This folder contains
the ASCII-Data-Files: "FRA11DEL.F37", "FRA11N.EP", "FRA11NN.EP",
the handbook files (Ms-Word):"Generating SIMMAX code" and "SIMMAX TECHNICAL README",
the form "SIMMAX24.INP MUSTER",
the Mac-Programm file SIMMAX28 apl
and the ASCII Text-Tables "TAB1.TXT", "TAB2.TXT", "TAB3.TXT".

From Species Count to SIMMAX Input Code.

Uwe Pflaumann,
Geol. Paläont. Institut Universität Kiel, Olshausenstr. 40, D-24098 Kiel,
Phone: 49 431-880-2928, Fax: 49 431 880 4376,
internet: up@gpi.uni-kiel.de

How to put census data into the computer?
Normally, counts will be easily transferred into a table using spreadsheets like Excel. However, the conversion of data from the spreadsheet into desired ASCII code results in left bounded numbers and often zero valued decimals are cut. This results in difficulties for FORTRAN based input requirements.

The SIMMAX procedure is written in FORTRAN, a language sensitive to columns and lines. To generate a file in the correct format we follow a strict procedure:
The file PLA.INP is a blank form, which can be copied into the scratch memory and then copied on demand into a newly generated file with the extension .PLA, which indicates "Plankton-Foram-Count".

1KERN 16867-3 SECTION 1-1

PROBE 1005.	>150	>200	>250		>315	>500	>630
TEIL 1/	0.0	0.0	0.0	PERC	0.0	0.0	0.0
1 AEQU	.	.	.	.00	.	.	.
2 BULL	.	.	.	.00	.	.	.
3 CALI	.	.	.	.00	.	.	.
4 CONG	.	.	.	.00	.	.	.
5 CRAS	.	.	.	.00	.	.	.
6 DEHI	.	.	.	.00	.	.	.
7 DIGI	.	.	.	.00	.	.	.
8 DUTE	.	.	.	.00	.	.	.
9 FALC	.	.	.	.00	.	.	.
10 GLUT	.	.	.	.00	.	.	.
11 HEXA	.	.	.	.00	.	.	.
12 HIRS	.	.	.	.00	.	.	.
13 HUMI	.	.	.	.00	.	.	.
14 INFL	.	.	.	.00	.	.	.
15 MENA	.	.	.	.00	.	.	.
16 NITI	.	.	.	.00	.	.	.
17 OBLI	.	.	.	.00	.	.	.
18 PAC.LI	.	.	.	.00	.	.	.
19 PAC.RE	.	.	.	.00	.	.	.
20 PELA	.	.	.	.00	.	.	.
21 QUIN	.	.	.	.00	.	.	.
22 RUB.R	.	.	.	.00	.	.	.
23 RUB.H	.	.	.	.00	.	.	.
24 RUBER	.	.	.	.00	.	.	.
25 RUS.R	.	.	.	.00	.	.	.
26 RUS.W	.	.	.	.00	.	.	.
27 SACC	.	.	.	.00	.	.	.
28 SACS	.	.	.	.00	.	.	.
29 SCIT	.	.	.	.00	.	.	.
30 TUMI	.	.	.	.00	.	.	.
31 TRU.L	.	.	.	.00	.	.	.
32 TRU.R	.	.	.	.00	.	.	.
33 UNIV	.	.	.	.00	.	.	.
34 UVUL	.	.	.	.00	.	.	.
35 P/D.I	.	.	.	.00	.	.	.
36 FLEX	.	.	.	.00	.	.	.
37 TENE	.	.	.	.00	.	.	.
38 PUNC	.	.	.	.00	.	.	.
39 TRIA	.	.	.	.00	.	.	.
40 HESS	.	.	.	.00	.	.	.
41 ROND	.	.	.	.00	.	.	.
42 VIOL	.	.	.	.00	.	.	.
43 CAVE	.	.	.	.00	.	.	.
	.	.	.	.	.	.	.

"16867-3" is read as a 7-character string
1005. is read as 7-digit real number as sample depth
The next line
" TEIL 1/ 0.0 0.0 0.0 PERC 0.0 0.0 0.0"
contains the portions of the different size fractions counted and they are read as 6 times a 6-digit number at the place of 0.0, starting in column 15. If the sample was not size-fractionated, only the first column should be filled in. 16. would mean the counts refer to 1/16 of the sample.The next 43 lines contain the number of specimens counted, each line for one species. The ordering of species is constant and should not be changed. The numbers have to be inserted before the dots and the dots shall not be removed. Species with no specimens can be left blank, as it is easier to edit the data input, if you see only real numbers and blanks instead of real numbers and 0.0. The last line contains nothing and should be always be included. After correction of the data the whole file has to be edited by a global search and replace command: search for "blank dot" and replace it by "zero dot". Save the file as .pla file. In our example we call it "16867-3.PLA"
This file is the input file for the Mac program KOR.apl, the source code of is called KOR.FOR

```
C      LESEN DES KORTAP SIRIUS VERSION 16415  18-4-85
       CHARACTER*255 INFIL,OUTFIL,OUTFL1
       CHARACTER*10 NAME(43)
       CHARACTER*5 TITEL(6),FREE,PERC
        DIMENSION X(9,43),XF(9),XX(43),XXP(43),SUM(9)
          WRITE(*,'(60X,A)')'KOR','20-NOV-90','PFLAUMANN'
          WRITE(*,'(29H Please, specify INPUT-file  ,\)')
        READ(*,'(A15)')INFIL
          WRITE(*,'(29H Please, specify OUTPUT-file ,\)')
        READ(*,'(A15)')OUTFIL
        WRITE(*,'(a)')' Please, specify LINES <PL=43,GRK=7 or 5> '
       READ(*,'(I2)')N
      N1=N
       IF(N.EQ.5)N1=6
       OPEN(25,FILE=INFIL,BLANK='ZERO')
                WRITE(*,*)INFIL
       OPEN(55,FILE=OUTFIL)
      L=LENGTH(OUTFIL)
        OUTFL1=OUTFIL(1:L)//'HO'
       OPEN(56,FILE=OUTFL1)
                OUTFL1=OUTFL1(1:L)//'  OPENED'
       WRITE(*,'(1X,A)') OUTFL1
      PERC='PERC.'
      NFRAK=6
10     DO 15 J=1,NFRAK
15     SUM(J)=0.
       DO 20 I=1,N1
20     XX(I)=0.
      SUMC=0.
      SUMM=0.
         READ (25,30,ERR=140,END=140)(TITEL(I),I=1,5)
30       FORMAT(20A5)
         READ(25,'(1X,A5,f7.1)') TITEL(6),dpth
         WRITE(*,'(1X,A5,f7.1)') TITEL(6),dpth
        READ(25,40)(XF(J),J=1,NFRAK)
40       FORMAT(14X,3F6.1,6X,3F6.1)
         WRITE(*,'(3F6.1,6X,3F6.1)')(XF(j),j=1,nfrak)
       IF(N.EQ.5)THEN
          READ(25,'(A)')FREE
            READ(25,60)((X(J,I),J=1,3),DUM,(X(J,I),J=4,NFRAK),I=1,2)
           READ(25,60) (X(J,6),J=1,3),DUM,(X(J,6),J=4,NFRAK)
            READ(25,60)((X(J,I),J=1,3),DUM,(X(J,I),J=4,NFRAK),I=3,N)
         GOTO 70
      ENDIF
         READ(25,60)((X(J,I),J=1,3),DUM,(X(J,I),J=4,NFRAK),I=1,N)
60       FORMAT(14X,3F6.0,F6.2,3F6.0)
70      READ(25,'(A5)')FREE
       DO 81 J=1,NFRAK
       DO 80 I=1,N
       XX(I)=XX(I)+X(J,I)*XF(J)
      SUMC=SUMC+X(J,I)
      SUMM=SUMM+X(J,I)*XF(J)
80     SUM(J)=SUM(J)+X(J,I)
81     IF(N.EQ.5)XX(N1)=XX(N1)+X(J,N1)*XF(J)
       DO 82 I=1,N
```

```
82      XXP(I)=XX(I)/SUMM*100.
        IF(N.EQ.5)XXP(N1)=XX(N1)/XX(3)*100.
           WRITE(56,'(A5,A3,F5.0,F10.0,43F5.1)')
     *  (TITEL(I),I=2,3),DPTH,SUMC,(XXP(I),I=1,N1)
        WRITE(55,30)(TITEL(I),I=1,5)
         WRITE(55,'(1X,A5,f7.1)')TITEL(6),dpth
        WRITE(55,120)(XF(J),J=1,3),PERC,(XF(J),J=4,NFRAK)
90        WRITE(55,130)((X(J,I),J=1,3),XXP(I),(X(J,I),J=4,NFRAK),I=1,N1)
        WRITE(*, 110)(SUM(J),J=1,3),SUMC,(SUM(J),J=4,NFRAK)
100       WRITE(55,110)(SUM(J),J=1,3),SUMC,(SUM(J),J=4,NFRAK)
110      FORMAT(1X,10F6.0)
120       FORMAT(1X,3F6.0,1X,A5,3F6.0)
130       FORMAT(1X,3F6.0,F6.2,3F6.0)
      GOTO 10
140     CLOSE(25)
       CLOSE(55)
       CLOSE(56)
      STOP
      END
                integer function length(s)
                character*255 s
                do 1 i=len(s),1,-1
                if(s(i:i).eq.' ')goto 1
                length=i
                RETURN
1               continue
                return
                end
```

This program asks in a dialog for the name of the input file, the output file and the number of lines. In our case we insert as input file 16867-3.PLA, as output file "16867-3.P" and as number of lines "43". The program converts the fraction counts into percentage values for the total sample and calculates the sums counted for the single fractions as well as for the whole sample. The program output is arranged like the input, contains the counts, percentages and sums but the species names are omitted to save space (therefore the extension .P).
This .P file is the input for the Mac routine MAKEPOCP.APL, with the source code MAKEPOCP.FOR

```
c       makepocp.for 14-4-92
c       to read .p file and to generate epoc file with 38 species
        character*60 infil,outfil,INFMT
        character*8 kern,cm
        character*4 name(43),INNAM,LINFMT
        character yes
        logiCAL q,LP,LPLA,LF24,LF26,LF16,LF27
        dimension art(43)
        real ment
          data (name(i),i=1,43)/" aeq"," bul"," cal"," con"," cra"," deh",
     *  " dig"," dut"," fal"," glu"," hex"," hir"," hum"," inf"," men",
     *  " nit"," obl","pacl","pacr"," pel"," qui"," rur"," ruh"," ruw",
     *  "rusr","rusw"," tri"," sac"," sci"," tum"," trl"," trr"," uni",
     *  " uvu"," pdi"," flx"," ten","punc","tria","hess","rond","viol",
     * " cav"/
C         1     2     3     4     5     6
C        " aeq"," bul"," cal"," con"," cra"," deh",
C         7     8     9     10    11    12    13    14    15
C      *          " dig"," dut"," fal"," glu"," hex"," hir"," hum"," inf"," men",
C        16    17    18    19    20    21    22    23    24
C      *            " nit"," obl","pacl","pacr"," pel"," qui"," rur"," ruh"," ruw",
C        25    26    27    28    29    30    31    32    33
C      *            "rusr","rusw"," tri"," sac"," sci"," tum"," trl"," trr"," uni",
C        34    35    36    37    38    39    40    43    42
C      *            " uvu"," pdi"," flx"," ten","punc","tria","hess","rond","viol",
C        43
C     *        " cav"/
C     1   2   3   4   5   6   7   8   9   10  11   12  13
C     AEQU BULL CALIDCONGLCRASSDEHISDIGITDUTERPD-VRFALC FLEX  GLUT HEXAG
C      14  15  16  17  18  19  20  21  22  23  24   25   26
C     HIRS HUM  INFL MENARNITIDOBLIQPACH PELAGQUINQRUBERRUBES SACC SCITU
C      27  28  29  30  31  32  33  34  35
C     TUMIDTRUNCUNIV UVULATENELRUBRDSACSCPACLItrur
        LP=.FALSE.
        LPLA=LP
        LF24=LP
        LF26=LP
        LF16=LP
        LF27=LP
        Q=LP
          WRITE(9,"(30X,A)")"MAKEPOCP"," 28-4-92 ","PFLAUMANN"
         WRITE(9,"(A)")
       * " PLEASE, SPECIFY INPUT FORMAT (.P,     .F24,.F26          ) "
c      *  " PLEASE, SPECIFY INPUT FORMAT (.P,.PLA,.F24,.F26,.F16,.F27) "
         READ(9,"(A)")LINFMT
         IF(LINFMT.EQ.".P")THEN
        LP=.TRUE.
        INNAM=".P  "
          INFMT="(19x,f6.0)"
        ENDIF
          IF(LINFMT.EQ.".PLA".OR.LINFMT.EQ.".pla")THEN
        LPLA=.TRUE.
        INNAM=".PLA"
          INFMT="(32x,f6.0)"
        ENDIF
```

```
      IF(LINFMT.EQ.".F24".OR.LINFMT.EQ.".f24")THEN
     LF24=.TRUE.
      INNAM=".F24"
          INFMT="(a7,A6,13f5.1/13x,13f5.1/13X,13F5.1)"
     ENDIF
       IF(LINFMT.EQ.".F26".or.LINFMT.EQ.".f26")THEN
     LF26=.TRUE.
      INNAM=".F26"
          INFMT="(a7,A6,13f5.1/13x,13f5.1/13X,13F5.1)"
     ENDIF
       IF(LINFMT.EQ.".F16".OR.LINFMT.EQ.".f16")THEN
     LF16=.TRUE.
      INNAM=".F16"
     ENDIF
       IF(LINFMT.EQ.".F27".or.LINFMT.EQ.".f27")THEN
     LF27=.TRUE.
      INNAM=".F27"
     ENDIF
        WRITE(9,"(A17,A4,A)")" PLEASE, specify ",INNAM," input file "
       read(9,"(a)")infil
      IF(LP.OR.LPLA)THEN
       write(9,"(a)")
     *       " read quinqueloba left instead of pelagica? (Y/N) /N/ "
       read(9,"(a1)")yes
         if(yes.eq."y".or.yes.eq."Y")q=.true.
      endif
        L=length(infil,60)-1
       if(.not.lp)L=LENGTH(INFIL,60)-3
        outfil=infil(1:L)//'epo'
       open(5,file=infil)
      WRITE(9,"(A9,A)")" READING ",INFIL
       open(6,file=outfil)
      WRITE(9,"(A9,A)")" WRITING ",OUTFIL
          write(6,"(2(a4,4x),38(1x,a4))")"kern","cm",
     *  (name(i),i=1,3),name(43),
       *    (name(i),i=4,10),(name(i),i=12,18),"par+","quil","quir",
     *  name(21),
       *    name(22),name(24),(name(i),i=26,29),name(37),name(31),name(32),
       *    "trus",name(30),name(33),name(34),"ment",name(35),"pacr"

1     IF(LP.OR.LPLA)THEN
          read(5,"(6x,a8/6x,a8/)",end=2)  kern,cm
        read(5,INFMT)(art(i),i=1,43)
       read(5,INFMT)dummy
         trus=art(31)+art(32)
      ENDIF
       IF(LF24)THEN
      READ(5,INFMT,END=2)KERN,CM,
     *       (ART(I),I=1,8),ART(35),ART(9),DUMMY,ART(10),DUMMY,
     *      (ART(I),I=12,17),PACSUM,
     *      DUMMY,ART(21),RUBSUM,ART(26),TRISUM,ART(29),ART(30),TRUS,
     *       ART(33),ART(34),ART(37),RUR,SAC,PACLI
       ART(18)=PACLI*PACSUM/100.
       ART(19)=PACSUM-ART(18)
      ART(22)=RUR*RUBSUM/100.
```

```
      ART(24)=RUBSUM-ART(22)
      ART(28)=SAC*TRISUM/100.
      ART(27)=TRISUM-ART(28)
     ENDIF
      IF(LF26)THEN
     READ(5,INFMT,END=2)KERN,CM,
    *      (ART(I),I=1,8),ART(35),ART(9),DUMMY,ART(10),DUMMY,
    *     (ART(I),I=12,17),PACSUM,
     *       DUMMY,ART(21),ART(24),ART(26),ART(27),ART(29),ART(30),TRUS,
     *        ART(33),ART(34),ART(37),ART(22),ART(28),ART(18),ART(32)
       ART(19)=PACSUM-ART(18)-ART(35)
      ART(31)=TRUS-ART(32)
     ENDIF
        pars=art(19)+art(35)
      quil=0.
       if(q)quil=art(20)
       quis=quil+art(21)
       ment=art(15)+art(30)
                SUM=0.
                DO 10 I=1,37
10              SUM=SUM+ART(I)
                SUM=SUM-ART(11)+QUIS-ART(20)-ART(21)-ART(23)-ART(25)
C      cm(6:6)=" "
         write(6,"(2a8,38f5.1,F6.1)")kern,cm,
    *       (art(i),i=1,3),art(43),
     *          (art(i),i=4,10),(art(i),i=12,18),pars,quil,art(21),quis,
     *          art(22),art(24),(art(i),i=26,29),art(37),art(31),art(32),
      *    trus,art(30),art(33),art(34),ment,art(35),art(19),SUM
     goto 1
2    CLOSE(6)
                CLOSE(5)
                stop
     end
                INTEGER FUNCTION LENGTH(STR,L)
     CHARACTER*255 STR
      DO 1 I=L,1,-1
      IF(STR(I:I).EQ." ")GOTO1
     GOTO 2
1    CONTINUE
2    LENGTH=I
     RETURN
     END
```

MAKEPOCP.APL reads the .p file and generates a file with 38 species percentages per line. In a dialog it asks for the input file and generates an output file with the same name like the input file, but with the extension .epo.
This . epo serves as input for the SIMMAX procedure.

The SIMMAX MAT Approach:

Manual for technical handling.

Uwe Pflaumann,

Geol. Paläont. Institut Universität Kiel, Olshausenstr. 40, D-24098 Kiel,

Phone: 49 431-880-2928, Fax: 49 431 880 4376,

internet: up@gpi.uni-kiel.de

SIMMAX is an acronym indicating an MAT approach based on maximum similarity. The number 28 refers to the version 2.8. This version is dimensioned for at most 1700 analog samples, 50 species and 20 best analogs. It runs on a Powerbook 180c as well as on a Quadra700. After some minutes of installation the paleoSST estimates are done within x seconds per sample.

It is strongly recommended to read this manual carefully before starting any computations.

Introduction

The source code of the program, **SIMMAX28.fOR,** is written by the author in FORTRAN77 on a Mac as a FREDITOR document and is supported at the end of this manual. An ATARI version similar to IBM code is available on request. Here we focus on the Mac version.

The executable code, **SIMMAX28.apl** is compiled by the MacFortran/020 compiler version 2.4 of Absoft Corporation.

The following text informs about

Software requirements

To enable the application on Mac the system folder must contain the two small files f77.rl and f81.rl. This version of the application needs at least 8000k of RAM memory, to be provided by 'command I', otherwise it would not even start.

The working folder

The application **SIMMAX28.apl**, not the source code, should be copied into a separate folder. It is recommended to use the harddisk as quickest medium for data handling.

The application needs a file called **SIMMAX24.INP** stored in the same folder as the application. SIMMAX24.INP contains all information about input and output file names and formats as well as some options to be set. These data are read by the main application and has to be set up following strictly the FORTRAN77 rules. An explication will be given below.

The application needs the input data written in strict right oriented FORTRAN format. (Note: Most spreadsheets produce a left oriented text code.)

- 1) An **analog species data set**, containing the sites, positions and species assemblage data in percentage form,
- 2) an **analog SST data set** containing the 'measured' SST data (or other environmental data of the user's choice) of the stations of the analog species data set, which has to be aligned in the same order as the sites in the analog species data set.
- 3) The **subject species data set**, i.e. the species assemblage data set in percentage form of the subject core counts, the SST of which have to be estimated, the percentages of the subject data have to be aligned in the same order as the analog species data set, each sample beginning with the core depth label.
- 4) The **results** will be written in files defined by the user (in SIMMAX.INP) on the same folder where the application is started. Synonym files are overwritten without any warning.

This version is dimensioned for a maximum of 1700 analog samples, 50 species, and 20 best analogs.

The file SIMMAX24.INP and its organization:

This file will be read by the SIMMAX application. As it contains line and column sensitive FORTRAN codes, it is urgently recommended to make a safety copy of this file and to edit by overwriting/replacing , **not** by inserting items.

In most cases when the application gives strange results or coded error messages these are related to wrong formats in the SIMMAX24.INP file.

```
INFIL1:PFLWpac+scs.EP
FMT1  :(ZX,A9,2x,2f15.0,10x,3f5.1,5x,8F5.1,5x,5f5.1,15X,f5.1,5x,8F5.1,5X,2f5.1,10x,2f5.1)
INFIL2:17940-3.ep
FMT2  :(a16,                3f5.1,5x,8F5.1,5x,5f5.1,15X,F5.1,5X,8F5.1,5X,2f5.1,10x,2f5.1)
HYPO  :0.90
NNN   : 20
NSP   : 29
PFIL  :PFLWpac+scs.P37
PFMT  :( 37x,2F5.2)
NPAR  :  2
OUTFIL:17940-2.simmax28 0.9/20 875
FMTO  :(1x,a,          I4,f6.3,2(7F10.2),20I4)
LAT   : 20.1167
LON   :117.3831
```

```
LDIST :T
LTEST :F
LKOO  :F
LFMT1 :F                FMT1 in A-FMT=T, in REAL=FMT=F
TTEST :F                 outlier or new samples T, core F
```

This file contains 19 lines. All lines begin with a fixed code word ending with a ':'. All editing refers to the columns following this colon.

• Line 1 contains the name of the analog species file, case sensitive.
INFIL1:PFLWpac+scs.EP

This file must be located in the same folder as the application.

• Line 2 contains the FORTRAN Format of the analog species file.

```
FMT1
:(2X,A9,2x,2f15.0,10x,3f5.1,5x,8F5.1,5x,5f5.1,15X,f5.1,5x,8F5.1,5X,2f5.1,10
x,2f5.1)
```

The data are organized: analog site name ,latitude, longitude (in decimals acoording to line 18), 29 (according to Line 7) percentage data.

For non FORTRAN insiders: 2X means overread 2 columns, A9 means read ASCII text from the next 9 columns, 2f15.0 means read two decimal numbers in the next 2 times 15 columns, decimal sign is dot (.).

For convenience the extension .ep in the analog species file name indicates that the species are arranged in the "epo format" (see EPOC.TIT).

The arrangement of species is not critical, but must be the same alignment as in the subject file, however using the X-format there can be overread some data in one line.

• Line 3 contains the name of the subject file

• Line 4 contains the format of the subject file. Data are read in first the depth in the core (in ASCII text) then the species percentages according to the alignment of Line 2. In the example

```
FMT2  :(a16,              3f5.1,5x,8F5.1,5x,5f5.1,15X,F5.1,5X,8F5.1,5X,2f5.1,10x,2f5.1)
```

a16 means for the depth of the core the fist 16 columns are reserved. The following formats are like that of line 2, i.e. also epo format. Blank columns in the format are neglected.

• Line 5 `HYF0  :0.90` defines the lower limit of scalar product to be considered in the selection of best analogs.

• Line 6 defines the maximum number of best analogs to be considered

• Line 7 defines the number of species being considered

• Line 8 PFIL :PFLWpac+scs.f37 defines the name of the file containing the measured analog SST data. This file must be within the same folder as the application.

• Line 9 defines the format of the analog SST file PFMT :(37x,2F5.2). The data are read in as decimals, here from columns 38 to 47. The decimal dot is optional, if it is given in the SST file, then it is read in as written, otherwise, if it is not written in the SST file it is read according to the format, here in 2x5 columns, the decimal dot inserted before the 2. column from right end of the 5 column fields.

• Line 10 defines the number of measured SST data per sample

• Line 11 OUTFIL:17940-2.simmax28 0.9/20 875

defines the name of the output file, which will be generated in the same folder as the application.

• Line 12 FMTO :(1x,a, I4,f6.3,2(7F10.2),20I4) defines the format of the output file. The arrangement of the output data is subject file core depth; registered number of best analogs according to the limits defined in line 5 and 6; average similarity of the best analogs, n groups of SST estimates (according to Line 12), each group containing the geogr.distance weighted SST estimate, the difference to the measured SST (zero in paleoSST estimates), the non geographical SST estimates,the difference to the measured SST (zero in paleoSST estimates), the minimum, the maximum and the standard deviation of the SST of the best analogs; following the data of the SST estimate groups there are given the ordinal numbers of the best analogs according to their line numbers in the analog species file. The number before the expression (7f10,2) has to be the same as in Line 10 (will be programmed for automatical replacement).

• Line 13 defines the latitude of the site of the subject data. In case of calibration the lat is read in from the site data of the subject file.

• Line 14 defines the longitude of the site of the subject data. In case of calibration the lon is read in from the site data of the subject file.

• Line 15 to 19 contain in the first column after the colon logical or Boolean items, T for true or F for false.

• Line 15 after the colon insert a T LDIST :T if a geographical weighting is proposed

• Line 16 after the colon insert a T if a calibration run is proposed, a F if not LTEST :F.

• Line 17 after the colon insert a T LKOO :T if a file containing the geogr. coordinates of the best analogs should be created. This file will be generated in the same folder as the application and has the name of the subject file and the extension .koo

• Line 18 letter after the colon LFMTI :F FMTI in A-FMT=T, in REAL-FMT=F

defines the kind of format of the site coordinates of the analog file. F means the koordinates are given as decimal numbers, T means the coordinates are given in the format degree in decimals with or without dot, hemisphere as N or S or E or W, minutes in decimals with or without dot.

• Line 19 TTEST :F outlier or new samples T, core F

defines by T that a calibration test run is to be done on outliers or new samples in the subject file, by F is defined that the subject file contains normal subject data for estimating paleoSSTs.

The analog species percentage file

Example of an analog species percentage file

```
-1015-2       11S4667   12E5082 1638.  730  850    .   0.   0. 10  5.9  5.3
0.5  0.0  0.0  0.9  0.0  0.1 10.5  0.0  5.3  0.0  0.0  6.0  1.5  0.0  0.6
0.0  0.0  0.0  0.0  0.0 15.4 28.2  0.0  0.6  0.0  1.9  0.0  0.0  0.0  0.0
3.5 13.4  0.0  5.0  0.4  0.0
```

to be read by format

```
(a11,2(f5.0,a1,f4.2),34x,3f5.1,5x,8F5.1,5x,5f5.1,15X,F5.1,5X,8F5.1,5X,2f5.1,10x,2
f5.1)
```

as

```
kern lat lon aeq  bul  cal con  cra  deh  dig  dut  fal  glu  hir  inf  men
nit  obl pacl qui  ruw rusw  tri  sac  sci  ten  trl  trr  tum  uni  uvu pacr
```

It is not crucial, if the sum of the requested species percentages in the species data files does't sum up to 100%, as the application will normalize the selected percentages to unity before working with them.

The analog SST file

Example of an analog SST file to be read by format

```
(37x,f5.2,5x,f5.2)
```

```
-1015-2       11S4667   12E5082       2109 2459 2809 2355 1744
1868 2242 1832 1562 1672 1867 1662 1497 1571 1626 1554 1927 2164
2526 2094 1805 2000 2306 1950 1728 1893 2136 1851 1653 1770 2055
1747 1601 1704 1912 1683 1530 1622 1747 1608 0. 0. 10   6
```

as SSTwi = 21.09 °C, SSTsu = 28.09 °C for station -1015-2. The minus before the station number indicates that the given caloric temperatures are from southern hemisphere, i.e. the coldest season is August - October.

The output SST file

Example of an output SST file

```
  16415-2   10  .960      25.6        .0      25.3        .0      23.5
26.9       1.1      27.6        .0      27.7        .0      26.1      28.8
.7  25 162 168 568 572  30 645 163  24 584

  Station   NNN  simm      widw      delta      windw      delta        min
max       stdev       sudw      delta      sundw      delta        min        max
stdev 1  2   3   4   5   6   7   8   9   10
```

The optional .koo file

Example of a .koo file

```
-18.593  25.005 3     0.
 -19.133  26.917 2       V30-68
 -18.593  25.005 2     015637-1      3
 -20.883  25.717 2       V30-58
 -18.593  25.005 2     015637-1      3
 -42.450 -27.950 2      -V16-190
 -18.593  25.005 2     015637-1      3
 -20.330  25.450 2       MIKS13
 -18.593  25.005 2     015637-1      3
 -26.280  30.130 2       MIKS18
 -18.593  25.005 2     015637-1      3
 -19.517  25.717 2       V30-59
 -18.593  25.005 2     015637-1      3
 -20.067  22.467 2       V29-170
 -18.593  25.005 2     015637-1      3
 -19.300  25.833 2       A180-39
 -18.593  25.005 2     015637-1      3
 -28.930  24.050 2       MIKP09
 -18.593  25.005 2     015637-1      3
 -25.400  28.880 2       MIKS14
 -18.593  25.005 2     015637-1      3
```

This file contains decimal values of latitiude, longitude, pen (3=up, 2=down, in plotter language convention), station and core depth data in ASCII text mode. By adding the necessary header lines. Therefore, this file can be used as line input file for the WHIZMAP graphic system, to generate line graphs of the geographic position of the best analogs of the subject sample.

Abbreviations and acronyms

(in file: wp13-g:\toni\epoc.tit 4-6-93)
headline for fra11n.ep
stat lat lon m # count aut r u l aeq bul cal x con cra deh dig dut fal glu hir hum inf men nit obl pal par+ qul qur qui rur ruw rus tri sac sci ten trl trr trus tum univ uvu ment pdi par

headline for fra11.f37
station lat lon wi1 sp1 su1 fa1 wi4 sp4 su4 fa4 wi5 sp5 su5 fa5 wi6 sp6 su6 fa6 wi14 sp14 su14 fa14 wi15 sp15 su15 fa15 wi16 sp16 su16 fa16 wi45 sp45 su45 fa45 wi46 sp46 su46 fa46 wi56 sp56 su56 fa56 r u len #

Legend for the abbreviations in the top line:
Latitude and longitude in degree and minutes*100
depth in m
# = internal code
count = specimens counted
aut = author of determination
.=Pflaumann, Duprat, Pujol
sp71 = Specmap
thi = Thiede
r, u, l = position of label (Plotting specials)

file .f37 contains temperatures from Levitus climat. Atlas interpolated for the resp. stations by the mean of the Euklidian distancy of the 4 nearest gridpoints of a 1 by 1 degree grid.

wi1 caloric winter Levitus level 1 (= 0 m)
sp1 caloric spring Levitus level 1 (= 0 m)
su1 caloric summer Levitus level 1 (= 0 m)
fa1 caloric fall Levitus level 1 (= 0 m)

wi4 dito level 4 (= 30 m)
sp4
su4
fa4

wi5 dito level 5 (= 50 m)
sp5
su5
fa5

wi6 dito level 6 (= 75 m)
sp6
su6
fa6

wi14 sp14 su14 fa14 are the means of level 1 and 4

wi15 sp15 su15 fa15 are the means of level 1, 4, and 5

wi16 sp16 su16 fa16 are the means of level 1, 4, 5, and 6

wi45 sp45 su45 fa45 are the means of level 4 and 5

wi46 sp46 su46 fa46 are the means of level 4, 5, and 6

wi56 sp56 su56 fa56 are the means of level 5 and 6

r u len # are plotting and organizing specifics needed for the ATARI version

dw means geogr. distancy weighted
ndw means not geogr. distancy weighted
std = standard deviation
NN = number of nearest neighbours= best analogs
sim = similarity index
N1 TO N10 catalogue number of the 10 nearest neighbours
DL = Levitus + Dietrich combined at equal weights
DP = Levitus replaced by Dietrich data
DLWI= DL winter, DPWI= DP winter
DLSU= DL summer, DPSU= DP summer

LWI1 = LEVITUS wi1
LSP1 = LEVITUS sp1
LSU1 = LEVITUS su1
LFA1 = LEVITUS fa1

9999 or 99.99 means no value

Table A1. Faunal taxa distinguished for the SST estimation and summary statistics of species occurrence.

Variable	Mean	Standard Deviation	Minimum Value	Maximum Value	Range	Name
bul	11.063	12.742	0,0	73.7	73.7	*Globigerina bulloides*
cal	0.425	0.808	0.0	5.9	5.9	*Globigerina calida*
fal	1.598	3.580	0.0	31.0	31.0	*Globigerina falconensis*
qui	2.935	6.842	0.0	48.9	48.9	*Globigerina quinqueloba*
rus	0.566	1.236	0.0	0.3	0.3	*Globigerina rubescens*
dig	0.238	0.409	0.0	2.4	2.4	*Globigerina (Beella) digitata*
aeq	1.863	2.057	0.0	10.1	10.1	*Globigerinella aequilateralis*
con	0.354	0.810	0.0	10.0	10.0	*Globigerinoides conglobatus*
rur	1.620	3.264	0.0	21.8	21.8	*Globigerinoides ruber pink*
ruw	13.958	16.100	0.0	78.2	78.2	*Globigerinoides ruber white*
ten	0.498	1.085	0.0	8.6	8.6	*Globigerinoides tenellus*
tri	4.607	6.595	0.0	40.0	40.0	*Globigerinoides trilobus trilobus*
sac	1.911	3.230	0.0	23.5	23.5	*Globigerinoides trilobus sacculifer*
uni	1.086	1.874	0.0	19.6	19.6	*Orbulina universa*
deh	0.101	0.317	0.0	3.8	3.8	*Sphaeroidinella dehiscens*
cra	0.512	0.986	0.0	8.2	8.2	*Globorotalia crassaformis*
mentum	2.980	5.630	0.0	40.1	40.1	*Globorotalia cultrata* and *tumida*
hir	0.800	1.500	0.0	8.6	8.6	*Globorotalia hirsuta*
inf	7.643	9.508	0.0	61.0	61.0	*Globorotalia inflata*
sci	0.982	1.471	0.0	9.3	9.3	*Globorotalia scitula*
trs	2.301	3.331	0.0	19.8	19.8	*Globorotalia truncatulinoides*
dut	3.944	7.148	0.0	49.1	49.1	*Neogloboquadrina dutertrei*
pal	16.852	32.394	0.0	100.0	100.0	*Neogloboquadrina pachyderma* left coiling
par	13.809	17.167	0.0	80.5	80.5	*Neogloboquadrina pachyderma* right coiling including "p/d-intergrades"
glu	5.542	5.149	0.0	35.7	35.7	*Globigerinita glutinata*
obl	0.735	1.794	0.0	15.8	15.8	*Pulleniatina obliquiloculata*

Abbreviated variable names are explained in the last column.

Source code of SIMMAX28.fOR

```
      PROGRAM MAIN
c       simmax28a 11-04-95
c      ex age.for a.altenbach diss.
c       korrelationskoeff. zwischen einer kernprobe und faunen aus der
c      sedimentoberflaeche
c       schema einlesen der oberflaechenfaunen (a2(50,1500)),phys.prop.
c           normieren
c            einlesen einer kernfauna       (a1(50))
c            normieren
c              errechnen des sim.koeff. zu allen oberflaechenfaunen
c            suche nach nnn hoechsten sim. koeff
c              errechnen der mittelwerte der tw, ts, ss, pp aus den oberfl.
c            wichten nach sim . koeff.
c                        standardabw. fÜr die kernfauna
c           wegschreiben
c     CHARACTER*188 A188
      CHARACTER*80 INFIL1, INFIL2, PFIL, OUTFIL,KOOFIL
     CHARACTER*160 FMT1,FMT2,FMTO,FMTP
      CHARACTER*65 TEUFE,STAT,STATION
     CHARACTER SN,EW,YES
     LOGICAL LDIST,LTEST,LKOO,LFMT1,TTEST,LCOLUMN
       REAL LON(1700), LAT(1700), GKM(1700)
         DIMENSION A1(50),A2(50,1700),P(20,1700),A(1700,2)
      DIMENSION PN(20),PNQ(20),PSP(20),DPN(20),PNST(20)
      DIMENSION PSPMIN(20),PSPMAX(20),PNNMIN(20),PNNMAX(20)
      DIMENSION PNN(20),DPNN(20),JMAX(20),
       DIMENSION A1N(50),A2N(50,1700),MINSP(50),MAXSP(50)
      DIMENSION SP(1700),STAT(1700)
        WRITE(9,"(60X,A)")"SIMMAX28a_1700"," 11-04-95","PFLAUMANN,
KIEL"
C     TESTVERSION
C       FMTO="(i5,A54,i5,12F6.2,4f6.0,a13)"
      DUMMY=0.
c       Einlesen des Parameterfiles
     OPEN(5, FILE="SIMMAX24.INP")
      READ(5,"(7X,A)")INFIL1
      READ(5,"(7X,A)")FMT1
       READ(5,"(7X,A)")INFIL2
      READ(5,"(7X,A)")FMT2
      READ(5,"(7X,F10.0)")HYP0
      READ(5,"(7X,I3)")NNN
```

```
      READ(5,"(7X,I3)")NSP
      READ(5,"(7X,A)")PFIL
     READ(5,"(7X,A)")FMTP
      READ(5,"(7X,I3)")NPAR
     READ(5,"(7X,A)")OUTFIL
     READ(5,"(7X,A)")FMTO
      READ(5,"(7X,F10.0)")RN
      READ(5,"(7X,F10.0)")RW
     READ(5,"(7X,L1)")LDIST,LTEST,LKOO,LFMT1,TTEST
C     TESTVERSION
C      CLOSE(5)
      LCOLUMN=.true.
      WRITE(9,"(7X,A80)")INFIL1
      WRITE(9,"(7X,A160)")FMT1
      WRITE(9,"(7X,A80)")INFIL2
      WRITE(9,"(7X,A160)")FMT2
     WRITE(9,"(7X,F10.4)")HYP0
     WRITE(9,"(7X,I3)")NNN
      WRITE(9,"(7X,I3)")NSP
      WRITE(9,"(7X,A80)")PFIL
     WRITE(9,"(7X,A160)")FMTP
     WRITE(9,"(7X,I3)")NPAR
     WRITE(9,"(7X,A80)")OUTFIL
     WRITE(9,"(7X,A160)")FMTO
     WRITE(9,"(7X,F10.0)")RN
     WRITE(9,"(7X,F10.0)")RW
    WRITE(9,"(7X,L1)")LDIST,LTEST,LKOO,LFMT1,TTEST,LCOLUMN
c     testen auf richtiges datenformat
        open(50,file=infil2)
C     TESTVERSION
c     OPEN(50,FILE="FRA11N.EPO")
c       if(lkoo)then
c                 koofil=outfil(1:(length(outfil)-4))//".koo"
c           koofil="koofil.tmp"
c                   write(9,"(a)")koofil
c                   pause 1
c            ipen=3
c            dummy=0.
c              open(7,file=koofil)
c                 write(7,"(2f8.3,i2,f6.0)")rw,rn,ipen,dummy
c            ipen=2
c      endif
     WRITE(9,"(A18,A80)")" reading testfile ",INFIL2
C     TESTVERSION
```

```
c           write(9,"(a18,a)")" reading testfile ",infil2
c      pause 2
       NK=0
       DO 577 I=1,NSP
577      A1(I)=0.
        READ(50,fmt2,END=502,ERR=501)TEUFE,(A1(I),I=1,NSP)
C      TESTVERSION
555   FORMAT
     *  (A65,3F5.1,5X,8F5.1,5X,F5.1,10X,3F5.1,10X,8F5.1,5X,3(5X,F5.1))
c           write(9,fmto)i,teufe
c          write(9,"(a)")" j, n, nr, t1, rq"
      LENT=LENGTH(TEUFE,80)
C      TESTVERSION
c      WRITE(9,"(a15,i4)")" length(teufe)=",LENT
c      pause 1

          SN=" "
          EW=SN
      IF(LTEST.OR.TTEST)THEN
             OPEN(10)
            WRITE(10,"(A)")TEUFE(1:LENT)
C      TESTVERSION
c                   chardum=teufe(1:lent)//"."
c                   write(9,"(1x,a80)")chardum
c                   pause 1
            REWIND (10)
            IF(LFMT1)THEN
              READ(10,FMT1)STATION(1:9),X1,SN,X2,X3,EW,X4
            ELSE
              READ(10,FMT1)STATION(1:9),RN,RW
            ENDIF
               CLOSE(10,STATUS="delete")
             RN=X1+X2/60.
            IF(SN.EQ."S")RN=-RN
              RW=-(X3+X4/60.)
            IF(EW.EQ."E")RW=-RW
             IPEN=3
            DUMMY=0.
              IF(LKOO)WRITE(7,"(2F8.3,I2,A5,A)")
     *                RW,RN,IPEN,"   0",STATION
             IPEN=2
       ENDIF
          CLOSE(50)
          CLOSE(7)
```

```
          GOTO 501
500           WRITE(9,"(a22,A)")" FORMAT ERROR IN FILE ",INFIL2
              WRITE(9,"(A16,A)")" FORMAT READ IS ",FMT2
              PAUSE
              STOP
502           WRITE(9,"(A17,A)")" FILE NOT FOUND: ",INFIL2
              PAUSE
              STOP
501   CONTINUE
c       einlesen des arten-datenfiles
C     TESTVERSION
c      OPEN (20,FILE="fra11N.epo")
       OPEN (20,FILE=infil1)
      write(*,*)'  reading  ',infil1
      J=1

10    IF(LFMT1)THEN
C     TESTVERSION
c       +"(A9,2x,2(f5.0,a1,f4.2),34X,3f5.1,5x,8F5.1,5x,f5.1,10X,3F5.1,10X,
c      *8F5.1,5X,3(5x,f5.1))"
c     *,ERR=110,END=11)
       READ(20,FMT1
     *,ERR=110,END=11)
     *STAT(J),X1,SN,X2,X3,EW,X4,
      *   (A2(I,J), I=1,NSP)!(1000*50 arten)
        A(J,2)=X1+X2/60.
        IF(SN.EQ."S")A(J,2)=-A(J,2)
        A(J,1)=-(X3+X4/60.)
       IF(EW.EQ."E")A(J,1)=-A(J,1)
      ELSE
       READ(20,FMT1,END=11)STAT(J),A(J,2),A(J,1),
      *   (A2(I,J),I=1,NSP)!(1000*50 arten)
      ENDIF

      IF(MOD(J,100).EQ.0)THEN
c       call atrtop
       WRITE(9,"(I4)")J
      ENDIF

      J=J+1
      GOTO 10
110   WRITE(9,"(a15,A80)")" error in file ",INFIL1
11     NS=J-1
      WRITE(9,"(A15,A80)")" END OF FILE   ",INFIL1
```

```
      CLOSE(20)
C     NOW ALL COORDINATES ARE IN MEMORY
C     STAT(J),A(J,2),A(J,1),(A2(I,J),I=1,NSP)!(1000*50  arten)
C     TESTVERSION
C     AND MAY BE OPTIONALLY COLUMN NORMALIZED
      IF(LCOLUMN)THEN
      DO 111 I=1,NSP
      MAXSP(I)=(-1000.)
111   MINSP(I)=-(MAXSP(I))
      DO 112 J=1,NS
      DO 112 I=1,NSP
      MINSP(NSP)=AMIN1(MINSP(NSP),A2(I,J))
112   MAXSP(NSP)=AMIN1(MAXSP(NSP),A2(I,J))

      DO 113 I=1,NSP
113   A2(I,J)=(A2(I,J)-MINSP(I))/(MAXSP(I)-MINSP(I))
      ENDIF

       OPEN(21,FILE=pfil)
        WRITE(9,"(A9,I4,A10,A)")" reading ",NS,"lines of ",PFIL
       DO 400 J=1,NS
       READ(21,FMTP
     * ,ERR=300)(P(I,J),I=1,NPAR)
C     TESTVERSION
c     write(9,*)(P(I,J),I=1,NPAR)
c     pause 1
      IF(MOD(J,100).EQ.0)THEN
c           call atrtop
           WRITE(9,"(I4)")J
      ENDIF
400   CONTINUE
      CLOSE(21)

      CALL NORM2(A2,NSP,NS,A2N)

c      einlesen der kerndaten
404     OPEN(51,FILE=infil2)
       OPEN(6, FILE=outfil)
       write(6,'(a,23x,a,20(a,a))') ' Station        cm','NNN  simm',
     *     ('       dw     delta',
     *      '      ndw     delta      min      max     stdev'
     * ,I=1,NPAR)
       write(9,'(a,23x,a,20(a,a))') ' Station        cm','NNN  simm',
     *     ('       dw     delta',
```

```
     *      '      ndw     delta      min      max     stdev'
     *  ,I=1,NPAR)
        IF(LKOO)THEN
             KOOFIL=OUTFIL(1:(LENGTH(OUTFIL,80)-4))//".KOO"
             IPEN=3
            DUMMY=0.
            OPEN(7,FILE=KOOFIL)
             WRITE(7,"(2F8.3,I2,F6.0)")RW,RN,IPEN,DUMMY
             IPEN=2
        ENDIF
           WRITE(9,"(A18,A)")" reading testfile ",infil2
        NK=0
1       DO 77 I=1,NSP
77        A1(I)=0.
          READ(51,fmt2,end=200)TEUFE,(A1(I),I=1,NSP)

C       TESTVERSION
c       *       "(a31,34x,3f5.1,5x,8F5.1,5x,f5.1,10X,3F5.1,10X,8F5.1,5X,
c        *  3(5x,f5.1))",END=200)TEUFE,(A1(I),I=1,NSP)

       LENT=LENGTH(TEUFE,65)
             SN=" "
             EW=SN
       IF(LTEST.OR.TTEST)THEN
             OPEN(10)
            WRITE(10,"(A)")TEUFE(1:LENT)
C       TESTVERSION
c         chardum=teufe(1:lent)//"."
c         write(*,"(1x,a)")chardum
c       pause
           REWIND (10)
           IF(LFMT1)THEN
             READ(10,1010)STATION(1:9),X1,SN,X2,X3,EW,X4
1010         FORMAT(A9,2X,2(F5.0,A1,F4.2))
C       TESTVERSION
c         *,34x,3f5.1,5x,8f5.1,5x,f5.1,10x,3f5.1,10x,8f5.1,5x,3(5x,f5.1))

            RN=X1+X2/60.
           IF(SN.EQ."S")RN=-RN
             RW=-(X3+X4/60.)
           IF(EW.EQ."E")RW=-RW
           ELSE
             READ(10,FMT1)STATION(1:9),RN,RW
           ENDIF
```

```
              CLOSE(10,STATUS="delete")
             IPEN=3
            DUMMY=0.
              IF(LKOO)WRITE(7,"(2F8.3,I2,A5,A)")
     *              RW,RN,IPEN,"   0",STATION
             IPEN=2
       ENDIF
        IF(LCOLUMN)THEN
        DO 114 i=1,NSP
114    A1(I)=(A1(I)-MINSP(I))/(MAXSP(I)-MINSP(I))
        endif
c        normieren
       CALL NORM1(A1,NSP,A1N)
C      TESTVERSION
c            write(*,"(1x,10f7.4)")(a1n(i),i=1,nsp)
       NK=NK+1
c          errechne similaritaetsindex (sp(j) zu allen oberflaechenfaunen
c         sp(j) ist skalarprodukt
       DO 8 J=1,NS
       SP(J)=0.
       DO 8 I=1,NSP
         SP(J)=SP(J)+A1N(I)*A2N(I,J)
8      CONTINUE
c
c       suche die nnn hoechsten sp und speichere deren NUMMER in
(JMAX(K),K=1,nnn)
      CALL SIMMAX(SP,NNN,NS,JMAX,NK,LTEST,HYP0,NNNN)
              IF(NNNN.LT.1)THEN
              WRITE(6,fmto)
     *       TEUFE(1:37),NNNN
              GOTO 1
              ENDIF

c         mitteln der nnn naechst aehnlichen proben
       DO 100 I=1,NPAR
         pnnmin(i)=64000.
       pnnmax(I)=-(pnnmin(i))

       PN(I)=0.
       PNN(I)=0.
100      PNQ(I)=0.
      SPSUM=0.
       SIMSUM=0.
```

```
      DO 101 K=1,NNNN
      J=JMAX(K)
       LON(K)=A(J,1)
        LAT(K)=A(J,2)
      GKM(J)=1.
      IF(LDIST)THEN
c      suche distanz gkm(j) der jten probe von der testprobe
      CALL ORTHOD(A(J,1),A(J,2),RW,RN,GKM(J),   GSM)
       IF(GKM(J).LT.2.)GKM(J)=20000. !unterdrücken von eigener
substation
      ENDIF

       SIMSUM=SIMSUM+SP(J)
       SPSUM=SPSUM+SP(J)/GKM(J)   !merken der wichtungen
(summe(scalarproducte))
      DO 102 I=1,NPAR
        PSP(I)=P(I,J)*SP(J)/GKM(J) !gewichtete parameter (nach aehnl.
u.distanz)
C     TESTVERSION
C     if(j.ge.60.and.j.le.61)then
c           write(*,*)i,j,psp(i),p(i,j),sp(j),gkm(j)
c           pause 44
c           endif

102  CONTINUE

     DO 103 I=1,NPAR
C     TESTVERSION
C        PSP(I)=P(I,J)*SP(J)/GKM(J) !gewichtete parameter (nach aehnl.
u.distanz)
       PN(I)=PN(I)+PSP(I)        !sum of weighted params
      PNN(I)=PNN(I)+P(I,J)       !sum of unweighted param
      PNQ(I)=PNQ(I)+(P(I,J) * P(I,J))!sum of squares of unw. params
C     TESTVERSION
C     if(i.eq.1)write(*,*)i,pn(i),pnn(i),pnq(i)
      pnnmin(I)=amin1(pnnmin(I),p(i,j))
      pnnmax(I)=amax1(pnnmax(I),p(i,j))

103  CONTINUE

101   CONTINUE
C     pause 7777
      fn=float(nnnn)
           fN1=fn-1.
```

```
      DO 105 I=1,NPAR
      IF (FN1.LE.1.)THEN
            PNST(I)=0.
            GOTO 105
      ENDIF
C
       PNST(I)=SQRT((PNQ(I)-PNN(I)*PNN(I)/fN)/fn1) !standard dev. of
C     &unw. params
105   CONTINUE
      DO 106 I=1,NPAR
C     TESTVERSION
C           if(i.eq.1)write(*,*)"pn(i),spsum,nnn",pn(i),spsum,nnn
            PNN(I)=PNN(I)/fN
106     PN(I)=PN(I)/SPSUM       !recalc. weights
      DO 107 I=1,NPAR
      DPN(I)=0.0
            DPNN(I)=0.
      IF(LTEST)THEN
            DPN(I)=P(I,NK)-PN(I)
            DPNN(I)=P(I,NK)-PNN(I)
            ENDIF
107   CONTINUE
            SIMMEAN=SIMSUM/NNNN
C     TESTVERSION
c           write(6,fmto)teufe,nnn,(pn(i),i=1,npar),(lon(k),lat(k),k=1,nnn)
     IF(LTEST.OR.TTEST) THEN
C     TESTVERSION
c            write(6,"(a31,i4,f6.3,6(5f6.1))")
      WRITE(6,FMTO)
    *  TEUFE(1:16),NNNN,SIMMEAN,(PN(I),DPN(I)
     *         ,PNN(I),DPNN(I),pnnmin(i),pnnmax(i),PNST(I),I=1,NPAR)
     *       ,(jmax(k),k=1,nnnN)
c     *         ,(p(i,jmax(k)),k=1,nnnN),i=1,npar)
      ELSE
       WRITE(9,fmto)
   *TEUFE(1:37),NNNN,SIMMEAN,(PN(I),DUMMY
     *         ,PNN(I),DUMMY   ,pnnmin(I),pnnmax(I), PNST(I),I=1,NPAR)
     *      ,(JMAX(K),K=1,NNNN)
      WRITE(6,fmto)
   *TEUFE(1:37),NNNN,SIMMEAN,(PN(I),DUMMY
     *         ,PNN(I),DUMMY   ,pnnmin(I),pnnmax(I), PNST(I),I=1,NPAR)
     *      ,(JMAX(K),K=1,NNNN)
c     *         ,(p(i,jmax(k)),k=1,nnnN),i=1,npar)
      ENDIF
```

```
c
C     TESTVERSION
c       WRITE(9,"(I4,1X,a16)")NK,TEUFE(1:16)
c     ,(pn(i),i=1,npar)
c     *       ,(dpn(i),i=1,npar),simmean
c
      IF(.NOT.LKOO) GOTO 1
C     GOTO 1
c     schreibe in den .koo file
      DO 108 I=1,NNNN
      J=JMAX(I)
            IF (LTEST.OR.TTEST)THEN
       WRITE(7,"(2F8.3,I2,A5,A13,20F10.4)")A(J,1),A(J,2),IPEN,"      "
     *      ,STAT(J)
     *      ,(P(IP,J),IP=1,NPAR)
      WRITE(7,"(2F8.3,I2,A5)")RW,RN,IPEN,"    0"
            ELSE
       WRITE(7,"(2F8.3,I2,A5,A13)")A(J,1),A(J,2),IPEN,"      ",STAT(J)
      WRITE(7,"(2F8.3,I2,A5,A13)")RW,RN,IPEN,"    0",TEUFE
            ENDIF
108         CONTINUE
      GOTO 1

200   CONTINUE
      close(51)
      close(7)
      close(6)
      WRITE(9,*) ' '
      write(9,*) ' Do you like to calculate an additional core? (Y/N)'
      read (9,'(a)') yes
      if(yes.eq.'Y' .or. yes.eq. 'y')then
       READ(5,"(7X,A)",end=405)INFIL2
      READ(5,"(7X,A)")FMT2
      READ(5,"(7X,A)")OUTFIL
      READ(5,"(7X,A)")FMTO
      READ(5,"(7X,F10.0)")RN
      READ(5,"(7X,F10.0)")RW
C     TESTVERSION
c       call  parread(infil2,fmt2,outfil,fmto,rn,rw)
      goto 404
      else
405   close(5)
      endif
      WRITE(9,*) ' '
```

```
      WRITE(9,*) 'DONE! RETURN TO EXIT!'
      PAUSE
      STOP
300      WRITE(9,*)I,J,(P(II,J),II=1,I)
     PAUSE
     STOP
     END

      subroutine  normcol(infil,range,nsp)
C     IN DEVELOPMENT
c     read modern analog file and search for min, max of defined species
c     then normalize species percentages and write into
columnnormalized
c     colnormfile. The sum of the rows will surely be not 100 but the
file
c     could be used like the percentage file. To  be consistent the test
c     file data have to be column normalized  by the same vectors.
      character*80  infil
      real  range(50)
      open(5,  file=infil)

      close(5)
      return
      end


     SUBROUTINE NORM1(A1,NSP,A1N)
c     normalize selected  NSPspecies percentages in A1(50) that sum of
their  squares equals unity
c     and store the result in A1N (50)
      DIMENSION A1(50),A1N(50)
     RSUM=0.
     DO 1 I=1,NSP
1      RSUM=RSUM+A1(I)*A1(I)
     RSUM=SQRT(RSUM)
     DO 2 I=1,NSP
      A1N(I)=0.
      IF(RSUM.NE. 0.)A1N(I)=A1(I)/RSUM
2     CONTINUE
     RETURN
     END

     SUBROUTINE NORM2(A2,NSP,NS,A2N)
       DIMENSION A2(50,1700),A2N(50,1700),RSUM(1700)
```

```
      DO 2 J=1,NS
      RSUM(J)=0.
      DO 1 I=1,NSP
1       RSUM(J)=RSUM(J)+A2(I,J)*A2(I,J)
      RSUM(J)=SQRT(RSUM(J))
      DO 2 I=1,NSP
       A2N(I,J)=0.
       IF(RSUM(J).NE. 0.)A2N(I,J)=A2(I,J)/RSUM(J)
2     CONTINUE
      RETURN
      END
      SUBROUTINE ATRTOP
c      this routine is called only by the ATARI version
c        character e
c        e=char(27)
c           write(*,"(1x,6a1)")e,"d",e,"h",e,"k"
      RETURN
      END

      SUBROUTINE SIMMAX(SP,NNN,NS,JMAX,NK,LTEST,HYP0,NNNN)
c       suche die nnn hoechsten sp und speichere deren Nummer in
(Jmax(K),k=1,nnn)
c       search for the maximum nnn scalar products and store their
ordinal line number in the analog file
c       nfil1   into the array (Jmax(K),k=1,nnn)
      LOGICAL LTEST
       DIMENSION SP(1700),JMAX(20),SP1(1700)
            NNNN=0
      DO 3 J=1,NS
3       SP1(J)=SP(J)
      DO 2 K=1,NNN
      SIMMX=0.
      DO 1 J=1,NS
      IF(LTEST.AND.J.EQ.NK)GOTO 1
C      TESTVERSION
C      SP1(J).GT.HYP0 .AND.
      IF (SP1(J) .GT. SIMMX)THEN
            SIMMX=SP1(J)
            JMAX(K)=J
C      TESTVERSION
C      PAUSE 3
      ENDIF
1     CONTINUE
      J=JMAX(K)
```

```
      SP1(J)=0.
C     TESTVERSION
c          write(*,*)j,sp(j)
      IF(SP(J).GT.HYP0)NNNN=NNNN+1
2     CONTINUE

C     TESTVERSION
c          WRITE(*,*)' NNNN= ', NNNN
c          pause  6666

      RETURN
      END

      SUBROUTINE ORTHOD(X1,Y1,X2,Y2,GKM,GSM)
c     grossk.for
c     entfernung in sm oder km auf grosskreis
c     x1,y1 und x2,y2 sind geogr.koordinaten in grad
      REAL*8 DD,DY1,DY2,DX,DG,DKM
      D=ATAN(1.)/45.
      DD =DBLE(D)
      DY1=DBLE(Y1)*DD
      DY2=DBLE(Y2)*DD
      DX =DBLE(X1-X2)*DD
C     TESTVERSION
c     write(*,*)'X1,Y1,X2,Y2'
c     write(*,*)X1,Y1,X2,Y2
c     pause

      DG=DSIN(DY1)*DSIN(DY2)+DCOS(DY1)*DCOS(DY2)*DCOS(DX)
      DKM=DACOS(DG)/DD
      GKM=SNGL(DKM)*111.2
      GSM=GKM/1.852
      RETURN
      END

      FUNCTION LENGTH(STR,L)
c     this function is called only by the Mac version
      CHARACTER*255 STR
      DO 1 I=L,1,-1
      IF(STR(I:I).EQ." ")GOTO1
      GOTO 2
1     CONTINUE
2     LENGTH=I
      RETURN
```

END