Optimization of the Ewens/Caswell neutral model program for community diversity analysis

N. Goldman¹, P. J. D. Lambshead²

¹Biometrics Section & ²Lower Invertebrates Division, Zoology Department, British Museum (Natural History), Cromwell Road, London SW7 5BD, United Kingdom

ABSTRACT: The neutral model program of Ewens (1972) was introduced into ecology by Caswell (1976) and has become a useful tool for benthic ecology. Lambshead & Platt (1988) noted that Ewens's program presented serious computational problems which could not be effectively overcome by subsampling or by substitution of a simpler equitability method. In this paper a modified program is presented which greatly reduces these computational difficulties, making it suitable for personal computers.

INTRODUCTION

The Ewens/Caswell neutral model was originally programmed by Ewens (1972) to test for neutrality between alleles in population genetics. This program was introduced into ecology by Caswell (1976) who used it with great success to test for interspecific competition. May (1984) suggested that Caswell's paper 'deserves more attention than it has received'. Indeed, in the field of terrestrial ecology only Hanski (1983) appears to have utilized the model. In contrast, a number of marine benthic biologists (e.g. Gray 1978, Rainer 1981, Warwick 1981, Warwick & Gee 1984, Platt & Lambshead 1985, Lambshead 1986) have made significant use of the model in both competition and disturbance analysis. It may be that the type of quantitative samples produced in benthic experiments are more suited to this type of work than those produced in other investigations.

The neutral model program as written by Ewens (1972) was not developed for high speed or low memory requirements, as these factors were not apparently a significant problem with the size of data sets used in population genetics. But computer memory and speed restrictions often become significant with ecological data. Platt & Lambshead (1985) were unable to analyse some samples in a data series using this program, despite employing a large and capable mainframe computer system. Lambshead & Platt (1988) attempted to avoid these computational problems by either reducing sample sizes or by utilizing a simpler equitability index which could be 'calibrated' against the model. However, these authors eventually concluded that 'there is no good substitute for running the raw data from a whole individual sample' and that 'no simpler equitability index performed overall as well'.

The only remaining possibility was to revise the program to reduce the effects of the computer constraints on speed and memory size, so that larger data sets could be analysed within reasonable time periods. For example, Caswell (1976) used an optimized version of Ewens's program which resulted in 'considerable savings in computer time' (Caswell pers. comm.). This paper reports on an improved version of Ewens's program, suitable for microcomputers. It has been optimized for both speed and memory requirement.

PROGRAM DETAILS

The revised program was written in FORTRAN to run on computers using the MS-DOS or PC-DOS operating systems. It requires an input file, called CASDAT.DAT, containing the data to be analysed. The format of the input file is shown in the Appendix. Note that the user has to specify N (the number of individuals in the sample), k (the number of species in the sample) and the data for the number of individuals in each species (which may be in any order). Output is sent to a file created by the program called CASOUT.DAT: if the...
computation is unable to create a new file with this name, it will stop immediately with an error report.

The program calculates $H'$ (Shannon diversity index), $E[H']$ (expected $H'$ for a 'neutral' sample), $SD[H']$ (standard deviation of $E[H']$) and $V$ (number of standard deviations $H'$ is from $E[H']$). A statistical test of neutrality is possible, under the assumption that $H = \log(k)$ has a beta distribution. The program calculates the appropriate F-statistic, together with its (non-integer) degrees of freedom (see Ewens 1972).

The program contains some further error checks. It adds the entered data and checks the total against the value entered for $N$. If the two do not tally it gives an error message but continues to run, using the value for $N$ obtained from the data. In the event of such an error message, both $N$ and the data as listed in the output file need to be checked. If an incorrectly low value for $k$ has been entered then all the data will have been read and thus $N$ will not tally even though it and the data may have been correctly typed. An incorrectly high value for $k$ will cause the program to stop immediately, with an error message. Note that the results produced by the program will only be valid if the data and $k$ have been correctly entered ($N$ is just used for error checking).

**COMPUTATIONAL PROBLEMS**

**Memory constraint.** To calculate $E[H']$ and $SD[H']$, Ewens's (1972) FORTRAN program calculates a matrix of numbers $w_{ij}$ defined by:

$$w_{ij} = \frac{i!}{j!}$$

The $l_{ij}$ are Stirling numbers of the first kind, also written $[i]j$ (Knuth 1973). They are defined by the equation:

$$\sum_{i=1}^j (-1)^{i-1} l_{i,j} x^i = (-1)^{i-1} l_{i,j} x + (-1)^{i-2} l_{i,j} x^2 + \ldots$$

$$-l_{j-1,j} x^{j-1} + l_{j,j} x^j = x(x-1)\ldots(x-j+1) = \frac{x!}{(x-j)!} = j! \binom{x}{j}$$

Ewens's (1972) program embodies the calculation of $w_{ij}$ for $i = 1, 2, \ldots, k$ and $j = i, i+1, \ldots, N$, where $k = $ number of species; $N = $ total number of individuals in the data being analysed. These values were stored in (part of) an array of $k$ 'rows' and $N$ 'columns' of double precision real numbers, requiring a memory allocation of

$$M_{Ewens} = 8kN$$

bytes (8 bytes per double precision real number). This value soon approaches the memory limits of many computers. On an IBM PCXT 286 (with 640K of RAM) it was found to be impossible to compile and run versions of the program allowing much more than 40 species $\times$ 400 individuals ($k > 40$ and $N > 400$).

**Speed constraint.** The program published by Ewens (1972) is quite slow. One reason for this was traced to a part of the program where a nested loop includes the evaluation of double precision logarithms, which are, by computing standards, slow to calculate. There are

$$L_{Ewens} = \frac{1}{2} \left( N - k + 1 \right) \left( N - k + 2 \right)$$

such calculations.

**PROGRAM MODIFICATIONS**

**Memory constraint.** Only $2(N-k) + 3$ of the values of $w_{ij}$ calculated are required in the remainder of the program. (These are $w_{k:0}, w_{k-1:j}$ for $k-1 \leq j \leq N-1$, and $w_{k-2:j}$ for $k-2 \leq j \leq N-2$.) Thus only these need to be held permanently in computer memory. The other values need only be held temporarily, since each 'row' of values $(w_{i,k}, w_{i,i+1}, \ldots, w_{i,N})$ is calculated from the previous row $(w_{i-1,k}, w_{i-1,i+1}, \ldots, w_{i-1,N})$ which may then be discarded. Using this technique, it was possible to reduce the memory requirement to

$$M_{G&L} = 16(N - k) + 24$$

bytes. Clearly, $M_{G&L}$ is significantly smaller than $M_{Ewens}$ for all practical values of $k$ and $N$.

**Speed constraint.** Ewens's (1972) program uses only the double precision logarithms of the integers $1, 2, \ldots, N-k+1$, but many are calculated more than once. It was found that greater efficiency is achieved by calculating the double precision logarithms of these numbers once, and storing the results in a 'look-up' table. The number of double precision logarithms calculated becomes

$$L_{G&L} = N - k + 1$$

$L_{G&L}$ is always smaller than $L_{Ewens}$; the ratio is

$$\frac{L_{G&L}}{L_{Ewens}} = \frac{2}{N - k + 2}$$

**PERFORMANCE IMPROVEMENTS**

**Memory constraint.** In FORTRAN, all memory allocations are made before compilation. With Ewens's (1972) program, it is necessary to reserve memory for an array of $kN$ double precision real numbers, i.e. $M_{Ewens} = 8kN$ bytes.
With the new version of the program, larger problems may be analysed. A parameter MEM, specified before compilation, controls the maximum size of problem. Any problem with \( k < MEM \) and \( N - k < MEM \) (and \( N \geq k > 0 \)) can be tackled, and requires a memory allocation of 16MEM bytes (2MEM double precision real numbers). Fig. 1 shows the permitted problem sizes as functions of \( k, N \) and MEM, and compares them with our old 40 x 400 maximum. It was possible to compile and run the program with MEM = 35 000, but MEM = 2000 was considered adequate for most uses. With such a value for MEM, the constraint \( k < MEM \) will be satisfied for all practical values of \( k \) and only the constraint \( N - k < MEM \) need be considered.

### Speed constraint.

Tests were made of the old and new versions of the program over the permitted ranges of values of \( k \) and \( N \). The IBM Professional FORTRAN Compiler, Version 1.00, and an IBM PCXT 286 microcomputer with numeric co-processor were used. These products provide average performance and so represent a good test of the program. The compiler utilises the numeric co-processor chip to increase the speed of floating-point arithmetic operations. This greatly improves the speed of the program.

Inspection of the revised program (see Appendix) suggested that the most time-consuming sections would include (1) the loop at Statement 10, performed \( O(k) \) times, (2) the loop at Statement 30, performed \( O(k) \) times, (3) the loop at Statement 40: \( O(N-k) \), (4) the nested loop at Statement 50: \( O(k(N-k)) \), (5) the loop at Statement 60: \( O(N-k) \), (6) the loop at Statement 70: \( O(N-k) \), and (7) the nested loop at Statement 80: \( O(N-k)^2 \). Using test times taken over the whole range of values of \( k \) and \( N \) allowed by the constraint \( MEM = 2000 \), a multiple regression was performed to discover how the time taken by the program was related to \( k \), \( N-k \), \( k(N-k) \), and \( (N-k)^2 \). This produced the formula:

\[
T_{G&L} = \frac{k}{165} + \frac{(N-k)}{369} + \frac{(N-k)^2}{7450} + \frac{k(N-k)}{1919}
\]  

where \( T_{G&L} \) = time taken in seconds. A graphical representation is given in Fig. 2. The most significant terms are those involving \( k(N-k) \) and \( (N-k)^2 \), and a regression using just these gave the slightly less accurate, but more easily calculated, formula:

\[
T_{G&L} = \frac{(N-k)^2}{7378} + \frac{k(N-k)}{1890} = \frac{(N-k)(N + k)}{7378 + 2541}
\]

These formulae (8 and 9) will vary with different computers or different compilers but give an indication of the way that the values of \( k \) and \( N \) affect run-times.

Tests on the 2 versions of the program, using the same computer and compiler, show that over the original, smaller, range of permitted values of \( k \) and \( N \) the modified version is from 3 to 10 times faster. On average, the modified version seems about 4 times as fast.

### CONCLUSIONS

The improved program allows most practical problems to be analysed routinely, with acceptable run-times on microcomputers. For example, a 40 x 400 data set which was our realistic limit on such computers now runs in something under 30 s. This greatly enhances the utility of neutral model analysis as the method of choice for diversity analysis, even with the largest practically possible data sets.
APPENDIX

Below is the FORTRAN listing of the optimized program. It was written in IBM Professional Fortran for an IBM PCXT 286 computer, but should be simple to convert for other machines or compilers. The statements most likely to require alteration are the file handling 'OPEN' statements (Lines 17 and 28) and the 'CALL GETTIM' statements which read the computer's clock (Line 25 and the 15th line after Statement 80). A 'universal' FORTRAN is used, requiring that the input and output filenames are CASDAT.DAT and CASOUT.DAT respectively; the program may be altered simply to allow the user to specify other file-names at the time the program is run.

```
C EWENS - CASWELL NEUTRAL MODEL
C ORIGINAL PROGRAM BY W. J. EWENS
C PROGRAM MODIFICATION BY NICK GOLDMAN
  BRITISH MUSEUM (NATURAL HISTORY)
  LONDON, ENGLAND
C
C SET-UP
C
PARAMETER (MEM = 2000)
C
DOUBLE PRECISION W1(MEM), W2(MEM), W3N, SIGA, SIGB, SIGC, VARINF,
  & DK, DN, XX, YY, ZZ,
  & SUM, AVINF, SDINF, V, FR, DF1, DF2
OPEN (8,ERR=995,FILE='CASDAT.DAT',STATUS='OLD')
C
READ DATA, CHECK PROBLEM SIZE, ECHO DATA TO OUTPUT
C
READ (8,*,ERR=996) N, K
IF (K.LT.3.OR.K.GT.MEM.OR.N.LT.K.OR.N.GT.K+MEM-1) GOTO 997
DN=DBLE(FLOAT(N))
READ (8,*,ERR=996) (W1(J),J=1,K)
CALL GETTIM(IH, IM, IS, I)
T=FLOAT(IS+60*(IM+60*IH))+FLOAT(I)/100.
C
OPEN (9, ERR=998, FILE='CASOUT.DAT', STATUS='NEW')
WRITE (9,1000) MEM
  FORMAT (1X///28H Caswell's Neutral ModeUlOX, 'Version 1.',15, 'by Nick Goldman, B.M.(N.H.)')
C
ZZ=0.DO
DO 10 J=1,K
  ZZ=ZZ+W1(J)
IF (ZZ.NE.DN) GOTO 999
C
WRITE (9,1010) N, K, (W1(J),J=1,K)
  FORMAT (1X///35H N = ',15,5X,'k = ',15//5Xr 'Frequencies:'/(6D12.3))
C
K1=K-1
NK1=N-K1
DK=DBLE(FLOAT(K))
C
CALCULATE H' = SUM
C
SUM=0.DO
DO 30 J=1,K
  ZZ=W1(J)/DN
  SUM=Sum-ZZ*DLOG(ZZ)
C
CALCULATE STIRLING'S NUMBERS, STORING THOSE NEEDED
IN VECTORS W1(*), W2(*) AND SCALAR W3N
C
W1(1)=1.DO
W2(1)=1.DO
W3N=1.DO
DO 40 J=2,NK1
```
APPENDIX (continued)

40 \( W_2(J) = 1.00 / \text{DBLE}(\text{FLOAT}(J)) \)
DO 50 J=2,NK1
DO 50 J=2,NK1
\( W_1(J) = W_2(J) \)
\( ZZ = \text{DBLE}(\text{FLOAT}(J+I-1)) \)
50 \( W_2(J) = ((ZZ-1.00) \cdot W_2(J-1) + \text{DBLE}(\text{FLOAT}(J)) \cdot W_1(J))/ZZ \)
\( YY = \text{DBLE}(\text{FLOAT}(K)) \)
DO 60 J=2,NK1
\( ZZ = \text{DBLE}(\text{FLOAT}(J+K1)) \)
60 \( W_3N = ((ZZ-1.00) \cdot W_3N + YY \cdot W_2(J))/ZZ \)

C

MAIN CALCULATIONS START HERE

SIGA=0.00
SIGB=0.00
SIGC=0.00
DO 70 J=1,NK1
\( YY = \text{DBLE}(\text{FLOAT}(NK1-J+1)) \)
\( ZZ = \text{DLOG}(YY) \)
SIGA=SIGA+ZZ \cdot W_2(J)
SIGB=SIGB+YY \cdot ZZ \cdot ZZ \cdot W_2(J)
70 C
CALCULATE LOOK-UP TABLE OF LOGS IN VECTOR W2(*)

W2(J)=DLOG(DBLE(\text{FLOAT}(J)))

C

CALCULATE \( E[H^1] = \text{AVINF}, SD[H^1] = \text{SDINF} \) AND NORMAL DEVIATE=V

SIGA=SIGA \cdot DK/W3N
SIGB=SIGB \cdot DK/W3N
AVINF=DLOG(DN)-SIGA/DN
DO 80 I=1,NK1
\( DO 80 J=1,NK1-I+1 \)
80 SIGC=SIGC+W2(I) \cdot W2(J) \cdot W1(NK1-I-J+2)
SIGC=SIGC \cdot DK \cdot \text{DBLE}(\text{FLOAT}(K1))/W3N
VARINF=(SIGB+SIGC-SIGA \cdot SIGA)/(DN \cdot DN)
SDINF=DSQRT(VARINF)
V=(SUM-\text{AVINF})/\text{SDINF}

C

CALCULATE F-RATIO=FR AND DEGREES OF FREEDOM=DF1, DF2

XX=DLOG(DK)
YY=SUM/XX
ZZ=AVINF/XX
XX=VARINF/(XX \cdot XX)
DF1=2.00 \cdot ZZ \cdot (ZZ \cdot (1.00-ZZ)/XX-1.00)
DF2=DF1 \cdot (1.00-ZZ)/ZZ
FR=(1.00-ZZ) \cdot YY/(ZZ \cdot (1.00-YY))
CALL GETTIM(IH,IM,IS,I)
T=FLOAT(IS+60.00 \cdot (IM+60.00 \cdot IH))/FLOAT(I)/100.00-T

C

OUTPUT OF RESULTS

WRITE (9,1020) SUM,AVINF,SDINF,V
1020 FORMAT (/// 11H 'H',11X,5HE[H'],9X,6HSD[H'],7X, & V N.D.)/4(3X,D10.3,2X))
WRITE (9,1030) FR, DF1, DF2
1030 FORMAT (//' F-ratio Degrees of freedom for F' & 3(3X,D10.3,2X))
WRITE (6,1040) T
1040 FORMAT (/5X,'Time taken: ',F6.2, ' seconds')
STOP

C

ERROR MESSAGES
APPENDIX (continued)

995  WRITE (6,1050)
1050  FORMAT ('//5X,'ERROR: Cannot open file CASDAT.DAT/',/12X,
& 'Data file not present?')
STOP
996  WRITE (6,1060)
1060  FORMAT ('//5X,'ERROR: Cannot read data/',/12X,
& 'Value of k too large?')
STOP
997  WRITE (6,1070)
1070  FORMAT ('//5X,'ERROR: Values of N and k inappropriate',
& 'for this version of the program')
STOP
998  WRITE (6,1080)
1080  FORMAT ('//5X,'ERROR: Cannot open file CASOUT.DAT/',/12X,
& 'Old output file present?')
STOP
999  WRITE (6,1090)
1090  FORMAT ('//5X,'ERROR: Sum of data read not equal to N/',/12X,
& 'N or k typed incorrectly?/12X,
& 'Continuing using CALCULATED value of N')
WRITE (9,1090)
DN=ZZ
N=INT(DN)
GOT0 20
END

INPUT FILE - CASDAT.DAT: For a sample containing 10 species and 100 individuals. The species are shown ranked in order of abundance in this example for clarity, but this is not essential.

100 10
45 30 10 6 3 2 1 1 1 1

OUTPUT FILE - CASOUT.DAT: After an introductory message, the output contains the data read from the input file followed by the values of $H'$, $E[H']$, $SD[H']$ and $V$, and the $F$-ratio and degrees of freedom for an $F$-test. In this example, the $F$-ratio of 0.706 does not imply a significant deviation from neutrality.

Caswell's Neutral Model
Version 1. 2000
by Nick Goldman, B.M. (N.H.)

$N = 100$  
$k = 10$

Frequencies:
\begin{align*}
.450D+02 & .300D+02 & .100D+02 & .600D+01 & .300D+01 & .200D+01 \\
.100D+01 & .160D+01 & .100D+01 & .100D+01 & & \\
.149D+01 & .166D+01 & .230D+00 & .750D+00 & & \\
\end{align*}

$F$-ratio  
Degrees of freedom for $F$
\begin{align*}
.706D+00 & .275D+02 & .107D+02 \\
\end{align*}

LITERATURE CITED


Platt, H. M., Lambshead, P. J. D. (1985). Neutral model analy-


This article was submitted to the editor; it was accepted for printing on September 30, 1988