of the content of such a file; comments are intermixed with results. The same information is presented on the screen in the mainframe versions of the program; see above.

(1) As in the example of the previous section (the data are the same), tests of normality have been requested first, on the raw data, using the 5% significance level. The corresponding histograms, which are displayed on the screen, have been sent to a PICT file and incorporated in the present page; the frequencies of the various columns are written in the left-hand part of the histogram, while the number of the variable is written at the bottom of each graph (although it is too small to be read on these highly reduced copies). The test rejects the hypothesis of normality for variables 1 and 2. The histograms show why: the distribution of variable 1 is highly skewed to the right but could probably be made symmetrical by transformation; variable 2 presents a bimodal distribution which cannot be normalized using a transformation of the type proposed in this program; finally, variable 3 is already recognized as normal by the K-S test and presents a unimodal, symmetrical distribution.

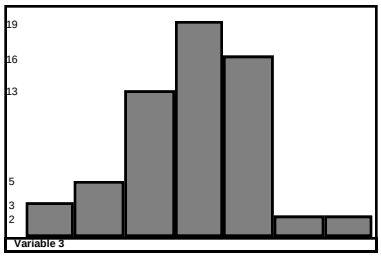
```
Kolmogorov-Smirnov-Lilliefors tests

 Hypothesis: R=rejected, Nr=not rejected, Nc=not computable

   Variable:    1      2      3
   Distance: 0.1667 0.2629 0.0821
   Crit.Val: 0.1144 0.1144 0.1144
 Hypothesis:      R      R     Nr
```



Variable 1



Variable 2



Variable 3

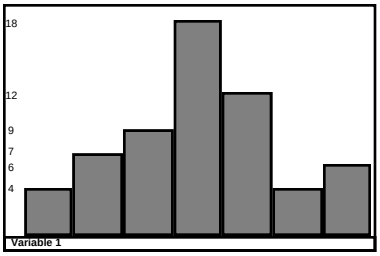
(2) Next the program is asked to look for the best normalizing transformation in the sense of Box & Cox. For each variable, the program presents the maximum likelihood value found for parameter *lambda*, with the limits of the 95% confidence interval ("limit lambda"). For the first variable, the value -3.96688 will be used as the exponent of the Box-Cox transformation; for variable 2, the value 8.77780 will be used, since value "1" is not within the limits of the 95% confidence interval of the parameter. In the case of the third variable, although the best value of the parameter found by the method is 1.03672, no transformation will be done because value "1" (no transformation) lies within the limits of the 95% confidence interval.

```
Box & Cox transformation

     Variable    1
       limit lambda  -6.31388
             lambda  -3.96688
       limit lambda  -1.78488
```

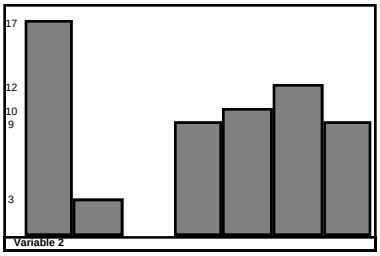
```
Variable    2
  limit lambda   3.62280
        lambda   8.77780
  limit lambda  14.08280
Variable    3
  limit lambda   0.59872
        lambda   1.03672
  limit lambda   1.52372
```

(3) Kolmogorov-Smirnov tests of normality are computed again. They show that the Box-Cox transformation has successfully transformed the first variable; this is confirmed by looking at the histogram. Transforming the second variable has not succeeded in reducing the bimodal character of that distribution. In the case of the third variable, it is easy to verify that the K-S test result as well as the histogram are identical with those of the previous page, no transformation having been done.

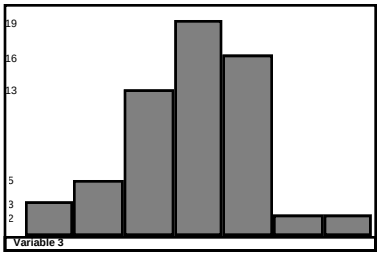
```
Kolmogorov-Smirnov-Lilliefors tests

 Hypothesis: R=rejected, Nr=not rejected, Nc=not computable

   Variable:    1      2      3
   Distance: 0.0700 0.2055 0.0821
   Crit.Val: 0.1144 0.1144 0.1144
 Hypothesis:     Nr      R     Nr
```



Variable 1



Variable 2



Variable 3

(4) The program is now asked to rewrite the transformed data into a new file. VERNORM provides the following information on the line flagged by an arrow (⇒): the first two variables have been subjected to a Box-Cox transformation (code a), while the third variable has not been transformed (code 0).

```
Transformations on output file
 0=No transformation, a=Box-Cox, b=Division into classes,
 c=Your transformation choice, d=Standardization
⇒  aa0
```

(5) The program has been asked to rewrite the data into a format of 10 characters with 5 decimal places (Fortran format 3F10.5). In the case of the second variable, however, the transformation selected by the Box-Cox method (bimodal distribution) generates gigantic numbers, which cannot be rewritten in 10 characters. In such a case, the program takes the liberty of imposing the most economical regular format capable of accommodating all data; that format requires 21 characters per variable, which includes at least one blank to prevent the numbers from touching (Fortran format 3F21.5).

```
Insufficient field, new format: 3f21.5
```

REFERENCES

Anderberg, M. R. 1973. Cluster analysis for applications. Academic Press, New York. xiii + 35p.

Blanc, F., P. Chardy, A. Laurec & J.-P. Reys. 1976. Choix des métriques qualitatives en analyse d'inertie. Implication en écologie marine benthique. Mar. Biol. (Berl.) 35: 49-67.

Burgman, M. 1987. An analysis of the distribution of plants on organic outcrops in southern Western Australia using Mantel tests. Vegetatio 71: 79-86.

Cailliez, F. & J.-P. Pagès. 1976. Introduction à l'analyse des données. Société de Mathématiques appliquées et de Sciences humaines, Paris. xxii + 616 p.

Cheetham, A. H. & J. E. Hazel. 1969. Binary (presence-absence) similarity coefficients. J. Paleontol. 43: 1130-1136.

Cliff, A. D. & J. K. Ord. 1981. Spatial processes: Models and applications. Pion Ltd., London.

Clifford, H. T. & W. Stephenson. 1975. An introduction to numerical classification. Academic Press, New York. xii + 229 p.

Cooper, D. W. 1968. The significance level in multiple tests made simultaneously. Heredity 23: 614-617.

Daget, J. 1976. Les modèles mathématiques en écologie. Collection d'Écologie, No 8. Masson, Paris. viii + 172 p.

Dirichlet, G. L. 1850. Über die Reduction der positiven quadratischen Formen mit drei unbestimmten ganzen Zahlen. Journal für die reine und angewandte Mathematik 40: 209-234.

Dow, M. M. & J. M. Cheverud. 1985. Comparison of distance matrices in studies of population structure and genetic microdifferenciation: quadratic assignment. Am. J. Phys. Anthropol. 68: 367-373.

Edgington, E. S. 1987. Randomization tests, 2nd ed. Marcel Dekker Inc., New York.

Estabrook, G. F. & D. J. Rogers. 1966. A general method of taxonomic description for a computed similarity measure. BioScience 16: 789-793.

Everitt, B. 1980. Cluster analysis, 2nd edition. Halsted Press, John Wiley & Sons, New York.

Frontier, S. 1976. Étude de la décroissance des valeurs propres dans une analyse en composantes principales: comparaison avec le modèle du bâton brisé. J. exp. mar. Biol. Ecol. 25: 67-75.

Gabriel, K. R. & R. R. Sokal. 1969. A new statistical approach to geographic variation analysis. Syst. Zool. 18: 259-278.

Galzin, R. & P. Legendre. 1987. The fish communities of a coral reef transect. Pacific Science 41: 158-165.

Gower, J. C. 1966. Some distance properties of latent root and vector methods used in multivariate analysis. Biometrika 53: 325-338.

Gower, J. C. 1982. Euclidean distance geometry. Math. Scientist 7: 1-14.

Gower, J. C. 1983. Comparing classifications. Pp. 137-155 *in*: Felsenstein, J. [ed.] Numerical taxonomy. NATO ASI Series, Vol. G 1. Springer-Verlag, Berlin. x + 644 p.

Gower, J. C. 1985. Measures of similarity, dissimilarity, and distance. Pp. 397-405 *in*: Kotz, S. & N. L. Johnson [eds.] Encyclopedia of Statistical Sciences, Vol. 5. Wiley, New York.

Gower, J. C. & P. Legendre. 1986. Metric and Euclidean properties of dissimilarity coefficients. Journal of Classification 3: 5-48.

Harris, C. W. & H. F. Kaiser. 1964. Oblique factor analytic solutions by orthogonal transformations. Psychometrika 29: 347-362.

Hope, A. C. A. 1968. A simplified Monte Carlo significance test procedure. J. Roy. Stat. Soc. Ser. B 30: 582-598.

Hubert, L. J. 1985. Combinatorial data analysis: association and partial association. Psychometrika 50: 449-467.

Hubert, L. J., R. G. Golledge & C. M. Constanzo. 1982. Analysis of variance procedures based on a proximity measure between subjects. Psychological Bull. 91: 424-430.

Hudon, C. & G. Lamarche. 1989. Niche segregation between American lobster *Homarus americanus* and rock crab *Cancer irroratus.* Mar. Ecol. Prog. Ser. 52: 155-168.

Isaaks, E. H. & R. M. Srivastava. 1989. An introduction to applied geostatistics. Oxford University Press, New York. xix + 561 p.

Jackson, D. A. & K. M. Somers. 1988. Are probability estimates from the permutation model of Mantel's test stable? Can. J. Zool. 67: 766-769.

Jain, A. K. & R. C. Dubes. 1988. Algorithms for clustering data. Prentice Hall, Englewood Cliffs, New Jersey. xiv + 320 p.

Kaiser, H. F. 1958. The varimax criterion for analytic rotation in factor analysis. Psychometrika 23: 187-200.

Lance, G. N. & W. T. Williams. 1966a. A generalized sorting strategy for computer classifications. Nature (Lond.) 212: 218.

Lance, G. N. & W. T. Williams. 1966b. Computer programs for hierarchical polythetic classification ("similarity analyses"). Computer Journal 9: 60-64.

Lance, G. N. & W. T. Williams. 1967. A generalized theory of classificatory sorting strategies. I. Hierarchical systems. Computer Journal 9: 373-380.

Legendre, L., M. Fréchette & P. Legendre. 1981. The contingency periodogram: A method of identifying rhythms in series of nonmetric ecological data. J. Ecol. 69: 965-979.

Legendre, L. & P. Legendre. 1983. Numerical ecology. Developments in environmental modelling, 3. Elsevier Scient. Publ. Co., Amsterdam. xvi + 419 p.

Legendre, L. & P. Legendre. 1984a. Ecologie numérique, 2ième éd. Tome 1: Le traitement multiple des données écologiques. Tome 2: La structure des données écologiques. Collection d'Écologie, 12 et 13. Masson, Paris et les Presses de l'Université du Québec. xv + 260 p., viii + 335 p.

Legendre, P. 1987. Constrained clustering. Pp. 289-307 *in*: P. Legendre & L. Legendre [eds.] Developments in numerical ecology. NATO ASI Series, Vol. G 14. Springer-Verlag, Berlin. xi + 585 p.

Legendre, P. & A. Chodorowski. 1977. A generalization of Jaccard's association coefficient for *Q* analysis of multi-state ecological data matrices. Ekol. Pol. 25: 297-308.

Legendre, P., S. Dallot & L. Legendre. 1985. Succession of species within a community: chronological clustering, with applications to marine and freshwater zooplankton. Amer. Nat. 125: 257-288.

Legendre, P. & M.-J. Fortin. 1989. Spatial pattern and ecological analysis. Vegetatio 80: 107-138.

Legendre, P. & V. Legendre. 1984b. Postglacial dispersal of freshwater fishes in the Québec peninsula. Can J. Fish. Aquat. Sci. 41: 1781-1802.

Legendre, P., N. L. Oden, R. R. Sokal, A. Vaudor & J. Kim. 1990. Approximate analysis of variance of spatially autocorrelated regional data. J. Class. 7: 53-75.

Legendre, P. & M. Troussellier. 1988. Aquatic heterotrophic bacteria: Modeling in the presence of spatial autocorrelation. Limnol. Oceanogr. 33: 1055-1067.

MacQueen, J. 1967. Some methods for classification and analysis of multivariate observations. Pp. 281-297 *in*: L. M. Le Cam & J. Neyman [eds.] Proceedings of the fifth Berkeley symposium on mathematical statistics and probability, Vol. 1. University of California Press, Berkeley. xvii + 666 p.

Mantel, N. 1967. The detection of disease clustering and a generalized regression approach. Cancer Res. 27: 209-220.

McCune, B. & T. F. H. Allen. 1985. Will similar forest develop on similar sites? Can. J. Bot. 63: 367-376.

Mielke, P. W. 1978. Clarification and appropriate inferences for Mantel and Valand's nonparametric multivariate analysis technique. Biometrics 34: 277-282.

Miles, R. E. 1970. On the homogeneous planar Poisson point process. Math. Biosci. 6: 85-127.

Miller Jr., R. G. 1977. Developments in multiple comparisons. J. Amer. Stat. Ass. 72: 779-788.

Oden, N. L. 1984. Assessing the significance of spatial correlograms. Geogr. Anal. 16: 1-16.

Oden, N. L. & R. R. Sokal. 1986. Directional autocorrelation: An extension of spatial correlograms to two dimensions. Syst. Zool. 35: 608-617.

Oden, N. L. & R. R. Sokal. Investigation of 3-matrix quadratic assignment tests. (Submitted).

Orlóci, L. 1978. Multivariate analysis in vegetation research. 2nd ed. Dr. W. Junk B. V., The Hague. ix + 451 p.

Ripley, B. D. 1981. Spatial statistics. John Wiley & Sons, New York.

Rohlf, F. J., J. Kishpaugh & D. Kirk. 1971. NT-SYS. Numerical taxonomy system of multivariate statistical programs. Tech. Rep. State University of New York at Stony Brook, New York.

SAS. 1985. SAS user's guide: statistics. SAS Institute Inc., Cary, North Carolina.

Smouse, P. E., J. C. Long & R. R. Sokal. 1986. Multiple regression and correlation extensions of the Mantel test of matrix correspondence. Syst. Zool. 35: 627-632.

Sneath, P. H. A. 1966. A comparison of different clustering methods as applied to randomly-spaced points. Classification Society Bulletin 1: 2-18.

Sneath, P. H. A. & R. R. Sokal. 1973. Numerical taxonomy — The principles and practice of numerical classification. W. H. Freeman, San Francisco. xv + 573 p.

Sokal, R. R. 1986. Spatial data analysis and historical processes. Pp. 29-43 *in*: Diday, E. *et al.* [eds.] Data analysis and informatics, IV. Proc. Fourth Int. Symp. Data Anal. Informatics, Versailles, France, 1985. North-Holland, Amsterdam.

Sokal, R. R., I. A. Lengyel, P. A. Derish, M. C. Wooten & N. L. Oden. 1987. Spatial autocorrelation of ABO serotypes in mediaeval cemeteries as an indicator of ethnic and familial structure. J. Archaeol. Sci. 14: 615-633.

Sokal, R. R. & N. L. Oden. 1978. Spatial autocorrelation in biology. 1. Methodology. Biol. J. Linnean Soc. 10: 199-228.

Sokal, R. R. & F. J. Rohlf. 1962. The comparison of dendrograms by objective methods. Taxon 11: 33-40.

Sokal, R. R. & F. J. Rohlf. 1981. Biometry, 2nd ed. W. H. Freeman, San Francisco. xviii + 859 p.

Sokal, R. R. & P. H. A. Sneath. 1963. Principles of numerical taxonomy. W. H. Freeman, San Francisco. xvi + 359 p.

Späth, H. 1980. Cluster analysis algorithms. Ellis Horwood, Chichester.

Thiessen, A. W. 1911. Precipitation averages for large areas. Monthly Weather Review 39: 1082-1084.

Upton, G. & B. Fingleton. 1985. Spatial data analysis by example. Vol. 1: Point pattern and quantitative data. John Wiley & Sons, Chichester. xi + 410 p.

Voronoï, G. F. 1909. Recherches sur les paralléloèdres primitifs. Journal für die reine und angewandte Mathematik 136: 67-179.

Ward, J. H. Jr. 1963. Hierarchical grouping to optimize an objective function. J. Amer. Stat. Ass. 58: 236-244.

Watson, D. F. 1981. Computing the n-dimensional Delaunay tesselation with application to Voronoi polygones. Computer J. 24: 167-172.

Williams, W. T. & M. B. Dale. 1965. Fundamental problems in numerical taxonomy. Adv. bot. Res. 2: 35-68.