WELCOME TO MAP-CODE!

MAP-CODE is an interactive program used to transform a series of morphological measurements or counts which are variable within populations, species, or taxa of any kind into discrete form which can then be used for phylogenetic or phenetic analysis using other programs. MAP-CODE is very easy and self explanatory to use as the user is queued successively for each decision and the results of each decision are displayed on the screen and may be modified indefinitely until stored on a disk. Due to its versatility and ease of use, the program may find use simply to investigate the nature of variability among groups for a set of characters.

The program is written in TURBO Pascal (version 3.0) for IBM-PC compatible machines and is fairly specific for that compiler and class of computers. It does not require a graphics card, although adding graphics capabilities is my next priority. Either a compiled version or the source code for the program has been supplied. In either case, the user should first make a copy of the distribution diskette from which a working copy of the compiled version of the program can be created.

For users with two floppy drives and no hard disk:

a) Place the distribution diskette in DRIVE A: and a blank, formatted diskette in DRIVE B:.

b) Type the following: copy a:*.* b: <RETURN>

If either the compiled or source version has been supplied:

a) Place the copy of the distribution diskette in DRIVE A: and a blank, formatted diskette in DRIVE B:.

b) Type the following: MAPCODE <RETURN>

c) Place a second, blank, formatted diskette in DRIVE B:

d) Type the following: QUIKCODE <RETURN>

For users with hard disks:

Copy all of the files on the distribution diskette to an appropriate place on the hard disk. For example, first make a new directory to house the MAPCODE files, change to that directory, and copy all of the files from the distribution diskette to the hard disk.

```bash
md \mapcode <RETURN>
cd \mapcode <RETURN>
copy a:*.* c: <RETURN>
```
For all users:

If the compiled versions of the program have been supplied you should be able to execute the program directly. See below for instructions on getting a simple demonstration using one of the supplied data sets.

If only the source code has been supplied you will have to create compiled (.COM) versions of the program with your TURBO compiler. This procedure is straightforward and is explained fully in the User’s Manual.

A copy of the USER’S MANUAL has been supplied and you should read through this carefully. There is an extensive section on creating data files and executing the program which should help you get going. There is also a WordStar version of the USER’S MANUAL supplied on the distribution diskette. Cursed or beloved, WordStar is available almost everywhere. In order to read this file with another word processing program you will have to strip off all of the control characters. Most word processing programs have procedures to do this.

In the USER’S MANUAL, I explain step-by-step how to use the TURBO Pascal compiler to create a compiled version of MAPCODE and QUIKCODE (a must for all those with only the source code supplied to them).

If you have trouble executing the program, you may have to recompile it for your system. If you don’t have the source code, you will have to get it from me (after all). It will not compile with any Pascal compiler except the TURBO Pascal compiler (version 3.0, and possibly earlier versions). It can’t be run with a BASIC interpreter, a C-compiler, or a FORTRAN compiler. TURBO Pascal is cheap (around $50 US through a discount house) and it has as large a distribution as virtually any other compiler.

For those of you who just can’t wait to try the program (and assuming you have the compiled version) first copy one of the data files, e.g., GROUPDAT.PAS onto your working MAPCODE disk. Place this disk with the compiled version of MAPCODE in DRIVE A and type MAPCODE <RETURN>. When asked:

1) the filename is: GROUPDAT <RETURN>
2) the type of data is: G(roup (type G), and
3) the measure of variability is: Standard Deviations (type 1 <RETURN>).

From here you can investigate the program either with or without the manual - but just don’t use the results for anything else! Please do read the manual and my Systematic Zoology paper (SZ 34:326-345) which describes all of the techniques implemented in the program.

Important Note: The current versions of MAPCODE and QUIKCODE will not
compile under version 4.0 of TURBO Pascal. Modifying the program to run under 4.0 and use graphics is one of my summer 1988 projects.

Good luck with the program. Please let me know if you have any problems with it (either major or minor).

James W. Archie
Department of Zoology
Honolulu, Hawaii 96822
(808)-948-6146 (office phone)
(808)-948-8617 (Dept. phone - secretar
MAP - CODE

A Program for Coding
Variable Character Information
for
Phylogenetic and Phenetic Analysis

Version 1.1
Written By
James W. Archie

Department of Zoology
University of Hawaii
Honolulu, HI 96822
USA

9/20/86
**Summarized Data Input Format**

This type of data is characterized by having the sample size and the character mean specified for each group along with some pooled measure of variability for the character among individuals within groups, e.g., the pooled within-group variance from an analysis of variance. The data are reported by character with the measurement or count information for each character for all groups separated sequentially.

LINE TYPE 1: Number of Groups, Number of Characters

LINE TYPE 2: Group Names (may be continued onto several lines)

LINE TYPE 3: A single pooled measure of within group variability for a character. One of 5 different measures of variability may be used, although these cannot be mixed within a data set. These are:

a. Pooled within-group standard deviation.
b. Pooled within-group variance.
c. Pooled within-group standard error (deviation) of the mean.
d. Two times the pooled within-group standard error of the mean.
e. Pooled within-group variance of the mean.

LINE TYPE 4: Sample sizes within groups on which the sample means for a character are based.

LINE TYPE 5: Sample means from each group for a single character.

Line types 3, 4, and 5 are repeated successively for each character.
2. Grouped Data - These data are the kind commonly reported in the literature or summarized by some statistical packages, where, although the raw data are not available, the sample sizes, means, and some measure of variability is provided for each character and within each group under study. Five different measures of variability may be used by the program (the same statistic must be used for all characters). These include:

a) the standard deviation or
b) variance of the individual characters,
c) the standard deviation of the mean (= standard error),
d) variance of the mean (square of c), and
e) twice the standard error of the mean.

3. Summarized Data - These data are also commonly reported in the literature or summarized by some statistical packages. For this type of data, however, only a single, pooled measure of within-group variability is available for each character, e.g. pooled, within-group mean square from an analysis of variance.

Sample input files of all three types are provided on the distribution diskette and may be used as models for creating your own data files as well as for experimenting with the program.

THE MAIN MENU

After the data have been successfully read by the program, the main menu will be displayed at the top of the screen. From this menu, all of the major options of the program are selected by depressing one of a variety of single keys on the keyboard. As you learn to use the program, these key strokes will become essentially automatic. Selecting one of the options will invoke either a new menu or complete a particular procedure. These are described below in detail.

SELECTING A CHARACTER TO CODE

A N(ew character must first be selected to undergo examination.*** After depressing N from the main menu, a list of previously coded (initially none) and uncoded characters is displayed and the user is asked to select one from the list of uncoded characters. Entering the value 0 will automatically select the next available character. An additional option is to select sE(quential from the main menu. This is simply a toggle which allows the user to select any uncoded character once N(ew is selected (the default) or has the program automatically select the next available character.

**Note: The designation N(ew used in the prompt at the top of the screen indicates that the user should depress the key N to select a new character. The other options from the main menu are M, C, D, S, E, and Q for M(ethod, C(ode, D(isplay, S tore, sE(quential, P(rint, and Q uit.
Once a character has been selected, some basic information about it is displayed on the screen. This includes the Character Number, the Range of Means among groups, and the pooled within-group Standard Deviation. In addition, the means are sorted from smallest to largest and the group numbers are displayed on the screen in sorted order.

SELECTING THE CHARACTER CODING METHOD

Next, one of a series of coding procedures or M(ethods) is selected to convert the information on variation among groups into a usable form. A new prompt menu will be displayed upon depressing M from the main menu.

A. Gap Coding--The classic procedure is that of Gap Coding, first introduced by Mickevich and Johnson (1976). Archie (1985) discussed the limitations of this procedure and, in particular, its sensitivity to sampling intensity.

B. Generalized Gap Coding--Archie (1985) also introduced a generalization of the gap coding procedure, the Generalized Gap Coding method, which retains the information on among group differentiation that is lost with simple gap coding.

C. Subset Coding--Simon (1983) introduced a method into the field of character coding which used univariate statistical data analysis techniques. She used multiple comparison procedures to code the information on differentiation among groups into a usable form. In the present program, this approach is known as S(ubset coding).

Two other experimental procedures are also included in the program, and may be most useful in simply investigating differentiation among groups for the different characters rather than as final coding procedures.

D. Group Coding--The first of these is called G(roup coding which considers groups of means as being identical if they do not differ from one another by at least some constant, k, times the pooled within-groups standard deviation (s), i.e., once the means are ordered by size, starting with the smallest mean, all means which do not differ by k * s are included in a single group. The coding procedure continues by selecting the next largest mean above the first group and comparing successively larger means until one differs by at least k * s, etc.

E. Range Coding--The final method is called R(ange coding which simply divides the range of means into some number, k, of intervals and assigns group means to the appropriate interval. Note that this is distinctly different than the more standard procedure of scaling the range of means to lie between 0 and 1.

F. Mean Coding--It is also possible to save the raw character means rather than using one of the above character coding
methods. The program does allow the raw character means to be
scaled by dividing by the pooled within group standard deviation
if desired. This latter procedure was recommended by Farris
(1968). The raw or scaled means cannot be displayed.

The name of the method selected is automatically displayed
on the table at the top of the screen. After selection of a
coding method, the user will be requested to enter various
parameter values to be used with the method. These are:

a) the gap size for either the simple or generalized gap
   procedures,

b) the alpha level used in the formation of homogeneous
   subsets,

c) the constant \( k \) used to form groups, and

d) the number of intervals into which the range is to be
divided.

Default values are initially assumed for each of these and
the user may select the default simply by depressing the <RETURN>
key. Alternatively, new values may be selected by entering them
on the keyboard. Once a new value is entered, it will be used as
the default the next time that same M(ethod) is selected. This
allows consistency between successive characters and also
increases the user execution speed of the program.

CODING THE DATA

Once the coding method and its parameters are selected, the
user must actually carry out the coding procedure. This is not
done automatically immediately after the selection of a method as
the user may decide to change his/her mind before carrying out a
particular procedure. Coding may take up to several seconds (but
usually is essentially instantaneous) depending on the method
selected and the number of taxa. A table lookup must be
performed with the S(ubset) method and requires access to the file
QTABLE.DTA on the distribution disk. If this method is to be
used repeatedly, the program can be speeded up substantially if
this file is transferred to a hard disk or a RAM disk. The
program looks for the file on the default drive, but if it cannot
be located, then it will ask the user for the drive code (A:, B:,
etc.) on which it can be found.

DISPLAYING THE RESULTS

The results of the coding operation can next be D(isplayed)
by depressing D from the main menu. The results will appear as a
series of horizontal bars which many users will recognize as
those often used to display homogeneous subsets in multiple
comparison procedures. If not all of the bars can be displayed
on the screen, the user will be cued to successively have some of
them eliminated from the display.
CHOOSING A CHARACTER ENCODING SCHEME

One of four character encoding schemes may now be selected from the Display Menu. The four methods are:

a) A(dditive binary coding, and
b) N(onadditive binary coding,
c) I(nteger coding, which assigns integer values to the different bar patterns, or
d) R(atio coding, which assigns values in increments of 0.5 to the bar patterns.

These encoding schemes are described and discussed by Simon (1983) and Archie (1985). The option selected will often depend on whether real number or only integer values are allowed in the program with which the data will be analyzed further, e.g., PAUP allows only integer values while NTSYS allows real numbers.

It may be the case that both the horizontal bars and the A(dditive or N(onadditive binary codes cannot all be displayed on the screen at once. The user may select whether the bars are to be eliminated to display the binary character codes or the binary characters are to be eliminated to maintain the display of the bars. The user is prompted to depress the <RETURN> key to eliminate successive bars or binary character lines.

PRINTING A COPY OF THE SCREEN

A new option added to versions 1.1 and later of MAPCODE is that of saving a copy of the main menu screen by printing it on a printer connected to COM1: or by saving it in a file. This might be desired as it is the only way that a hard copy of the sorted group numbers, sets of group bars, and encoding schemes can be saved. From the main menu, simply depress the P(rint option. You will then be asked whether you want to print the screen or save it in a file. If the latter option is chosen, a new file will be created in which to save a copy of the screen or, if the file already exists, the screen will be appended to the present contents of the file.

VERSATILITY OF THE PROGRAM

The user may sequentially redisplay the horizontal bars and a new encoding scheme by simply depressing D(isplay and the prompt of the encoding scheme desired. Alternatively, a new M(ethod may be selected, followed by the C(ode process, and the D(isplay and encoding of the character. A different character may be selected at any time from the main prompt menu or the main menu screen may be P(rined. The program has been written at this stage such that an illogical sequence of steps, which might normally crash the program, is not permitted. If such a sequence were attempted, the program chastises the user with small electric shocks - it actually just beeps belligerently at you and displays a message.
STORING THE CODED DATA FOR EACH CHARACTER

The results of the above procedures are saved separately for each character ONLY when the S(tore option is selected from the main prompt menu. THIS IS VERY IMPORTANT as otherwise, none of the results of the coding session will be saved. When the S(tore option is selected, the results are stored in two temporary data files and a summary of the coding procedures used for that character is displayed on the screen. The first file contains the coded characters themselves while the second file contains the summary of the coding procedures. These files will be used latter on when the coding session is completed to create a permanent file of new characters and to print a summary of the coding session. REMEMBER: Once you are through with the selection of a coding method and encoding procedure for each character, STORE THE RESULTS FOR THAT CHARACTER by depressing S for S(tore from the main menu.

Once a character has been stored, the user should select a N(eu character from the remaining uncoded characters and complete the subsequent processes outlined above.

QUITTING THE PROGRAM - CREATING A DATA FILE AND OUTPUT SUMMARY

After the set of characters that are to be coded have been S(tored, the user Q(uits the main prompt menu. He/She is then prompted for the name of the appropriate output file where the results of the coding session are to be permanently saved. The suffix .PAS is added to the name of the output file if not entered by the user. If the file already exists on the default drive, the user will be asked if the file should be overwritten. If not, the user may enter an alternative name for the output file. He/She is also prompted for the appropriate output format of the data. Available output formats include those for

a) PAUP,
b) PHYLIP,
c) NTYSYS-PC.

Other output formats will be added as requirements are made known to me. Finally, the user will be prompted to determine whether a summary of the coding session is to be printed or saved in an output file. This is important if the user has selected a variety of coding methods, parameter values, or encoding schemes for the original characters. The output summary is a copy of the information displayed on the screen after each character has been coded, encoded, and stored.
THE QUIKCODE PROGRAM

Two versions of the program are supplied on the distribution disk. One is the basic MAPCODE program that permits interactive analysis and coding of all characters as described in the previous pages. The second version is QUIKCODE which eliminates the interactive portions of the program but instead rapidly goes through all of the characters and codes them all using the same coding and encoding options. The latter are entered only once at the beginning of the program. As the program proceeds through its various operations, the user is informed of its progress with a screen display. This is an ideal way to investigate the overall affects on a phylogenetic or phenetic analysis due to different character coding and encoding procedures.
PROGRAM EXECUTION

To use the program the user should first make a copy of the necessary executable versions of the program files on a separate disk. This can be done with the following operations, assuming that the distribution diskette is located in DRIVE A: and a blank, formatted diskette is located in DRIVE B:

copy a:*.COM b:
copy a:*.000 b:
copy a:qtable.* b:

Next replace the distribution diskette in drive A: with the newly created diskette containing the executable versions of the program. A disk with the user’s data located on it, and formatted as described in this manual, should then be placed in DRIVE B: (to try the demonstration data files, place a copy of the distribution diskette or a disk with the relevant file.PAS file in DRIVE B:). Then, to execute the program the user should only have to type in the name of the program which he/she wants to execute, i.e., type either:

mapcode <RETURN>
or
quikcode <RETURN>

You may type these in with uppercase or lowercase letters. If the program does not work at this stage, one of several things may be wrong.

1. The program is not located on the default disk drive. Make sure all of the file.COM, file.000, and QTABLE.DTA files are located on a disk on the default disk drive.

2. The versions compiled on my system won’t run on your system. I don’t know why this would happen, but the only remedy to this is to recompile the program using the TURBO Pascal compiler. In other words, you will have to purchase or scrounge a version that will work on your machine. Once the program is recompiled and file.COM and file.000 files are created, you shouldn’t need the compiler any more except if you want to modify the program or use the TURBO editor to create data files.
COMPILING THE MAPCODE AND QUIKCODE PROGRAMS

The following operations need to be carried out:

a. Make sure all of the file.PAS and file.DTA files from the distribution diskette are located on a disk in DRIVE B:. Copy these from the distribution diskette to a new, formatted diskette. Don’t copy any .MAN files. Place a diskette with a version of the TURBO Pascal compiler in DRIVE A:.

b. Invoke the TURBO Pascal program - type TURBO <RETURN>.

c. Type Y to include the error messages.

d. Changed the logged disk drive - type L followed by B <RETURN>.

e. Reset the compiler options to create a .COM file - type O, then move the cursor to the .COM option with the arrow keys and press return. Then press Q to quit the options menu.

f. Set the work file to either MAPCODE.PAS or QUIKCODE.PAS - type W followed by the name of the file and <RETURN>.

g. Compile the program - type C. The TURBO compiler will compile the program selected and create MAPCODE.COM and MAPCODE.000 or QUIKCODE.COM and QUIKCODE.000 on your diskette. This operation will take several minutes, but you will be cued when its done. Quite a bit of disk access is required and this procedure would be much quicker if you could store all of the file.PAS files in a RAM disk or on a hard disk, although these aren’t necessary. If you do this on a RAM disk, however, remember to copy the files onto a floppy after compilation.

h. Change the work file and compile the other program if desired.

i. Exit TURBO - type Q.

j. Copy the compiled versions of the program(s) onto a new disk (all of the file.COM, file.000, and QTABLE.DTA files) and execute the program by typing in its name at the DOS prompt.

Technical Points

MAP-CODE has been written in TURBO Pascal (version 3.0) for the IBM PC and compatibles. A major part of the development was done on a COMPAQ Portable. TURBO Pascal is a product of BORLAND International and is a very inexpensive and high quality product. Although it lists for $99, it can be purchased for somewhat less. For that price, everyone should be able to purchase a copy.

The distribution diskette contains either the source version or a compiled version of the program (under version 3.0 of TURBO Pascal with or without the 8087 option). If the latter version is supplied, you don't actually have to purchase the program if the compiled version works on your machine and you never want to modify it. The program is between 3700 and 3800 lines long and has been broken up into several OVERLAY sections to permit its execution in the allowable program space of the compiler. For the user, the use of overlays means that between the main sections of program, the main program will need to access the default disk drive to load in the next section of the program. This is a very quick procedure and only occurs three times during the program. The following files should be present on your distribution diskette. In most cases either the source files or the compiled program files will be supplied unless otherwise requested.

A. Source Program Files

1. MAPCODE.PAS - The main program which controls the execution of the various overlay sections.

2. READCODE.PAS - The overlay section in charge of reading in the data.

3. DECISION.PAS - The overlay section that actually performs all of the coding, display, encoding, and storage operations.

4. WRITESUM.PAS - The overlay section that creates the output file and writes a summary of the coding session to the printer.

5. SUBCODE.PAS - A group of supplemental procedures and functions used in the program.

6. CRTSTUFF.PAS - Much of this and the following three sets of procedures and functions are taken more or less directly from the book Advanced Pascal Programming Techniques written by Paul A. Sand, 1984, Osborne/McGraw-Hill Publishers and modified by me for use with TURBO Pascal. They are used extensively throughout the program for screen control (CRTSTUFF.PAS), integer input and output (INTSTUFF.PAS), real number input and output (REALSTUF.PAS), and text file input and output (TEXTSTUF.PAS). Users of the present program who are interested in these routines should refer to that text.

7. INTSTUFF.PAS - A group of procedures and functions used
in integer number I/O.

8. REALSTUF.PAS - A group of procedures and functions used in real number I/O.

9. TEXTSTUF.PAS - A group of procedures and functions used in handling text file operations.

10. QUIKCODE.PAS - A version of MAPCODE that performs all of the coding operations on a data set without user intervention. Parameters and options are entered at the beginning of the program only.

11. QUIKREAD.PAS - A modified version of READCODE.PAS used by QUIKCODE.PAS.

12. QUIKDEC.PAS - A modified version of DECISION.PAS used by QUIKCODE.PAS.

13. QUICKSET.PAS - A modified version of READCODE.PAS used by QUIKCODE.PAS.

14. QTABLE.DTA - The critical value table of studentized ranges (in binary form) used in the subset coding procedure.

B. Compiled Program Files

1. MAPCODE.COM - The compiled version of the MAP-CODE program.
2. MAPCODE.000 - Overlay file used by MAPCODE.COM.
3. QUIKCODE.COM - The compiled version of QUIKCODE.PAS.
4. QUIKCODE.000 - An OVERLAY file used by QUIKCODE.COM.
5. QTABLE.DTA - The critical value table of studentized ranges (in binary form) used in the subset coding procedure.

C. User’s Manual Files

1. README.MAN - Basic instruction manual.
2. MAPCODE.MAN - A WordStar version of the user’s manual.

D. Sample Data Files - The following set of example data files are also supplied.

1. INDDATA.PAS - A set of raw input data with 10 species and 10 characters in individual mode. The first two lines of the file contain the number of groups and characters in the file and the names of the groups. The data are oriented such that for each individual in a group, 5 character values are reported. No NC’s are permitted in this format. A different number of individuals are present in each group. This number is specified on a separate line before the data from each group. On a floppy disk based system, a large number of disk accesses must be done in calculating the summary statistics for each group when the data are in this format. This is because the data are stored in a scratch file from which individual character information is then
read, i.e., the matrix must be transposed. When asked by the
program, the user should enter that the data are STORED BY
INDIVIDUAL.

2. CHARDATA.PAS - A set of raw input data with 10 species
and 10 characters in character mode. The first two lines of the
file contain the number of groups and characters in the file and
the names of the groups. The data are oriented such that for
each character in each group, the data from n individuals are
reported. The number of individuals for which data have been
recorded differs not only among groups but also within groups for
the different characters. The number of individuals is the first
number on each data line or set of lines associated with each
character, therefore, when asked by the program, the user should
enter that the data are STORED BY CHARACTER, that the number of
individuals per group are NOT THE SAME, and that the actual
number of individuals are stored WITH THE DATA. These options
are discussed further in the section on Data Input File Formats.

3. GROUPDAT.PAS - A set of Grouped data with 10 species and
5 characters. The first two lines of the file contain the number
of groups and characters in the file and the names of the groups.
Within each group, there are one or more lines containing each of
the following: a) sample sizes for each character, b) mean
character values for each character, and c) the measure of
variability of each character within the group. For this
example, the latter are standard deviations of the individual
observations (option 1 of the prompt screen).

4. SUMDATA.PAS - A set of summarized data with 10 species
and 5 characters. The first two lines of the file contain the
number of groups and characters in the file and the names of the
groups. Three lines or groups of lines follow for each
character. These are a) the pooled measure of within-group
variability, b) the sample sizes for that character within each
group, and c) the group means for the character from each group.

5. TAMIAS.PAS - This is a set of real data on 15
morphometric characters from 11 populations of Eutamias chipmunks
as taken from Sullivan, R. M., 1985, Phyletic, Biogeographic, and
Ecologic Relationships Among Montane Populations of Least
Chipmunks (Eutamias minimus) in the Southwest, Systematic
Zoology, 34:419-448. Select the GROUPED DATA option.
Data Input File Formats

You will need to create a file containing your input data file. The file should be an ASCII text file. This means that if you create this file with a text editor such as WORDSTAR, you must use the NON-DOCUMENT mode. Otherwise all sorts of weird characters are inserted in the file. Presumably, an appropriate output data file can be created from various data base manager programs, such as 1-2-3, DBase, or Reflex. I have not yet written the program to accept .DIF files. A proper data file can also be created with the TURBO Pascal editor. The file extension is assumed to be .PAS, although any other extension may be used.

The program has been written to accept a variety of formats. All of the data are read in free format style so you need only (and must) place a single space between any two numbers in a sequence. More spaces may be placed between numbers for convenience or clarity. Unfortunately, the program is not immune to errors in data formats and is most likely to crash while reading in data which has not been properly formatted, e.g., the incorrect number of data values or taxon names are placed on a line. It is very unlikely to crash after the data have been read in. A tally of where the program is in reading your data is continuously displayed on the screen and should help you in finding any deficiencies in the data.

If the program does crash while reading your data, a very cryptic little message will be displayed on the screen and the program will halt. If this happens, initially try running the program again, making sure that you have entered the proper options. Next, get a printout of your data and make sure that it is properly formatted and contains the proper number of names, sample sizes, raw data values, measures of variability, and/or character means. Recheck the format with the instructions given below and with the appropriate sample data file. If all else fails, call me (just remember, there's a 5 or 6 hour difference between Hawaii [no daylight savings time] and the east coast of North America and a 2 or 3 hour difference between Hawaii and the West coast - and I don't get in to work until 8 or 9 AM, but don't leave until about 6 PM). My phone number is 808-948-6146 (my office) or 808-948-8617 (Dept. Office). You may also write to me at Department of Zoology, University of Hawaii, Honolulu, HI 96822.
General Input Formats

In the discussion below, from 3 to 5 line types are specified. A line type may specify that only a single value will appear on a single data line in your file or it may specify that a large number of values will appear on a series of sequential data lines. All of the associated values will, in general, be of the same type, e.g., a list of character means from a particular group may require 2 or more data lines in the file. In general, you will repeat the various line types for each group or character. Although it is not necessary for all the data values on a physical line to appear on your screen at one time, this is convenient for checking that the values are all correct. You may have different numbers of data values on each line if this is necessary or convenient.

All data files should contain the following:

LINE TYPE 1: Number of groups, taxa, or species; Number of characters
(Two numbers separated by at least one space; no commas or other punctuation)

LINE TYPE 2: GROUP NAMES - A list of group names. Each name may be up to 20 characters long, although the output from the program will contain only the first 8 or 10 characters depending on the user specified output format (whether for input into PAUP, PHYLIP, or NTSYS). Successive names should be separated by at least one space and the names should not contain any spaces themselves (otherwise the part before the space will be used as one name and the part after the space will be used as the name of the next group). Obviously, if you have any more than a couple of groups you will have to extend the names onto additional lines. You may have any number on a single line.

LINE TYPES 3, 4, and 5: The DATA in the format specified below. Numbers do not need to be aligned in columns. They must, however, be separated by at least one space. NO PUNCTUATION is permitted in the data, i.e., DO NOT USE COMMAS OR SEMICOLONS to separate values in a line like you might use in data statements of a BASIC program. You have moved out of the dark ages into Pascal - see the light between your data - USE SPACES.

Data files

1. # of Groups; # of characters → 10...37
2. List of Group names (up to 20 chars each) i.e. Toot-Toot
3. 

For MAPcode
- cannot use >40 indivs/set and must use
more than a indivs group
Raw Data Input Formats

1. Individual Mode: In individual mode, a variable number of individuals have been measured for each of the c characters with all of the measurements associated with a single individual appearing on one data line or a sequence of data lines. This implies that no MC's (missing data values) are permitted if this input mode is used.

LINE TYPE 1: Number of Groups, Number of Characters

LINE TYPE 2: Group Names (may be continued onto several lines)

LINE TYPE 3: Number of Individuals in Group i

LINE TYPE 4: Separate measurements on an individual for all characters. The data for a single individual may be continued onto additional lines. Each data value is separated from the next by at least one space. LINE TYPE 4 will be repeated for each of the individuals in a group.

Line types 3 and 4 are repeated for groups 1 to t. The data for each group will begin with a line specifying the number of individuals in the group. The next set of n lines or groups of lines will contain the actual data for the c characters for the n individuals.

2. Character Mode: There are three possible formats for raw data in character mode. Each will have c characters and a total of n(i) individuals measured in a group. If n(i) is constant for all characters within a group, then it needs to be specified only once for each group. However, if n(i) varies from character to character, then n(i) must be specified for each character. The values of n(i) may appear as a list of integer values (separated by spaces) prior to the actual data or a single integer value may appear on the same data line as the data for each character.

LINE TYPE 1: Number of Groups, Number of Characters

LINE TYPE 2: Group Names (may be continued onto several lines)

a) Equal Numbers Of Measurements For Each Character (although this may differ among taxa).

LINE TYPE 3: Number of individuals per character.

LINE TYPE 4: Measurements for character i, with all measurements separated by at least one space. The values for a single character may continue onto several lines, but the values for each new character must start on a new line.

Line types 3 and 4 are repeated for each group (taxon).
b) Unequal Numbers Of Measurements For Each Character. The user informs the program whether option 1 or 2 is used at the time of execution of the program (OPTION 1 => Number of individuals NOT THE SAME and actual numbers stored SEPARATELY FROM THE DATA; OPTION 2 => NOT THE SAME and stored WITH THE DATA).

OPTION 1: (Numbers of individuals stored SEPARATELY FROM THE DATA)

LINE TYPE 3: A list of the number of individual measurements for each character in group i.

LINE TYPE 4: Measurements for character i, with all measurements separated by at least one space. The values for a single character may continue onto several lines, but the values for each new character must start on a new line.

Line types 3 and 4 are repeated for each group (taxon).

OPTION 2: (Number of individuals STORED WITH DATA)

LINE TYPE 3: A number specifying the number of individual measurements for character i followed by the measurements for character i, with all measurements separated by at least one space. The values for a single character may continue onto several lines, but the values for each new character must start on a new line and be preceded by an integer specifying the number of individual measurements for that character.

Line type 3 is repeated for each group (taxon).
Group Data Input Format

This type of data is characterized by having three separate values specified for each character in each group: the sample size, the character mean, and some measure of variability among individuals within the group. The data are reported by group with the character information for all characters for each group separated sequentially.

LINE TYPE 1: Number of Groups, Number of Characters

LINE TYPE 2: Group Names (may be continued onto several lines)

LINE TYPE 3: Sample Sizes - a list of number of individual measurements upon which the group means and measures of variability are based for each character. Each sample size should be separated by at least one blank. The list may continue onto more than one data line.

LINE TYPE 4: Sample Means - a list of sample means for each of the c characters. Each value should be separated by at least one blank. The list may continue onto more than one data line.

LINE TYPE 5: Sample Measures of Variability - a list of measures of variability corresponding to the c sample sizes and character means in line types 3 and 4 above. One of 5 different measures of variability may be used, although these cannot be mixed within a data set. These are:

a. Sample standard deviation.
b. Sample variance.
c. Standard error (deviation) of the mean.
d. Two times the standard error of the mean.
e. Variance of the mean.

Line types 3, 4, and 5 are repeated successively for each group.
Summarized Data Input Format

This type of data is characterized by having the sample size and the character mean specified for each group along with some pooled measure of variability for the character among individuals within groups, e.g., the pooled within-group variance from an analysis of variance. The data are reported by character with the measurement or count information for each character for all groups separated sequentially.

LINE TYPE 1: Number of Groups, Number of Characters

LINE TYPE 2: Group Names (may be continued onto several lines)

LINE TYPE 3: A single pooled measure of within group variability for a character. One of 5 different measures of variability may be used, although these cannot be mixed within a data set. These are:

  a. Pooled within-group standard deviation.
  b. Pooled within-group variance.
  c. Pooled within-group standard error (deviation) of the mean.
  d. Two times the pooled within-group standard error of the mean.
  e. Pooled within-group variance of the mean.

LINE TYPE 4: Sample sizes within groups on which the sample means for a character are based.

LINE TYPE 5: Sample means from each group for a single character.

Line types 3, 4, and 5 are repeated successively for each character.
REFERENCES


Farris, J. S. 1968.

