CANPLOT: A FORTRAN-77 PROGRAM FOR PLOTTING
STRATIGRAPHIC DATA ON A POSTSCRIPT DEVICE

IAN D. CAMPBELL and JOHN H. MCANDREWS

Department of Botany, Royal Ontario Museum, 100 Queen's Park, Toronto, Ontario, Canada M5S 2C6
and Department of Botany, University of Toronto, Toronto, Ontario, Canada M5S 1A1

(Received 26 June 1991; accepted 19 November 1991)

Abstract—Many types of stratigraphic data (for example pollen, diatoms, bulk geochemistry) are
presented conveniently in a depth/frequency diagram having several frequency curves on a common depth
scale. CANPLOT is a FORTRAN-77 program for plotting such stratigraphic data on a PostScript output
device. Several options are included, such as stratigraphically constrained cluster analysis of the data, and
a variety of graphical enhancements for producing camera-ready figures.

Key Words: Biostratigraphy, Pollen diagrams, Stratigraphically constrained cluster analysis.

INTRODUCTION

Biostratigraphic and other types of data may be best
presented as a series of frequency curves with a
common Y-axis representing depth or relative age
(Figs 1 and 2). Such diagrams may be used to
represent stratigraphic pollen data, and are termed
resolved diagrams (Faegri and Iversen, 1989). CANPLOT is a FORTRAN-77 program to produce
resolved pollen diagrams using a PC and a PostScript
output device.

PostScript is a versatile device-independent page-
description language, usually incorporated in laser
printers and typesetting machines. Several software
companies also sell PostScript interpreters which
are able to drive almost any raster output device. CANPLOT produces a PostScript file which can be
sent to any PostScript interpreter.

CANPLOT was developed in the Department of
Botany at the Royal Ontario Museum. The pro-
gram's precursor, PLOT10, itself was derived from
another program named CALPLOT.PRO, whose
origin is lost in the mists of early 1960s Minnesota.
CALPLOT.PRO was brought to Toronto by J. H.
McAndrews, where it and its descendants have been
running for the last 23 years on a mainframe, produc-
ing draft-quality diagrams. Escalating mainframe
costs and yet another change in the University's
hardware prompted a major revamping of the pro-
gram in 1989, during which it was determined to be
a relatively simple task to modify it to run on a PC.
CANPLOT now runs on a PC, with printouts coming
from a NEC LC 890 Silentwriter PostScript laser
printer. Because the mainframe account was used
almost exclusively for pollen diagrams, it has been
closed; we have realized considerable savings. Com-
piled using WATFOR-77 and running on an 80286
machine, the program takes about 2 min to plot an
average pollen diagram; the printer takes another
minute or two to print it. Hence, our time-savings
also are considerable.

Several features have been built into CANPLOT,
to enhance its usefulness. Some of these are:
stratigraphically constrained cluster analysis, follow-
ing the method developed by Grimm (1987), the
ability to add taxa together to form synthetic taxa,
the use of R-values and Y-intercepts for converting
data from pollen to biomass, vertical scaling by
either level number (depth) or by interpolated date,
exaggeration curves for rare taxa, upright and italic
lettering with variable pitch, several pages of output
may be created, each numbered, up to 250 taxa and
250 levels may be used in a single data set, the
data format is relatively compact for efficient use
of disk space, the output file may be modified using
a text editor to alter the PostScript code, convenient
for use on PCs, nine styles of zone boundary are
supported, comments can be placed in both left
and right margins, adjacent to specific levels, the
"pollen sum", or total count for each level, can
be plotted as a separate curve, and bargraph or
sawtooth output.

Despite all these features, there may be a need to
modify the printed output. This can be accomplished
effectively and easily through cut-and-paste, if the
user is not familiar with PostScript.

PROGRAM STRUCTURE

Most of the calculations are done in the main
portion of the program. Subroutines are used for
controlling (subroutine CONCTL), carrying out
(subroutines CONISS and CCLUS, and functions
UPDATE and D2, all of which are modified from
Grimm, 1987), and plotting (subroutine TREED) the cluster analysis, if this option is selected. Subroutines are used for the plotting of text (subroutine SYMBOL), for the actual drawing of the curves (subroutines SCREEN and CLIP), for drawing the horizontal axes (subroutine HORAX), for initializing the PostScript file (subroutine INIT), for drawing zone boundaries (subroutine ZONE), and for reading R-value and Y-intercept data (subroutine RVALIN).

The code is presented in Appendix 1 and contains numerous explanatory comments. Appendix 2 shows the data file used for Figure 2.

INPUT DATA FORMAT

The input format is described in "cards" and "columns"; in standard FORTRAN, no line may exceed 80 characters.

Cards 1 and 2

These cards are COMMENT LINES 1 and 2. They will be printed in small type below the title of the diagram. They are 72 characters long, and may be used for such information as the name of the analyst, the date of the diagram, the name of the data file, etc. Note that although the special characters {, }, [, ], (, ), , /, , and % may be used anywhere else in the CANPLOT input, they must not be used in the comment lines.

Card 3

The first part of this line is the 52 character title, printed in bold letters at the top of the plot, and also printed at the top of the listing.

The second part, starting in column 54 and ending in column 73, is the 20 character Y-axis label. Where this is significantly shorter than the allowed 20 characters, the final diagram may look better if the label is centered approximately in its field.

Card 4

NTAX is the number of taxa, in columns 1–3. It is right justified in its field, with no decimal. Any synthetic taxa should be included in this number.

NLEVS is the number of levels in columns 5–7. It is right justified in its field, with no decimal.

NBSC is the number of taxa in the sum used as divisor for calculating percentages, in columns 9–11. It is right justified in its field, with no decimal.

INSTDV is the instruction on the method to use for calculating the percentages, in column 13. A 0 or blank indicates that the sum is divisor in all situations. A 1 indicates that the divisor for each taxon is the sum plus the value for that taxon if it is not listed as one of the taxa in the sum. This option is

Hams Lake ON Percentage Pollen Diagram

Figure 1. CANPLOT pollen diagram. Filled portion of curve represents percentage for that species at that level; gray curve represents 10 x exaggeration and is useful for taxa with low abundances. Dendrogram at right is optional in CANPLOT. Hams Lake diagram was published originally by Bennett (1987); additional pollen counts in Zone 4 are from McAndrews (unpublished).
used when taxa outside the sum are extremely abundant. A 2 indicates that the sum of all taxa is used in all situations. This option is rarely used.

ICTL is the instruction for CONISS, in column 15. A 0 indicates that CONISS should not be run. A 1 indicates that a square-root transformation should be used. A 2 indicates that the data should be normalized. Any other value causes no transformation of the data to be performed in the cluster analysis.

INSZON is the number of zone boundaries to be placed, in column 17.

INSPAP is the paper length for the plot, in column 19. A 0 indicates USA letter-size paper. A 1 indicates USA legal-size paper.

IBAR is the switch for bargraph or sawtooth curve, in column 21. A 1 is for bar, 0 for sawtooth. The convention for the data entry for bars is that the sample level is the top of the sample interval; the bar thickness goes down to the next level number. For unsampled intervals, use a dummy level with no counts (but keep the sum nonzero, or the program will crash with divide-by-zero error). Because the sample number is the top of the sampled interval, a dummy level must be used at the end of the file to indicate the thickness of the last bar. For instance, if the last level is 595 and levels are at 5 cm intervals, then a dummy level of 600 should be added. In this dummy level, all values should be 0 except the sum.

If the sum is composed of several taxa being added together rather than a separate variable for the sum, then a dummy taxon may be added to the list for the sum, so that the dummy level may have a non-0 sum.

INNO is the zero-value switch, in column 23. A 0 indicates that taxa with no values should be ignored. A 1 indicates that all taxa are to be plotted, even if they have no values.

INSSUM is the instruction on the plotting of the sum, in column 25. A 0 indicates that the sum is not to be plotted. A 1 indicates that the sum is to be plotted following the last taxon.

INSYN is the number of synthetic taxa to use, in column 27.

INSR is the switch for R-values, in column 29 (see Card 10.1). A 0 indicates R-value conversion off, 1 is conversion on.

INDATE is the instruction regarding the vertical scale, in column 31. A 0 causes the level numbers to be used for vertical scaling. A 1 causes the vertical scale to be adjusted by interpolating from radiocarbon (or other) dates. The number of dated levels is given in IDATES. Note that vertical scaling must be ON for level-bound comments (see Card 4.3) to work.

INSCOM is the number of level-bound comments to be printed in either the left- or right-hand margins of the diagram, in columns 33–34. See Card 4.3.
IDATES is the number of dates to be used for vertical scaling, in column 36-37. Note that if INDATE is switched off, then the cards carrying dates become equivalent to level-bound comments. See Card 4.3.

IEXAG is the value for the exaggeration curve, in columns 39-40. It is any positive integer, right justified in its field. The normal value is 10. To obtain no exaggeration curve, use a value of 1 or 0. The maximum value is 99.

XDMIN is the minimum width of plot for each taxon, in columns 42-43. It must have a decimal point. The usual value is 0.3; it is not recommended to use less than 0.2 or greater than 0.5.

DS is the value to use as the X-axis increment for the sum only, in columns 45-52. If INSSUM is 1, then the pollen sum for each level is graphed using 1" = DS for the X-axis. It may occur anywhere in its field, but must have a decimal point. Only two digits after the decimal are read. The usual value is the average pollen sum for that data set.

**Card 4.1**

This card carries the identity number (columns 1–3) and number of taxa (columns 5–6) to use for a synthetic taxon. The maximum number of taxa to be included in a single synthetic taxon is 25. Synthetic taxa may be included in any of the operations involving regular taxa, including CONISS and the formation of other synthetic taxa (these "higher order" synthetic taxa should have a higher number). This card occurs only if INSYN (Card 4) is not 0. If INSYN is 2 or more, then Cards 4.1 and 4.2 alternate until all synthetic taxa have been numbered.

**Card 4.2**

This card carries the numbers to be summed to form a synthetic taxon. They are right justified in fields of 3. The maximum number of taxa to be included in a single synthetic taxon is 25, but synthetic taxa may be added to form higher order synthetic taxa (see Card 4.1). This card occurs only if INSYN is > 0.

**Card 4.3**

This card is optional, occurring only if INSCOM and IDATES (Card 4) are together greater than 0. It carries and controls level-bound comments and data control information for vertical scaling.

Column 1 carries a code for the printing of the comment. A 0 indicates that no comment is to be printed. A 1 indicates that a 1-line comment (12 characters) should be printed on the left-hand margin. A 2 causes a 2-line comment to be printed in the left margin. A 3 causes a 1-line comment to be printed in the right margin. A 4 causes a 2-line comment to be printed in the right margin.

Columns 3–7 carry the date to be associated with a level for vertical scaling purposes. It must be an integer and should be expressed in years BP.

Columns 9–20 are a 12-character comment. It will be printed in the position specified by the code in column 1. Special characters are allowed.

Columns 21–32 are a second 12-character comment. It also will be printed (as the second line) in the position specified by the comment control in column 1.

Columns 34–37 are a level number signifying the start of the interval for the comment.

Columns 39–42 are the level number for the bottom of the comment. CANPLOT will average these two level values to determine the level at which to attach the comments, as well as for the interpolation of dates for the levels. Two levels are allowed to permit the use of bulk sediment dates which may be taken over several centimeters of sediment.

Note that cards carrying dates must precede cards carrying only comments. Note also that all fields must be filled, either with values or with blanks.

The number of repetitions of this card should equal the sum of INSCOM and IDATES (Card 4).

**Card 5**

Columns 1–3 are NVARS, the number of taxa to be used in the cluster analysis. It should be right justified in the field.

Columns 5–7 are CCX, the width of the constrained cluster analysis X-axis in inches. A decimal is required.

Columns 9–12 are CHITE, the size in inches of a capital A in the name of a taxon of average abundance. All lettering will be scaled relative to this value. A decimal is required. The value 0.13 is suggested.

Columns 14–18 are the resolution of the PostScript output device in dots per inch (dpi); it should be right justified in the field. Most laser printers use 300 dpi. No decimal should be used.

Columns 20–36 are the label to be used for the "sum", if INSSUM (J on Card 4) is 1. Even if no label is used and the sum is not to be plotted, this line must contain at least 36 characters.

**Card 5.1**

This card is included only if INSZON (Card 4) is not 0. Columns 1–4 are the level below which a zone boundary should be drawn. Note that the number supplied here is not the level itself, but the number of the level—for instance, 5 indicates the fifth level, not the level labeled as 5.

Columns 6–7 are a style definition for the boundary. At present, 9 styles are defined: thin, medium, or thick lines in white, black, or gray (Table 1). This set of a level and a style repeat with a single space between sets to a maximum of 9 sets (the maximum number of zone boundaries usable).
Table 1. Codes for selecting zone boundary style

<table>
<thead>
<tr>
<th>Thin</th>
<th>Medium</th>
<th>Thick</th>
</tr>
</thead>
<tbody>
<tr>
<td>White</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Black</td>
<td>4</td>
<td>5</td>
</tr>
<tr>
<td>Gray</td>
<td>7</td>
<td>8</td>
</tr>
</tbody>
</table>

**Card 5.2**

This card is included only if NVARS (Card 5) is not 0. It contains the numbers identifying the taxa (right justified in contiguous fields of 3 digits) to be used by the constrained cluster analysis. Note that the cluster analysis may be switched on or off with ICTL without affecting the need for this card; only NVARS determines the need for this card. This card is repeated until all taxa for CONISS have been listed.

**Card 6**

Columns 1–6 are DX, the value-per-inch for the horizontal axis for all taxa. It is anywhere in its field, but must have a decimal. Usual is 50.000.

Columns 7–12 are RX, unused at present.

Columns 13–18 are AY, the length in inches of the vertical axis. It is anywhere in its field, but must have a decimal. Usual is 5.0000.

Columns 19–24 are SY, the number to show at the bottom of the vertical axis. It is anywhere in its field, but must have a decimal. It is usually the maximum depth. If the levels are depths below surface and entered as positive values, then this number should have a minus sign.

Columns 25–30 are DYM, the minimum depth or highest level to be plotted. It is anywhere in its field but must have a decimal. It is normally either 0 or the minimum depth. If the levels are depths below surface and expressed on the data cards as positive values, then this number should have a minus sign.

Columns 31–36 are DYN, the maximum depth on the vertical axis. It is anywhere in its field but must have a decimal. It is normally the same as columns 19–24. If the levels are depths below surface and expressed on the data cards as positive values, then this number should have a minus sign.

**Card 7**

Columns 1–3 are the number of taxa to be plotted. It is right justified.

**Card 8**

This card carries the numbers of the taxa to be plotted, right justified in fields of three. This card is repeated until all taxa to be plotted have been listed. The maximum is 25 taxa per card. The plot terminates at the first blank field, so all cards except the last must contain the full 25 taxa numbers. The taxa are plotted in the order in which their numbers are listed on Card 11.

**Card 9**

Columns 1–3 are a species number, right justified. Columns 4–20 are a species name, 17 characters long. This 20 character sequence is repeated to fit 4 taxa to a card. This card is repeated until all taxa required have been listed. The number of taxa listed should equal NTAX (Card 4).

Italics may be obtained by inserting the character "~", which will not be printed. Upright letters will resume at the start of the next taxon name (unless otherwise specified with "\~"), or at the symbol "\w", which also will not be printed. Both symbols, where they occur (except in the first position of a name), take the place of a blank; that is, on the final plot, a blank will be inserted where these characters occur in the middle of a string.

**Card 10**

This card has the numbers of the taxa in the sum, right justified in fields of 3. Maximum is 26 numbers on 1 line. It is repeated to list all taxa in the sum. See NBSC on Card 4.

**Card 10.1**

This card is optional, occurring only when INSR (Card 4) is 1 rather than 0. It carries the number of taxa for which R-values are given, right justified in the first three columns.

**Card 10.2**

This card is optional, present only when INSR (Card 4) is 1 instead of 0. It carries the R-values and Y-intercepts for conversion from percent pollen to percent biomass.

The first three digits (columns 1–3) are a species number, right justified.

The second number (columns 5–10) is the R-value for that species. It must have a decimal.

The third number (columns 12–17) is the Y-intercept for that species. It must have a decimal.

Values are given for 4 taxa per card. If the number of taxa for which R-values are given is not divisible by 4, the last repetition of this card must be padded with blanks to fill all fields.

The formula used for the conversion is (pollen% - Y-intercept)/R-value. Negative values are converted to zeros. The percentages are recalculated following conversion, and some screen messages during the program run will repeat.

**Card 11**

This card signals the beginning of the data set. It carries the word "DAT" in the leftmost three columns, in upper or lower case.
Card 12

Columns 1–3 are for identification, and may contain any characters except “END”.
Columns 4–7 are the level number, right justified. It must be an integer.
Columns 8–10 are a taxon number, right justified.
Columns 11–14 are the value for that taxon at that level. It may be an integer or have a decimal, but no digits after the decimal will be used. It should be right justified in its field.

Taxon numbers and counts are repeated as groups of 3 and 4 columns, without intervening blanks, until the card is full (10 taxa with counts). This card is repeated until all data are given. If more than one card is needed for a level, then each card except the last for that level must be filled. The levels must be in numerically ascending order, but the order of the taxa within the cards does not matter.

Card 13

This card signifies the end of the data and of the file. It carries the word “END” in the leftmost three columns, in upper or lower case.

INPUT AND OUTPUT

The input file expected by CANPLOT is P10DATA.IN. The output files are named FT07F001 and FT10F001 by the WATFOR-77 compiler. All file names can be changed by running a batch file in DOS with the appropriate commands. FT07F001 contains a listing of the input data and the calculated percentages, as well as the actual numbers resulting from the cluster analysis, if performed. It also can be useful in detecting input errors. FT10F001 contains PostScript code for drawing the diagram on a PostScript device. The code in this file has been tested extensively on a NEC SilentWriter LC-890 in PostScript mode.

CALCULATION AND PLOTTING OF POLLEN CONCENTRATION AND INFLUX

Concentration (density) is the number of pollen per cubic centimeter or gram. It can be determined by the aliquot method (Traverse and Ginsberg, 1966; Davis, 1965) or more generally by the exotic (spike) method (Berglund and Ralska-Jasiewiczowa, 1986). Our preference is the exotic method where a known number of Lycopodium spores are mixed with a measured volume of sediment before fossil pollen are concentrated by physical and chemical reduction of the sediment matrix. The introduced spores are counted together with the fossils and the cc sediment examined is calculated, that is

\[
\text{cc examined} = \frac{(cc \text{ sediment}) \times (spores \text{ counted})}{\text{spores added}}.
\]

When the cc examined is divided into the fossil counts, the concentration is obtained, that is

\[
\text{fossil concentration per cc} = \frac{\text{fossils counted}}{\text{cc examined}}.
\]

In practice the cc examined value is entered into the data file as a “taxon” and used as the “pollen sum” for making concentration calculations.

For calculation of pollen influx the rate of sedimentation is determined as years/cm sediment accumulation and multiplied by the cc sediment examined to obtain the cm² year examined, that is

\[
\text{cm² year examined} = \frac{\text{years} \times (\text{cc sed. examined})}{\text{cm}}.
\]

The value is entered into the data file as a “taxon” and used as the “pollen sum” for making influx calculations.

Note that although both “cc examined” and “cm²year examined” values are real numbers, they must be entered as integers for CANPLOT. It is usually most practical to multiply the values by a factor of 10, 100, or 1000, then round off to the nearest integer. This will require annotating the horizontal scale of the resulting diagram, either through a comment line or through “cut-and-paste” on the resulting diagram.

Acknowledgments—We thank Ted Irwin for his patience in field-testing CANPLOT during its development. We also thank Terry Jones and all the others who have developed and maintained PLOT10 and its predecessors through the years. Copies of CANPLOT with a companion program for creating data-files may be obtained from Ian Campbell.

REFERENCES

APPENDIX 1

Program Listing

C The CANPLOT logo is generated with ASCII code, and may need to be replaced
C on some installations.

C The INQUIRE and subsequent statements checking for the input file may
C require modification for some installations.

INQUIRE(FILE='PIODATA.IN ', EXIST=EX)
IF(.NOT.EX) THEN
   WRITE(6,' ('^G***Sorry, I cannot find that file.''
   : ''Execution aborted.'')')
   STOP
END IF

OPEN(UNIT=5, FILE='PIODATA.IN')

COMMON/RV/RVAL(250), YVAL(250)
COMMON/LASER/IRES
COMMON/CHRT/CHITE
COMMON/PERC/MAT(251, 250)
COMMON/CTL1/LPAG, NTEX, PAPL, MSWIT(250)
COMMON/DEND1/AY, CCX
COMMON/CTL2/NVARS, ICTL, DYN, YLEV(252)
COMMON/CTL3/NN, XAXIS

CHARACTER*I7 NMT, NAME(252), SUMNAM
CHARACTER*20 LBL
CHARACTER*3 DEND

DIMENSION LABIL(52), NMBR(252), DCOM(25, 24)
DIMENSION LABEL(52), TXT(2, 75), DCOM(25, 24)
DIMENSION LABIL(52), NMBR(252), NTEX(252), ITEL(80), NCTAX(250), IDD(25)
DIMENSION NTEX(252), IBLANI(1), NLEV(506), KZON(10), ITZON(10)
DIMENSION PC(II), LEV(252), SUMV(252), LSWIT(252), IVA(25), IDB(25)
DIMENSION ISYN(10), MSYN(10, 25), ICOM(25), IDATE(25), IDC(25)
REAL MAT, XPER(252), XIOPER(252)
REAL XP(252), XIOP(506), XPL(506)

LOGICAL EX

C The CANPLOT logo is generated with ASCII code, and may need to be replaced
C on some installations.

WRITE(6,' (///,''Welcome to ...........'',///)
: lOX,,
: lOX,,
: lOX,,
: lOX,,
: lOX,,
: lOX,,
: lOX,,
: lOX,,
: lOX,,
: lOX,,
: lOX,,

WRITE(6,' (///,''..Campbell and McAndrews, 1990'')')
WRITE(6,' (///,''use <CTL>-<BREAK> to interrupt'',///)

LPAG=1

C CCX is length of x-axis for CONISS
C CHITE is character height in inches for basic taxon names

MAXTAX = 250
MAXLEV = 250

C read control cards

READ(5,' (75AI)', ERR=9900) ((TXT(J, I), I=I, 75), J=J, 2)
READ(5,' (52AI, IX, A20)', ERR=9900) (LABEL(I), I=I, 52), LBL

C IOS in the following statement is the error number generated by the compiler
C and may require modification of statements 9010 and subsequent for some
C compilers/installations
READ(5, ' (I3, 1X), 10(I1, 1X), 3(I2, 1X), F2.1, 1X, F8.2)',
  IOSTAT=IOS, ERR=9010)
  NTAX, NLEVS, NBSC, INSTDV, ICTL, INSZON, INSPAP, IBAR, INNO, INSSUM, INSYN,
  INSR, INDATE, INSCOM, IDATES, IEXAG, XDMIN, DS
IF(INSYN.GT.0) THEN
  DO 5 I=1, INSYN
    READ(5, ' (I3, 1X, I2)') ISYN(1), NSYN(I)
    IF(NSYN(I).GT.25) THEN
      WRITE(6, ('**G***NUMBER OF TAXA TO SUM FOR SYNTHETIC'',
          **'','' TOO LARGE'',/,**'',''MAX IS 25.'')) ISYN
    END IF
    READ(5, ' (25(I3))') (MSYN(I,J), J=I, NSYN(I))
  5 CONTINUE
END IF
C INSCOM is the number of comments attached to levels
C IDATES is the number of dates provided for interpolation
C The total of INSCOM+IDATES must not exceed 25
C INDATE is the switch for vertical axis by depth or date (0 or 1)
  INSLIN=INSCOM+IDATES
  IF(INSLIN.GT.25) THEN
    WRITE(6, ('**G***TOO MANY LEVEL COMMENTS AND DATES'',/
        **'',''MAX FOR BOTH TOGETHER IS 25 (CARD 4).'''))
    STOP
  END IF
  IF(INSLIN.GT.0) THEN
    DO 6 I=1, INSLIN
      READ(5, ' (II, IX, I5, 1X, 24AI, 2(IX, I4))')
      ICOM(I), IDATE(I), (DCOM(I,J), J=I, 24), IDA(I), IDB(I)
      IDC(I)=NINT((IDA(I)+IDB(I))/2.)
    6 CONTINUE
  END IF
  READ(5, ' (I3, 1X, F3.1, 1X, F4.2, 1X, I5, 1X, A17)'), NVARS, CCX, CHITE, IRES, SUMNAM
  IF(INSLIN.GT.0) THEN
    READ(5, ' (10(I4, 1X, I2, 1X))', ERR=9030)
    (KZON(I), ITZON(I), I=I, INSZON)
  END IF
  CALL INIT
  DO 10 I=1, NTAX
    MSWIT(I)=0
  10 CONTINUE
  IF(NVARS.GT.0) THEN
    READ(5, ' (2513)', ERR=9040) (NCTAX(I), I=I, NVARS)
    DO 20 I=1, NVARS
      DO 20 J=I, NTAX
        IF(J.EQ.NCTAX(I)) MSWIT(J)=I
      20 CONTINUE
    END IF
  READ(5, ' (6(F6.2))', ERR=9050) DX, RX, AY, SY, DYM, DYN
  C the variable RX is currently unused, but is retained for compatibility
  with data-files from older versions of the program.
  READ(5, ' (I3)', ERR=9060) NTP
  READ(5, ' (2513)', ERR=9070) (NTXP(K), K=I, NTP)
  DO 25 K=1, NTP
    IF(NTXP(K).EQ.0) THEN
      WRITE(6, ('**G***ERROR. BAD NUMBER OF TAXA TO PLOT.'',/
          **'','' OR BAD NUMBER IN LIST OF TAXA TO PLOT. ABORT.''))
      STOP
    END IF
  25 CONTINUE
  WRITE(7, (' CANPLOT''//'//2X, 52AI//'' Y-AXIS LABEL: '', A10,
      **'','' NTAX='', I3,'','' NLEVS='', I3,'','' NBSC='', I3,'','' INSTDV='', I1,/ 
      **'','' ICTL='', I1,'','' INSZON='', I1,'','' INSPAP='', I1,'','' IBAR='', I1,
      **'','' INNO='', I1,'','' INSSUM='', I1,'','' ICTL=', I1, //
      **'','' IEXAG='', I2,'','' XDMIN='', F3.1,'','' DS='', F9.2)'')
      (LABEL(I), I=I, 52), LBL, NTAX, NLEVS, NBSC,
      INSTDV, ICTL, INSZON, INSPAP, IBAR, INNO, INSSUM, ICTL, IEXAG, XDMIN, DS
C test for control card errors
  IF(NTAX.GT.MAXTAX .OR. NLEVS.GT.MAXLEV .OR. NBSC.GT.NTAX .OR. INNO.LT.0.
Plotting stratigraphic data on a PostScript device

:OR.INNO.GT.1.OR.INSSUM.LT.0.OR.INSSUM.GT.1.OR.IEXAG.LT.0.OR.
:XMFIN.LT.0.2) GO TO 430

C PAPL is the image-length, and is (paper-length) - 2.7 inches. It should be
C modified for installations using non-US standard letter or legal paper.
IF(INSPAP.EQ.0)THEN
  PAPL=8.3
ELSE IF(INSPAP.EQ.1)THEN
  PAPL=11.3
ELSE
  WRITE(6,'(***BAD INSPAP=",I5,"DEFAULT: LETTER")')INSPAP
  INSPAP=0
  PAPL=8.3
END IF
NN=NLEVS-1
IF(IBAR.EQ.1)THEN
  NBARS=NLEVS*2
ELSE IF(IBAR.NE.0)THEN
  WRITE(6,'(***BAD IBAR=',I5,"DEFAULT: SAWTOOTH")')IBAR
  IBAR=0
  PAPL=8.3
END IF

NN=NLEVS-1
IF(IBAR.EQ.1)THEN
  NBARS=NLEVS*2
ELSE IF(IBAR.NE.0)THEN
  WRITE(6,'(***BAD IBAR=',I5,"DEFAULT: SAWTOOTH")')IBAR
  IBAR=0
  PAPL=8.3
END IF

READ(5,’(4(I3,A17))’),ERR=9080)
  (NMBR(I),NAME(I),I=1,NTAX)
WRITE(7,’(TAXON NUMBERS AND NAMES."/(2X, 3(I3,1X,A17,2X))")
  (NMBR(I),NAME(I),I=1,NTAX)
C output divisor type
IF(INSTDV.EQ.0)THEN
  READ(5,’(2613)’,ERR=9090) (BSCTX(I),I=1,NBSC)
  WRITE(7,’(TAXA IN "BASIC SUM."/(2X,15(I4,IX))) ’
  (BSCTX (I), I=1, NBSC)
  WRITE(7,’("BASIC SUM" IS DIVISOR IN ALL CASES." / "))
ELSE IF(INSTDV.EQ.1)THEN
  READ(5,’(2613)’,ERR=9090) (BSCTX(I),I=1,NBSC)
  WRITE(7,’(TAXA IN "BASIC SUM."/(2X,15(I4,IX))) ’
  (BSCTX (I), I=1, NBSC)
  WRITE(7,’("REMAINING TAXA USE "BASIC SUM + VALUE" ")’)
ELSE IF(INSTDV.EQ.2)THEN
  WRITE(7,’("SUM OF ALL TAXA USED AS DIVISOR." / "))
ELSE
  WRITE(6,’(***INSTDV ERROR - DEFAULTED TO 0 (BASIC SUM")’)
  INSTDV=0
END IF
C initialise MAT(I,J) and LEV(J)
DO 30 J=1,MAXLEV
  LEV (J) =0
DO 30 I=1,MAXTAX+1
  MAT (I, J) =0.0
30 CONTINUE
C read R-values and Y intercepts
IF(INSR.EQ.1)CALL RVALIN
C read data cards
READ(5,’(A3)’)DEND
IF(DEND.NE."DAT")THEN
  WRITE(6,’("G***DAT DECLARATION NOT FOUND. CARDS MAY BE OUT"’
    ’,"OF ORDER",/,"OR A CARD MAY BE MISSING","/,”OR A CONTROL”
    "CODE MAY BE OUT OF POSITION/MISSING",/,
    "OR THE NUMBER OF TAXA MAY BE TOO LOW",/,
    "...CONTINUING...")")
END IF
WRITE(6,’(".....READING DATA CARDS")’)
JAC=1
J=1
40 READ(5,’(A3,s4,10(I3,F4.0))’,ERR=9100)
  DEND,LC, (NTX(I),PC(I), I=1,10)
IF(DEND.EQ."END")GO TO 90
IF(J.GT.GT.1)THEN
  IF(LC.LT.LEV(J-1))THEN
    WRITE(6,’("G***BAD LEVEL OR LEVEL OUT OF ORDER,"/,
      "BAD NUMBER IS ",I4)
    STOP
END IF
END IF
IF(JAC.EQ.1) LEV(J)=LC
JAC=0
IF(LEV(J).NE.LC) THEN
  J=J+1
  LEV(J)=LC
END IF
DO 70 I=1,10
  IF(NTX(I).EQ.0) GO TO 80
  DO 60 IJ=1,NTAX
    IF(NMBR(IJ).NE.NTX(I)) GO TO 50
    L=IJ
    GO TO 70
  50  IF(IJ.EQ.NTAX) GO TO 440
  60 CONTINUE
  70 MAT(L,J)=PC(I)
  80 IF(J.LE.NLEVS) GO TO 40
  90 CONTINUE
END IF
IF(INDATE.EQ.I) THEN
  C Interpolate dates from given dates, fix vertical axis
  C Do for each interval
  KK=1
  DO 91 I=1,IDATES-1
    K=KK
    DO 92 J=K,NLEVS
      IF(LEV(J).EQ.IDC(I)) THEN
        LEV(J)=NEWYR
        KK=KK+1
      ELSE IF(LEV(J).GT.IDC(I) .AND. LEV(J).LT.IDC(I+1)) THEN
        LEV(J)=NINT((LEV(J)-IDC(I))*YRSCM)+NEWYR
        KK=KK+1
      ELSE IF(LEV(J).EQ.IDC(I+1)) THEN
        LEV(J)=IDATE(IDATES)
        KK=KK+1
      END IF
  92 CONTINUE
  C interpolate levels for the level-bound comments
  IF(INSCOM.GT.0) THEN
    DO 94 J=IDATES+1,IDATES+INSCOM
      IF(IDC(J).GE.IDC(I) .AND. IDC(J).LT.IDC(I+1)) THEN
        IDD(J)=NINT((IDC(J)-IDC(I))*YRSCM)+NEWYR
      ELSE IF(I.EQ.(IDATES-1) .AND. IDC(J).EQ.IDC(I+1)) THEN
        IDD(J)=IDATE(IDATES)
      END IF
  94 CONTINUE
  END IF
  IDD(IDATES)=IDATE(IDATES)
  SY=-IDATE(IDATES)
  DYM=-IDATE(1)
  DYN=-IDATE(IDATES)
END IF
WRITE(7,(/'LEVELS''/(2014))') (LEV(J), J=1,NLEVS)
C check for synthetic taxa
IF(INSYN.NE.0) THEN
  DO 235 I=1,INSYN
  DO 235 J=1,NTAX
  END IF
DO 235 K=1,NLEVS
   DO 235 L=1,NSYN(I)
   IF (NMBR(J).EQ.MSYN(I,L)) THEN
      MAT(ISYN(I),K)=MAT(ISYN(I),K)+MAT(J,K)
   END IF
235 CONTINUE
END IF

C set switch for each species according to its occurrence/non-occurrence
95 CONTINUE
DO 100 I=1,252
LSWIT(I)=INNO
100 CONTINUE
DO 120 I=1,NTAX
   DO I=1,NLEVS
      IF (MAT(I,J).NE.0) LSWIT(I)=I
   I=1,NL
   CONTINUE
   IF (LSWIT(I).NE.0) WRITE (7, ' (Ix, AI7/ (20F4.0)) ') NAME(I), (MAT(I,J),
      J=1,NLEVS)
120 CONTINUE
WRITE (7, ' (/'' LEVELS AND SUMS USED AS DIVISORS. ''/)' )
C calculate percent according to instructions.
WRITE (6, ' ('' ...... CALCULATING''/)' )
IF (NL/NTAX*100.0).GT.1000 WRITE (6, ' (6X,'' (THIS MAY TAKE TIME.),'' )
DO 230 J=1,NLEVS
   SUM=0.0
   IF (INSTDV.EQ.2) GO TO 190
   DO 140 I=1,NBSC
      DO I=1,NTAX
         IF (NMBR(I).NE.BSCTX(I)) GO TO 130
         L=I
         GO TO 140
      I=1,NTAX
   DO 150 K=1,NBSC
      C check if this taxon is in basic sum
      IF (NMBR(I).EQ.BSCTX(K)) GO TO 160
   DO 150 K=1,NBSC
      C if taxon is not in basic sum and INSTDV calls for the sum + taxon (when
      C the taxon is not already in the sum), then calculate the percentage as
      C follows:
      MAT(I,J)=(MAT(I,J)/(SUM+MAT(I,J)))
   180 CONTINUE
   CONTINUE
190 CONTINUE
   GO TO 230
200 CONTINUE
   IF (SUM.EQ.0.0) SUM=1.0
   WRITE (6, ('''G***CAUTION. A SUM OF 0 HAS OCCURRED.'''/)' )
   END IF
   MAT(I,J)=MAT(I,J)/SUM
   GO TO 180
210 CONTINUE
   IF (SUM.EQ.0.0) SUM=1.0
   GO TO 170
C if taxon is not in basic sum and INSTDV calls for the sum + taxon (when
C the taxon is not already in the sum), then calculate the percentage as
C follows:
   MAT(I,J)=(MAT(I,J)/(SUM+MAT(I,J)))
220 CONTINUE
230 CONTINUE
IF (INSR.EQ.I) THEN
   INSR=0
   DO 231 I=1,NTAX
DO 232 J=1,NLEVS
   MAT(I,J) = NINT((MAT(I,J) - YVAL(I)) / RVAL(I))
IF(MAT(I,J).LT.0) MAT(I,J) = 0
232 CONTINUE
231 CONTINUE
GOTO 95
END
IF WRITE(7,'(LEVEL = SUM )')
WRITE(7,'(5(I4,'',' = ',F6.0,3X))') (LEV(J), SUMV(J), J=I,NLEVS)
WRITE(7,'(/PERCENTS TABLE. /LEVELS , /)')
   (LEV(J),J=1,NLEVS)
DO 240 I = I, NTA
   IF(LSWIT(I).EQ.0) GO TO 240
   WRITE(7,'(/,IX, I4,1X, AI7/(13F6.1))') NMBR(I), NAME(I),
      (MAT(I,J),J=I,NLEVS)
240 CONTINUE
DY = (DYM-DYN) /AY
IDY = NINT(DY)
C make y-axis and labels
   IHIT = NINT(I.5*CHITE*IRES)
   IXPOS = - NINT(IRES/I.8)
   IYPOS = NINT(AY*IRES/2-IRES/2)
   WRITE(10,'/(2(I6,1X),'',' M ',I4,' H 90 RS (',A20,') S RE'))
      IXPOS, IYPOS, IHIT, LBL
   C check for level-bound comments
   LCOM = 0
   IF(INSLIN.GT.0) THEN
      XPOS = -.33333
      DHITE = CHITE/1.666666
      DO 285 I=I, INSLIN
         IF(ICOM(I).LT.0 AND ICOM(I).GT.0) THEN
            YPOS = (-IDD(I) - SY) / DY - CHITE/0.5
            ELSE
            YPOS = (-IDD(I) - SY) / DY + CHITE*.1
         END IF
         DO 287 III=1,12
            LABIL(III) = ICHAR(DCOM(I, III))
287 CONTINUE
         CALL SYMBOL(XPOS, YPOS, DHITE, LABIL, 0., 12)
         LCOM = I
385 CONTINUE
   IF(LCOM.EQ.I) THEN
      ISLID = NINT(CHITE*IRES*7)
      WRITE(10,'(/,I6,'',' 0 translate '')') ISLID
      PAPL = PAPL - I
   END IF
IXPOS = 0
END IF
C do comments and title lines
DO 250 I=1,52
   LABIL(I) = ICHAR(LABEL(I))
250 CONTINUE
YPOS = AY + 1.3
XPOS = 0.0
C do comment lines
DO 270 J=1,2
   DO 260 I=1,75
      ITXT(I) = ICHAR(TXT(J,I))
260 CONTINUE
Plotting stratigraphic data on a PostScript device

260 CONTINUE
DHITE=CHITE*0.46
CALL SYMBOL(XPOS,YPOS,DHITE,ITXT,0.,70)
XPOS=XPOS+3.7
270 CONTINUE
IXPOS=0
IYPOS=-IRES/2
IHIT=NINT(CHITE*IRES)
WRITE(10,' (2(I6,1X),''M '',I4,'' H (plotted by CANPLOT) S'')')
: IXPOS,YPOS,IHIT
YPOS=AY+1.5
XPOS=0.0
C title line
DHITE=CHITE*1.25
CALL SYMBOL(XPOS,YPOS,DHITE,LABEL,0.,51)
DO 280 I=1,NLEVS
YLEV(I)=-LEV(I)
280 CONTINUE
WRITE(10,' (''0 0 M'')')
IF(AY.EQ.0.)THEN
WRITE(6,' (''^G***ERROR. VERTICAL AXIS IS 0 INCHES HIGH'')')
STOP
END IF
IXPOS=-NINT(IRES*0.21)
IYPOS=-NINT(IRES*0.22)
IHIT=NINT(CHITE*IRES)
WRITE(10,' (2(I6,1X),''M '',I4,'' H (%) S'')')IXPOS,IYPOS,IHIT
IX=0
KBOTP=-NINT(DYN)
KYP=-NINT(DYM)
IY=NINT(AY*IRES)
C cursor to start of vertical axis
WRITE(10,' ('' initclip newpath '',2(I6,1X),''M LTIC'')')IX, IY
IXL=IX-NINT(IRES/2.4)
IYL=IY-IRES/20
IHIT=NINT(CHITE*IRES)
WRITE(10,' (2(I6,1X),''M '',I4,'', '' S '')')
: IXL, IYL, IHIT, KYP
IF(DY.GT.10000)THEN
ISTP=I0000
ELSE IF(DY.GT.1000)THEN
ISTP=I000
ELSE IF(DY.GT.100)THEN
ISTP=I00
ELSE IF(DY.GT.10)THEN
ISTP=I0
ELSE
ISTP=I
END IF
LLB=IY-IRES/3
KYV=0-ISTP
DO 290 KVP=KBOTP*IRES/IDY, 0,-ISTP*IRES/IDY
KYV=KYV+ISTP
IF(KVP.LT.LLB) THEN
IXL=0-NINT(IRES/2.4)
IYL=KVP-IRES/20
WRITE(10,' (2(I6,1X),''M LTIC'',2(I6,1X),''M ('',I5,,'' S '')')
: 0,KVP, IXL, IYL, KYV
LLB=KVP-IRES/3
ELSE IF(KVP.LT.IY) THEN
WRITE(10,' (2(I6,1X),''M HTIC'')')0,KVP
END IF
290 CONTINUE
C end of vertical axis plotting : start of taxon plotting
330 XAXIS=0.0
IF(INSSUM.EQ.1)THEN
NTP=NTP+1
NTXP(NTP)=1
END IF
C main taxon loop for plotting
DO 420 K=I,NTP
IF(K.NE.NTP.AND.LSWIT(NTXP(K)).EQ.0)GO TO 420
IF(K.EQ.NTP.AND.INSSUM.EQ.1)THEN
   DO 340 I=I,NLEVS
   IF(DS.EQ.0.)THEN
      WRITE(6,'(G***ERROR. SCALE FOR SUM IS 0.'')')
      STOP
   END IF
   XPER(I)=SUMV(I)/DS
   340 CONTINUE
ELSE
   DO 350 I=I,NLEVS
   IF(DX.EQ.0.)THEN
      WRITE(6,'(G***ERROR. SCALE FOR X-AXIS IS 0.'')')
      STOP
   END IF
   XPER(I)=(MAT(NTXP(K),I))/DX
   350 CONTINUE
END IF
PMAX=0.
C find the largest value for this taxon
DO 360 I=I,NLEVS
   IF(PMAX.LT.XPER(I))PMAX=XPER(I)
   360 CONTINUE
XAXC=MAX((PMAX+.I),XDMIN)
ANLV=AY
XAX=XAXC
IF((XAXIS+XAXC).LT.PAPL)GO TO 370
C else finish page
IXPOS=INT((PAPL+I)*IRES)
IF(LCOM.EQ.1)PAPL=PAPL+I
C number the page of the diagram
IYPOS=7*IRES
WRITE(10,' (2(I6,1X),''M '',I4,'' H ('',I2,'') S'')')IXPOS, IYPOS, IHIT, LPAG
C draw a box around this taxon
XXAXIS=NINT(XAXIS*IRES)
KAY=NINT(AY*IRES)
C zone boundaries, if any
IF(INSZON.GT.0)THEN
   CALL ZONE(INSZON,KZON, ITZON, XAXIS,SY, DY, YLEV)
END IF
WRITE(10,' (''0 0 M '',I6,'' 0 D '',2(I6,1X),'' D 0 '',I6 : ,'' D showpage'''))XXAXIS,XXAXIS,KAY,KAY
LPAG=LPAG+1
WRITE(10,' (''%%Page: ? '',I3)')LPAG
WRITE(10,' (''INI'')')
XAXIS=0.0
370 CONTINUE
XXAXIS=NINT(XAXIS*IRES)
KANLV=NINT(ANLV*IRES)
WRITE(10,'(2(I6,1X),'', '' M''')')XXAXIS,KAXIS,KAXIS,KAXIS,KAXIS,
IPLAN(I)=ICHAR(' ')
C make x-axis and labels
IF(K.EQ.NTP.AND.INSSUM.EQ.1)THEN
   CALL HORAX(XAXIS,0.,XAX,DS)
ELSE
   CALL HORAX(XAXIS,0.,XAX,DX)
END IF
C draw the curve for this taxon
C part 1 - setup for XP and X1OPER vectors
DO 380 I=I,NLEVS
   XF(I)=XXAXIS+XPER(I)
   X1OPER(I)=XPER(I)*IEXAG
380 CONTINUE
C move to top of curve
XXAXIS=NINT(XAXIS*IRES)
KANLV=NINT(ANLV*IRES)
WRITE(10, '(M') XXAXIS, XAXIS, KANLV
YLEV(NLEVS+1)=-DYN
IF(IBAR.EQ.0) THEN
   DO 390 I=1,NLEVS
      XPL(I)=XP(I)
      XIOP(I)=XPER(I)*IEXAG+XAXIS
      YLEVL(I)=YLEV(I)
   390 CONTINUE
   NBARS=NLEVS
ELSE
   DO 400 I=1,NBARS,2
      JJ=INT(I/2.+0.5)
      XPL(I)=XP(JJ)
      XPL(I+I)=XPL(I)
      XIOP(I)=XPER(JJ)*IEXAG+XAXIS
      XIOP(I+I)=XIOP(I)
      YLEVL(I)=YLEV(JJ)
      YLEVL(I+I)=YLEV(JJ+I)
   400 CONTINUE
   XPL(NBARS+I)=0.0
   XPL(NBARS+2)=I.0
   XIOP(NBARS+I)=0.0
   XIOP(NBARS+2)=I.0
   YLEVL(NBARS+I)=SY
   YLEVL(NBARS+2)=DY
   XAXD=XAXIS+XAXC
   NVECT=NBARS
C set clipping path, screen under the exaggeration curve, screen under
C the basic curve, undo clipping path
CALL CLIP(XAXIS, XAXD)
CALL SCREEN(X10P, YLEVL, NVECT, 0.95, XAXIS, AY)
CALL SCREEN(XPL, YLEVL, NVECT, 0.0, XAXIS, AY)
WRITE(10, '('--initclip--'))
XAXI=XAXC
KXAXIS=NINT(XAXIS*IRES)
KAY=NINT(AY*IRES)
IYPOS=-IRES/15
WRITE(10, '(2(I6,1X),''M'',2(I6,1X),''D'')): KXAXIS, KAY
IF(K.EQ.NTP.AND.INSSUM.EQ.I) THEN
   NMT=SUMNAM
ELSE
   NMT=NAME(NTXP(K))
END IF
DO 410 I=1, 17
   NM(I)=ICHAR(NMT(I:I))
   XTAX=XAXIS+.I4
   AYY=AY+.I4
   AYYY=AYY+.42
   XPOS=XTAX
   YPOS=AYY-.04
   DHITE=CHITE
   IF(XAXC.LT.0.5) DHITE=CHITE*0.9
   IF(XAXC.GT.1.0) DHITE=CHITE*1.1
   XPOS=XPOS+.4*XAXC-.08
C species names
CALL SYMBOL(XPOS, YPOS, DHITE, NM, 45., +17)
WRITE(6, '('--..............--',I17,',''PLOTTED'')') NAME(NTXP(K))
   XAXI=XAXIS+XAXC
420 CONTINUE
C end of main taxon loop for plotting
IXPOS=INT((PAPL+I)*IRES)
IYPOS=IRES*7
WRITE(10, '(2(I6,1X),''M'',I4,'' H ('',I2,','') S''))
   : IXPOS, IYPOS, IHIT, LPAG
   KKAXIS=NINT(XAXIS*IRES)
KAY=NINT(AY*IRES)
WRITE(10,'(0 0 M ',',16,, 0 D ',',2(I6,1X),',', D 0 ',',16,, D'))
: KXAXIS,KXAXIS,KAY,KAY
C zone if needed
IF(INSZON.GT.0) THEN
  CALL ZONE(INSZON,KZON,ITZON,XAXIS,SY,DY,YLEV)
END IF
C right-end level-bound comments, if any
IF(INSLIN.GT.0) THEN
  LCOM=0
  DO 425 I=1,INSLIN
    IF(ICOM(I).GT.2) THEN
      LCOM=I
      XPOS=XAXIS+.05
      DHITE=CHITE/I.66666666
      IF(ICOM(I).EQ.3) THEN
        YPOS=(-IDD(I)-SY)/DY-CHITE*0.5
      ELSE
        YPOS=(-IDD(I)-SY)/DY+CHITE*0.1
      END IF
      DO 424 III=1,12
        LABIL(III)=ICHAR(DCOM(I,III))
      END IF
      CALL SYMBOL(XPOS,YPOS,DHITE,LABIL,0.,12)
      IF(ICOM(I).EQ.4) THEN
        YPOS=(-IDD(I)-SY)/DY-CHITE
        DO 423 III=13,24
          LABIL(III-12)=ICHAR(DCOM(I,III))
        END IF
      END IF
      END IF
  END IF
  424 CONTINUE
  CALL SYMBOL(XPOS,YPOS,DHITE,LABIL,0.,12)
  END IF
  425 CONTINUE
  IF(LCOM.EQ.I)XAXIS=XAXIS+.9
END IF
C do CONISS if needed
XAXIS=XAXIS+.1
KXAXIS=NINT(XAXIS*IRES)
WRITE(10,'(16,, 0 M')')KXAXIS
IF(ICTL.GT.0) CALL CONCTL
WRITE(10,'(showpage)')
C the following file-terminator may not be required for all PostScript installations.
WRITE(10,'(AI)')CHAR(4)
WRITE(6,'(showpage)')
STOP
C error messages
430 WRITE(7,'(***INPUT ERROR. BAD VALUE(S) ON FOURTH CARD.'')')
WRITE(6,'(***INPUT ERROR. BAD VALUE(S) ON FOURTH CARD.'')')
STOP
440 WRITE(7,'(***DATA ERROR. TAXON IN DATA NOT IN TAXA LIST.'',/,''
: 'BAD NUMBER IS ',',I4,'' AT LEVEL',,I5 ')')NTX(I),LC
WRITE(6,'(***DATA ERROR. TAXON IN DATA NOT IN TAXA LIST.'',/,''
: 'BAD NUMBER IS ',',I4,'' AT LEVEL',,I5 ')')NTX(I),LC
STOP
9000 WRITE(6,'(^G***BAD CHARACTER IN COMMENT LINE. ABORT.'')')
STOP
9010 WRITE(6,'(^G***BAD CHARACTER IN CARD 4. ABORT.'')')
STOP
9020 WRITE(6,'(^G***BAD CHARACTER ON CARD 5. ABORT.'')')
STOP
9030 WRITE(6,'(^G***EXPECTING ZONE BOUNDARY POSITIONS AND STYLES.'',/
: //,''PLEASE VERIFY INSZON AND KZON/ITZON CARDS CARDS.'',/,''
: 'ABORT.'')')
STOP
9040 WRITE(6,'(^G***EXPECTING LIST OF TAXA FOR CONISS. ABORT.'')')
STOP
Plotting stratigraphic data on a PostScript device

9050 WRITE(6,'("'**BAD CHARACTER IN DX, RX, AY, SY, DYM, OR DYN '':,'';FIELD(S). ABORT.'')')
STOP
9060 WRITE(6,'("'**EXPECTING NUMBER OF TAXA TO PLOT. ABORT.'')')
STOP
9070 WRITE(6,'("'**EXPECTING LIST OF TAXA TO PLOT. ABORT.'')')
STOP
9080 WRITE(6,'("'**MISSING LIST OF TAXON NUMBERS AND NAMES.'',/,,':,'';OR NUMBER OF TAXA TOO LARGE.'',/,,':,'';ABORT.'')')
STOP
9090 WRITE(6,'("'**EXPECTING LIST OF TAXA IN SUM. ABORT.'')')
STOP
9100 IF(J.GT.I)THEN
  WRITE(6,'("'**BAD CHARACTER IN DATA FIELD. LAST LEVEL READ: '':,'';I4) 'Lv(J-I)
ELSE
  WRITE(6,'("'**BAD CHARACTER IN FIRST LEVEL. ABORT.'')')
END IF
STOP
END

SUBROUTINE CONCTL
COMMON/PERC/MAT (251,250)
COMMON/CTL1/LPAG, NTAX, PAPL, MSWIT(250)
COMMON/BOTT/MXDP
COMMON/DEND 1/AY, CCX
COMMON/CTL3/NN, XAXIS
C subroutine controls the cluster analysis and dendrogram plotting
REAL MAT
NLEVS=NN+1
N=0
DO 20 I=I,NTAX
  IF (MSWIT(I).EQ. I) THEN
    N=N+I
    DO 10 J=I,NLEVS
      MAT(N, J)=MAT(I, J)
  10 CONTINUE
  END IF
20 CONTINUE
IF (XAXIS.GT. (PAPL-CCX)) THEN
  WRITE (10, '( 'showpage')')
  LPAG=LPAG+ 1
  WRITE(10,'( %Page: ? ',I3)')LPAG
  WRITE (10, '( 'INI')')
  XAXIS=0.0
END IF
CALL CONISS
CALL TREED
RETURN
END

SUBROUTINE SYMBOL(XIN, YIN, HT, STRIN, ANG, NCHIN)
COMMON/LASER/IRES
C builds a text string one character at a time, checking for font changes
C and PostScript reserved characters.
REAL HT
CHARACTER*80 STRNG
INTEGER STRIN(*), POS, POSMI, POS3
CHARACTER*I CSTR(100)
C initialise lots of stuff
POS=1
IX=NINT(XIN*IRES)
IY=NINT(YIN*IRES)
NCH=NCHIN
WRITE(10, '(2(I6,IX), ''M''')IX, IY
C round angle to integer - good to 1 degree
INTANG=NINT(ANG)
COSANG=COS(FLOAT(INTANG) * 0.017453292519)
\[ \sin \theta = \sin \left( \text{FLOAT}(\text{INTANG}) \times 0.017453292519 \right) \]
\[ \text{ICH} = \text{NINT}(\text{HT} \times 1.666666666 \times \text{IRES}) \]

C set char angle
IF(\text{INTANG} .NE. 0) THEN
  WRITE(10,'(I3,\' RS\'')) \text{INTANG}
END IF

J=1
C find correct font and set height
IF(\text{STRIN(J) \ EQ. 126}) THEN
  J=2
  WRITE(10,'(I3,\' HI \'')) \text{ICH}
ELSE
  WRITE(10,'(I3,\' H \'')) \text{ICH}
END IF

C plot a string of characters
C transfer chars into holding area
DO 10 I=J, \text{NCH}
  \text{CSTR(I)} = \text{CHAR(STRIN(I))}
  10 CONTINUE
C output "(string) S ", escape ( ) \theta
  \text{STRNG(POS:POS)}='( '
  POS=POS+1
  DO 20 I=J, \text{NCH}
  20 CONTINUE

C reserved character: ( \theta
  \text{STRNG(POS:POS3)}='\050'
C reserved character: ) \theta
  \text{STRNG(POS:POS3)}='\051'
C reserved character: % \theta
  \text{STRNG(POS:POS3)}='\045'
C reserved character: / \theta
  \text{STRNG(POS:POS3)}='\057'
C reserved character: < \theta
  \text{STRNG(POS:POS3)}='\074'
C reserved character: > \theta
  \text{STRNG(POS:POS3)}='\076'
C reserved character: \ \theta
  \text{STRNG(POS:POS3)}='\133'
C reserved character: \ \theta
  \text{STRNG(POS:POS3)}='\134'
Plotting stratigraphic data on a PostScript device

```
STRING(POS:POS3)='\135'
ELSE
STRING(POS:POS)=CSTR(I)
POS=POS-3
ENDIF
POS=POS+4
20 CONTINUE
POS3=POS+3
STRING(POS:POS3)=') S '
POS=POS+4
IF(INTANG.NE.0)THEN
POS3=POS+3
C reset angle to 0
STRING(POS:POS3)=') RE '
POS=POS+4
ENDIF
POSM1=POS-1
C output the string
WRITE(10,'(A)')STRING(I:POSM1)
RETURN
END
SUBROUTINE SCREEN(X,Y,N,G,B,A)
COMMON/LASER/IRES
the gray-level and paint below the curve
DIMENSION X(*),Y(*),IX(6),IY(6)
WRITE(10,'(''newpath '')')
IB=NINT(B*IRES)
IA=NINT(A*IRES)
WRITE(10,'(2(I6,1X),''M'')')IB, IA
DO 30 K=1,N,5
   DO 10 I=K,K+4
      J=I-K+I
      IF(I.GT.N) THEN
         DO I0 M=J,5
            IY(J) =0
            IX(J) =0
            10 CONTINUE
      END IF
      IY(J)=NINT(IRES*(Y(I)-Y(N+I))/Y(N+2))
      IX(J)=NINT(IRES*(X(I)-X(N+I)) /X(N+2))
   20 CONTINUE
   WRITE(10,'(250(2(I6,1X),''L '')') (IX(J),IY(J),J=I,5)
30 CONTINUE
WRITE(10,'(I6,'' 0 L '',2(I6,1X),''L closepath'')')IB,IB,IA
WRITE(10,'(F4.2,'' setgray gsave fill RE 0 setgray stroke'')')G
RETURN
END
SUBROUTINE CLIP (X1, X2)
COMMON/LASER/IRES
COMMON / DEND 1/AY, CCX
C makes a vertical rectangle the clipping path, for 1 taxon at a time
I=NINT (XI* IRES)
J=NINT (X2*IRES)
IYPOS=IRES*AY
WRITE(10,' (I6, '' 0 M '',2(I6,1X),''L '',2(I6,1X),''L '')') I, I, IYPOS, J, IYPOS
WRITE(10,' (I6, '' 0 L '',I6, '' 0 L clip'')')J, I
RETURN
END
SUBROUTINE HORAX(X,Y,AL, XIN)
COMMON/LASER/IRES
COMMON/CHRT/CHITE
C make x-axis starting at point X,Y; length is AL; XIN is scale(inch);
IX=NINT(X*IRES)
```
C moves cursor to start point
WRITE(10,' ("initclip newpath ",2(I6,1X),"M")' ) IX, IY
C make first 'tic'
  IYL=IY-IRES/5
  IXL=IX-IRES/30
  IHIT=NINT(CHIT/2* IRES)
  WRITE(10,' ("VLTIC ",2(I6,1X),"M ",I4," H (0) S")' ) IXL, IYL, IHIT
  IPT=IX
C locate next 'tic'
I=0
10 CONTINUE
  IX=IX+IRES/10
  I=I+1
  C check if room for another 'tic'
  IF(IX.GT.IAL) GO TO 30
  C make 'tic' and check if room for another label
  WRITE(10,' (2(I6,IX),"M")' ) IX, IY
  IF(IX.GT.(IAL-IRES/20)) GO TO 30
  IYL=IY-IRES/5
  IXL=IX-IRES/30
  IF (IX.GT. (IPT+NINT (IRES/4.3)) .AND. IX.LT. (IAL-IRES/9)) THEN
    XNMB= IX/10.
    IPT=IX
    IF(XNMB.GE.-9. AND. XNMB.LE.9.) THEN
      WRITE(10,' ("VLTIC ",2(I6,1X),"M ",I4," H (",F4.2," S")") IXL, IYL, IHIT, XNMB
    ELSE IF(XNMB.GT.9. AND. XNMB.LE.99.) THEN
      WRITE(10,' ("VLTIC ",2(I6,1X),"M ",I4," H (",I2," S")") IXL, IYL, IHIT, NMB
    ELSE IF(XNMB.GE.100. AND. XNMB.LE.999.) THEN
      WRITE(10,' ("VLTIC ",2(I6,1X),"M ",I4," H (",I3," S")") IXL, IYL, IHIT, NMB
    ELSE
      WRITE(10,' ("VSTIC ")')
    END IF
  ELSE
    WRITE(10,' ("VSTIC ")')
  END IF
  GO TO 10
30 RETURN
END
SUBROUTINE CONISS
COMMON/LASER/IRES
COMMON/DIST/D(3125)
COMMON/CHRT/CHIT/2
COMMON/PERC/MAT(251,250)
COMMON/BOTT/MXD
COMMON/CTL2/NVARS, ICTL, DYN, YLEV(252)
COMMON/CTL3/NN, XAXIS
COMMON/DEND2/ILEV(250), JLEV(250), BRANCH (250)
COMMON/NAM/NAME(250)
COMMON/NT/NTIM
C calculates and plots constrained clusters.
DIMENSION Z(250,150), N(250,150)
REAL MAT
EQUIVALENCE (MAT, Z,N)
C ICTL is control code for data transformations
C convert to proportions
NLEVS=NN+1
IX=NINT(XAXIS*IRES+IRES/30)
IY=-NINT(IRES/2.6)
IHIT=NINT(CHITE*IRES)
NTIM=NVARS*NLEVS
WRITE(6,'(''.............PERFORMING CLUSTER ANALYSIS'')')
DO 10 J=1,NLEVS
   NAME(J)=J
DO 10 I=1,NVARS
   MAT(I,J)=0.01*MAT(I,J)
10 CONTINUE
C square-root transformation
IF(ICTL.EQ.1)THEN
   DO 20 J=1,NLEVS
      DO 20 I=1,NVARS
         MAT(I,J)=SQRT(MAT(I,J))
20 CONTINUE
   WRITE (6,'(9X,''Data transformed by square roots'')')
   WRITE(10,'(2(I6,1X),''M '',I4,'' H (Total sum of squares) S'')')IX, IY, IHIT
ELSE IF(ICTL.EQ.2)THEN
C standardization
   XLEVS=REAL(NLEVS)
   XLEVS1=XLEVS-1.0
   DO 50 I=1,NVARS
      SX=0.0
      SX2=0.0
   DO 30 J=1,NLEVS
      XIJ=MAT(I,J)
      SX=SX+XIJ
      SX2=SX2+XIJ*XIJ
30 CONTINUE
   IF(SX.GT.0.0)THEN
      XBAR=SX/XLEVS
      SD=SQRT((SX2-SX*SX/XLEVS)/XLEVS1)
      DO 40 J=1,NLEVS
         MAT(I,J)=(MAT(I,J)-XBAR)/SD
40 CONTINUE
   END IF
50 CONTINUE
   WRITE(6,'(9X,''Data standardized'')')
   WRITE(10,'(2(I6,1X),''M '',I4,'' H (Total standardized',
      '' dispersion) S'')')IX, IY, IHIT
ELSE IF(ICTL.EQ.3)THEN
C normalization
   DO 70 J=1,NLEVS
      SX2=0.0
   DO 60 I=1,NVARS
      XIJ=MAT(I,J)
      SX2=SX2+XIJ*XIJ
60 CONTINUE
   SQ=SQRT(SX2)
   DO 70 I=1,NVARS
      MAT(I,J)=MAT(I,J)/SX
70 CONTINUE
   WRITE(6,'(9X,''Data normalized'')')
   WRITE(10,'(2(I6,1X),''M '',I4,'' H (Total normalized',
      '' dispersion) S'')')IX, IY, IHIT
ELSE
C no transformation
   WRITE(6,'(9X,''No data transformations performed'')')
   WRITE(10,'(2(I6,1X),''M '',I4,'' H (Total dispersion) S'')')
   IX, IY, IHIT
END IF
C generate distance matrix
ID=0
DO 90 I=2,NLEVS
DO 90 J=1,I-1
DSQD=0.0
DO 80 K=I,NVARS
   DSQD=DSQD+(MAT(K,I)-MAT(K,J))**2
80 CONTINUE
ID=ID+1
D(ID)=DSQD
90 CONTINUE
C cluster
MXDP=NINT(DYN)
CALL CCLUS(NN,Z(I,1),Z(I,5),N(I,6),N(I,7))
C write results
   : /T5, A, T15, A, 2X, 4(8X, F6.4)/)') 'MEAN' ,'WITHIN-', 'WITHIN-',
   : 'CLUSTERS', 'INCREASE IN', 'TOTAL', 'CLUSTER', 'CLUSTER', 'STAGE',
   : 'MERGED', 'DISPERSION', 'DISPERSION', 'DISPERSION', 'DISPERSION'
DO 100 I=1,NN
   WRITE(7,'(5X, I4,2X,215,2X, 4(8X, F6.4))')I,N(I,6),N(I,7),(Z(I,J),
   : J=1,4)
ILEV(I)=YLEV(N(I,6))
JLEV(I)=YLEV(N(I,7))
BRANCH(I)=Z(I,2)
100 CONTINUE
RETURN
END
SUBROUTINE CCLUS(NN,ES,ESS,NAM,EZ)
COMMON/DIST/D(31125)
COMMON/NAM/NAME(250)
COMMON/NT/NTM
COMMON/SORT/NCLUS(250), IP, IQ, NP, NQ, DSHORT
C constrained cluster analysis
DIMENSION ES(250,4)
DIMENSION NAMEP(250),NAMEQ(250),ESS(250)
C initialize arrays and variables
WRITE(6,'(('' .......... CLUSTERING''))')
IF(NTM.GT.1000)WRITE(6,'(10X,'' (THIS MAY TAKE SOME TIME....)''))
DO 10 I=1,150
   NCLUS(I)=1
   ESS(I)=0.0
10 CONTINUE
MSIZ=NN
E=0.0
C begin clustering
DO 70 ITER=1,MSIZ
C find most similar clusters
DSHORT=D(1)
   IP=1
   ID=1
DO 20 N=2,MSIZ
   IF(D(ID).LT.DSHORT)THEN
      DSHORT=D(ID)
      IP=N
   END IF
20 CONTINUE
IQ=IP+1
NAMP=NAME(IP)
       NAMQ=NAME(IQ)
       NAMEP(ITER)=NAMP
       NAMEQ(ITER)=NAMQ
       NP=NCLUS(NAMP)
       NQ=NCLUS(NAMQ)
C calculate merge data
DE=0.5*DSHORT
E=E+DE
ESS(NAMP)=ESS(NAMP)+ESS(NAMQ)+DE
ES(ITER,1)=DE
Plotting stratigraphic data on a PostScript device

Es(ITER,2)=E
Es(ITER,3)=ESS(NAMP)
Es(ITER,4)=ESS(NAMP)/REAL(NP+NQ)

C update distance matrix
IIP=(IP-1)*(IP-2)/2
IIQ=(IQ-1)*(IQ-2)/2
DO 30 J=1,IP-1
   ID=IIP+J
   D(ID)=UPDATE(J)
   DO 30 I=IQ,MSIZ
      IJ=J+(I-I)*(I-2)/2
      IIJ=IJ+I-1
      D(IJ)=D(IIJ)
30 CONTINUE
IR=IIQ-IQ+2
DO 40 I=IQ,MSIZ
   IR=IR+I-2
   ID=IR+IP
   D(ID)=UPDATE(I+1)
40 CONTINUE
DO 50 J=IQ,MSIZ-I
   DO 50 I=J+I,MSIZ
      IJ=J+(I-I)*(I-2)/2
      IIJ=IJ+I
      D(IJ)=D(IIJ)
50 CONTINUE
DO 60 M=IQ,MSIZ
   NAME(M)=NAME(M+1)
60 CONTINUE
NCLUS(NAMP)=NP+NQ
MSIZ=MSIZ-I
70 CONTINUE
RETURN
END

FUNCTION UPDATE(IR)
COMMON/NAM/NAME(250)
COMMON/SORT/NCLUS(250), IP, IQ, NP, NQ, DSHORT
C this function solves the update equation
NR=NCLUS(NAME(IR))
UPDATE=((NR+NP)*D2(IR, IP)+(NR+NQ)*D2(IR, IQ)-NR*DSHORT)/(NR+NP+NQ)
RETURN
END

FUNCTION D2(I,J)
COMMON/DIST/D(31125)
C this function locates value in lower semi-matrix
IF (I .GT.J) THEN
   ID=(I-1)*(I-2)/2+J
ELSE
   ID=(J-1)*(J-2)/2+I
END IF
D2=D(ID)
RETURN
END

SUBROUTINE TREED
COMMON/LASER/IRE
COMMON/CHRT/CHITE
COMMON/BOTT/HXDP
COMMON/DEND1/AY, CCX
COMMON/CTL3/NN, XAXIS
COMMON/DEND2/ILEV(250), JLEV(250), BRANCH(250)
C draw dendrogram from cluster analysis
DIMENSION IBASE(250), JBASE(250), MBRAN(250)
WRITE(6,'(\'.............DRAWING TREE\')')
C setup
FLAST=0.0
FIRST=0.0
FBIG=10000.0
DO 10 I=1,NN
   IF (BRANCH(I).GT.FBIG) FBIG=BRANCH(I)
10 CONTINUE
FSCAL=FBIG/CCX
DO 20 I=1,NN
   IBASE(I)=0
   JBASE(I)=0
   ILEV(I)=NINT((AY-ILEV(I)*IRES*A)/MXDP)
   JLEV(I)=NINT((AY-JLEV(I)*IRES*A)/MXDP)
   MBRAN(I)=NINT((BRANCH(I)/FSCAL)*IRES)
20 CONTINUE
C draw tree
ISH=NINT(AY*IRES)
IXX=NINT(XAXIS*IRES)
WRITE(10,' (2(I6,1X),''translate'')')IXX,ISH
DO 30 I=1,NN
   WRITE(10,' (2(I6,1X),''M'',3(2(I6,1X),''D''))')IBASE(I),
     ILEV(I),MBRAN(I),ILEV(I),MBRAN(I),JLEV(I),JBASE(I)
   DO J=I+1,NN
      IF (ILEV(J).EQ.ILEV(I)) THEN
         IBASE(J)=MBRAN(I)
         ILEV(J)=(ILEV(I)+JLEV(I))/2
      END IF
      IF (JLEV(J).EQ.ILEV(I)) THEN
         JLEV(J)=(ILEV(I)+JLEV(I))/2
         JBASE(J)=MBRAN(I)
      END IF
   END DO
30 CONTINUE
C write scale
ISH=-ISH
IXX=-IXX
WRITE(10,' (2(I6,1X),''translate'')')IXX,ISH
CALL HORAX(XAXIS,-0.05,CCX,FSCAL)
IX=NINT(XAXIS*IRES+IRES/30)
IY=NINT(AY*IRES+IRES/10)
IHIT=NINT(CHITI*IRES*I.66667)
WRITE(10,' (2(I6,1X),''M'',I4,''H (Constrained cluster analysis) S'')')IX, IY, IHIT
C end plot
RETURN
END
SUBROUTINE INIT
COMMON/LASER/IRES
C initialise postscript file
ITR=-NINT(IRES*8.5)
WRITE(6,(''....INITIALISING POSTSCRIPT FILE''))
WRITE(10,(''%!CANPLOT output file''))
WRITE(10,(''/INIT {72 '',I5,''} div dup scale 90 rotate''))IRES
WRITE(10,(''/0 '',I7,'' translate '',2(I5,1X),''translate'',
: '' 0 0 moveto'' ) )ITR,IRES,IRES
WRITE(10,(''/1 setlinewidth 1 setlinejoin 1 setlinecap} def''))
WRITE(10,(''/SF /Times-Roman findfont def'' ) )
WRITE(10,(''/SI /Times-Italic findfont def'') )
WRITE(10,(''/M /moveto load def''))
WRITE(10,(''/RM /rmoveto load def''))
WRITE(10,(''/D {lineto currentpoint stroke moveto} def''))
WRITE(10,(''/L /lineto load def''))
WRITE(10,(''/O (lineto currentpoint stroke moveto) def''))
WRITE(10,(''/W (currentpoint translate) def''))
WRITE(10,(''/H (SF exch scalefont setfont) def''))
WRITE(10,(''/HI [SI exch scalefont setfont] def''))
IHIT=IRES/30
WRITE(10,(''/HTIC {',I5,','' 0 rlineto stroke} def''))IHIT
IHIT=IRES/15
WRITE(10,(''/LTIC {',I5,','' 0 rlineto stroke} def''))IHIT
IHIT=IRES/30
SUBROUTINE ZONE(INSZON, KZON, ITZON, XAXIS, SY, DY, YLEV)
COMMON/LASER/IRES
C manually places zone boundaries using a style library
DIMENSION KZON(*), ITZON(*), YLEV(*)
DO 10 I=1, INSZON
   IX=4
   IY=NINT(IRES*(((YLEV(KZON(I))+YLEV(KZON(I)+1))/2)) - SY)/DY)
   WRITE(10, '('(2(I4,1X),''M '')') IX, IY
   IX=NINT(XAXIS*IRES)-4
C select boundary style from library
   GO TO (10, 20, 30, 40, 50, 60, 70, 80, 90) ITZON(I)
10 IT=2
   G=1.0
   GO TO 100
20 IT=6
   G=1.0
   GO TO 100
30 IT=10
   G=1.0
   GO TO 100
40 IT=2
   G=0.0
   GO TO 100
50 IT=6
   G=0.0
   GO TO 100
60 IT=10
   G=0.0
   GO TO 100
70 IT=2
   G=0.5
   GO TO 100
80 IT=6
   G=0.5
   GO TO 100
90 IT=10
   G=0.5
100 WRITE(10, '('(I2, '' setlinewidth '', F3.1, '' setgray'', 2(iX, I6),
   : '' D'')') IT, G, IX, IY
   WRITE(10, '(''1 setlinewidth 0 setgray'')')
CONTINUE
END
SUBROUTINE RVALIN
COMMON/RV/RVAL(250), YVAL(250)
COMMON/CTL/LPAG, NTAX, PAPL, MSWIT(250)
DO 5 I=1, 250
   RVAL(I)=1.0
   YVAL(I)=0.0
5 CONTINUE
READ(5, '('(I3)')') NLIN
NLIN=INT(NLIN/4.-.1)+1
DO 40 I=1, NLIN
   READ(5, '('(4(I3, 1X, 2(F6.3, 1X)))')') N1, R1, Y1, N2, R2, Y2, N3, R3, Y3, N4,
   : R4, Y4
   WRITE(10, '(''(2(I2, ''M '')') N1, R1, Y1, N2, R2, Y2, N3, R3, Y3, N4,
   : R4, Y4
   END
APPENDIX 2

Input File for Figure 2

filename: gren02.mac

Grenadier Pond ON core 0 Macrofossil Summary

120 Ii 1 0 0 0 1 0 0 0 0 0 0 10 .3 100.0
000 2.0 0.13 300 seed sum
50.0 4.5 -790.0-1.0 -790.0
19
84112108 20 10 39 30 75 21 42 99113 94 66 67 90 93106119
0 1-Abies"N
2-Abies"S
3-Abies"total
4-Alnus"sp."B
5-Alnus"sp. S
6-Alnus crispa"B
7-Alnus crispa"S
8-Alnus rugosa"B
9-Alnus rugosa"S
10-Alnus
11-Aster"S
12-Betula"sp. B
13-Betula sp."S
14-Betula lutea"B
15-Betula lutea"S
16-Betula pap."P
17-Betula pap."S
18-Betula pumila"B
19-Betula pumila"S
21-Bidens
22-Boehmeria
23-Brasenia
24-Carex"sp
25-Carex australis
26-Carex comosa
27-Carex pseudocy."S
28-Carex rostrata
29-Carex atpata
30-Carex
31-Cephalanthus
32-Ceratophyllum
33-Chenopodium
34-Conifer
35-Cornus stolonif.36Cyperaceae
37-Cyperus"sp
38-Cyperus engel."B
39-Decodon
41-Delichium
42-Eleocharis pal.
43-Eleocharis acic.44Ericaceae
45-Epipodium
46-Eriophorum
47-Eupatorium
48-Glyceria
49Gramineae
50-Heteranthera
51-Hippuris
52-Hypericum
53-Juniperus"N
54-Juniperus"S
55-Larix"N
56-Larix"S
57-Lax"total
58-Leersia
59ml examined
60-Lycopus amer.
61-Lyssimachia
62-Mentha
63-Menyanthes
64-Monarda
65-Myrica gale
66-Myriophyllum ex.67-Najas flexilis
68****nothing****
69-Nuphar
70-Nymphphaea
71-Panicum
72-Picea"N
73-Picea"S
74-Picea"total
75-Picea
76-Pinus banks."S
77-Pinus banks."N
78-Pinus bankian"N
79-Pinus resinosa"N
80-Pinus resin."S
81-Pinus resinos"T
82-Pinus strobos"N
83-Pinus strobos"S
84-Pinus strobos
85-Polygonum amph.86-Polygonum lapath 87-Polygonum"sp.
88-Eupotamogeton
89-Colpopotamogeton
90-Potamogeton
91-Potentilla
92-Ranunculus"sp
93-Ranunculus"Bat
94-Roquelle
95-Rorippa
96-Rumex
97-Sagittaria"sp
98-Sagittaria lat.
99-Sagittaria
100-Sambucus
101-Scirpus atrov 102-Scirpus validus 103-Scutellaria
104-Succ"N
105-Soldago
106-Sparagamium chl.107-Sparagamium eury.108-Thuja"N
109-Thuja"S
110-Tsuga
111-Tsuga"S
112-Tsuga
113-Pypha
114-Umbelliferae
115-Urtica
116-Utricularia
117-Verbena hastata
118-Viola
119-Zanichellia
120unknown
59
59

DAT

0

1 13 4 17 5 20 9 30 1 39 1 45 1 59 998 67 9 71 1 82 2 0
1 14 2113 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1
44 12 1 13 25 16 4 17 4 20 34 24 1 26 1 30 2 42 2 50 2 0
44 55 3 591035 67 21 75 1 79 2 81 2 82 9 84 9 88 1 90 1 0
44 47 1 98 3 99 4102 3108 5110 1112 1113 1120 2 0 0 0
95 5 6 8 14 9 35 10 55 16 2 13 15 14 1 15 5 18 17 18 39 0
95 20 80 21 16 26 7 28 2 30 9 31 2 33 1 36 12 37 1 42 33 0
95 47 5 49 1 50 2 53 1 55 4 57 4 591654 60 1 62 1 63 1 0
95 64 1 67 75 70 1 71 5 75 1 79 4 81 4 82 44 83 3 84 47 0
95 66 4 87 3 88 4 90 4 91 2 95 9 97 18 98 22 99 40100 2 0
95102 154104 3106 4107 3108 564109 3110 14111 4112 18113 1 0
95114 3115 6117 2118 2119 42120 18 0 0 0 0 0 0 0 0 0
185 9 1 10 1 17 4 20 4 24 1 30 1 59 609 67 5 82 2 84 3 0
185 93 1 97 1 99 1108 20110 1112 1 0 0 0 0 0 0 0 0 0 0 0
215 17 1 20 1 55 1 57 0 591411 68 5 82 2 84 2 0 0 0 0 0
300 12 8 13 16 14 1 15 12 16 21 17 56 20 114 21 1 23 1 24 3 0
300 26 3 30 6 37 1 39 1 42 1 47 1 50 1 55 1 57 1 593350 0
300 66 3 672319 68 6 69 2 77 1 78 1 79 2 81 2 82 217 83 4 0
300 84 21 88 20 90 20 97 3 98 1 94 4102 3108 17113 2119 1 0
Plotting stratigraphic data on a PostScript device

300120 3
480 9 1 10 1 16 1 17 18 20 19 21 1 24 5 26 2 28 17 30 24 0
480 33 1 42 193 59 994 60 1 63 1 66 4 67 172 68 2 69 1 82 16 0
480 83 4 84 20 85 7 88 7 90 7 97 1 99 1102 43113 2119 3 0
480120 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0
550 4 1 0 1 9 5 10 6 12 3 13 16 15 3 17 1 5 20 37 21 43 0
550 22 1 24 1 51 26 63 28 1 30 115 36 15 39 98 42 101 47 5 48 1 0
550 592639 60 2 61 2 62 66 66 2 67 156 69 1 75 48 82 12 83 3 0
550 84 15 86 6 87 3 88 8 90 11 91 2 93 1 94 13 95 1 97 4 0
550 98 2 99 6102 86108 2110 2111 1112 3113 4117 32 0
550118 1120 10 0
640 9 7 10 7 12 1 13 2 15 11 17 12 20 27 21 158 24 9 26 64 0
640 28 111 30 104 32 1 36 14 38 40 39 1 42 72 592128 60 1 62 4 0
640 67 32 68 2 75 5 82 61 83 2 84 63 87 23 88 63 90 13 91 1 0
640 93 9 94 8 97 6 98 5 99 11102 199108 7109 1110 1111 1 0
640112 1113 5114 5119 200120 12 9 0 0 0 0 0 0 0 0 0 0 0
730 4 1 9 7 10 8 12 1 15 2 16 1 17 6 19 2 20 12 21 6 0
730 24 2 26 4 28 12 30 18 37 1 42 3 59 913 62 4 67 81 68 4 0
730 69 1 75 1 79 1 81 1 82 8 83 1 84 9 88 4 90 4 93 10 0
730 94 1 97 8 98 5 99 13102 2104 2107 7113 1119 4120 2 0
790 59 1

END