Journal of Chromatography, 172 (1979) 57–66

CHROM. 11,653

GENERAL METHOD FOR COMPUTER MEASUREMENT OF COMPLEX CHROMATOGRAMS

GRAHAM D. BACON

Department of Genetics, University of Edinburgh, King's Buildings, West Mains Road, Edinburgh EH9 3JN (Great Britain)

(Received September 26th, 1978)

SUMMARY

Usual computerized methods for measuring peak areas and associated computations on chromatograms require relatively highly reproducible retention times. They are prone to mis-identification of substances in chromatography of physiological fluids. The method described was developed for amino acid analyses of physiological mouse tissues. It tolerates noisy output, drifting baselines, variable retention times and unidentified peaks.

INTRODUCTION

Most computerized chromatographs in use involve electronic integrators or on-line mini-computers. They identify peaks by retention time, sometimes measured on previous runs of standard mixtures processed in the same computer run[1,2]. These methods are suitable for, *e.g.*, protein hydrolysates with reasonably reproducible retention times and steady baselines. We screen mouse tissues (plasma liver, urine) for aminoacidopathies, and the complex chromatograms, particularly those from urine samples, may contain up to 50 ninhydrin-positive peaks. The presence of unexpectedly large peaks may affect the resolution of nearby peaks because of concentration effects. These problems were partially overcome by mechanically re-editing integrator paper tape output and inserting amino acid names[3–5]. We required a more flexible approach.

The rationale behind the method reported here is as follows. Few users of computing integrators linked to the output of chart recorders are happy to accept the printed calculations without reference to the visual recorded output. Principally the "visual check" is to eliminate artifacts of either the analytical or the computing systems. Persons with both skill and experience will therefore handle the recorder output at some time. This time/cost must be added to the nominal computing time/cost.

Our method accepts this handling which takes place before any computation

is carried out. It is used to specify a number of parameters, principally the naming of peaks, their approximate positions (distance/time) if required, and references to overlapping peaks. This is achieved by having on the recorder chart output a separate channel which gives a visual event mark simultaneously recording this on the output tape. It is thus possible to identify a substance by its position, shape and relation to other peaks, a feat difficult for a machine recognition system, and link it accurately to the digitized information from the analytical output.

During the development of our screening programme we accumulated data on inbred strains of mice for comparison with suspected mutant mice. We thus needed a method for comparing measured amino acid concentrations with those retrieved from a data store. It was important to be able to update the data store easily.

This paper describes the software in outline, such that a chromatogram analyst could either write his own software along these lines, but suiting his own individual needs, or get a computer scientist to write it for him.

THE HARDWARE

Any chromatographic data produced on a chart recorder can be used. The recorder must have at least two channels, one for the chromatographic output and another for the monitor line (see below). Thus, for instance, with the commonly used three channel recorder on an amino acid analyzer, it will be necessary to sacrifice, for example, the short flow-cell channel. Extra requirements are a data logger, a paper tape punch, a keyboard with paper tape output, a custom-made monitor (see below) and access to a computer.

Data logging

The chromatogram output (from amino acid analyzer colorimeters in our case) goes in parallel to the recorder and a Solartron data logger, which can scan up to twenty channels, which in turn drives an Addo paper tape punch. The data logger produces a four digit output representing the output voltage of the detector system, followed by a newline character. About 1800 lines of output are produced in a single 12-h run.

Monitor

This is not to be confused with the computer software of the same name. The monitor is a specially made electronic device which generates a steady d.c. output voltage. This output is connected in parallel to the data logger, and thence to the paper tape punch, and to a recorder channel. It also has an input from the data logger which enables the monitor to count the number of scans on its output. Its output to the data logger is normally inhibited, preventing it reaching the paper tape punch, but after every hundred scans it activates this output, producing a line on the paper tape with four extra digits, and simultaneously, fractionally increases its voltage to the recorder, producing a step in the otherwise straight horizontal line. Thus the steps on the recorder trace exactly align with the hundreds of lines on the paper tape. The monitor also marks the end of each run with a string of + characters on the paper tape to enable the tapes from several runs to be visually separated. This facility is

activated by the master clock on our amino acid analyzer. Circuit details are available from the authors.

Instruction tapes (see below) are punched on a Teletype keyboard paper tape punch or on a Ferranti free-scan digitiser and manually spliced onto the data tapes. These are then sent to an ICL 4-75 computer at the Edinburgh Regional Computer Centre (ERCC), and are usually dealt with on-line through a local terminal of a time-shared computing system.

OPERATION

The peaks on the chart are identified and named by the analyst himself. The section of paper tape (eventually to become a magnetic disc file) corresponding to the peak in question on the chart is identified by the position of the peak on the chart relative to the stepped monitor line on the chart, each step representing completion of 100 lines (scans) on the paper tape. Fractions of 100 may be obtained roughly by using a glass grid placed on the chart. For each peak, coded information, representing the name and position of the peak, sometimes the approximate positions of the tops of multiple peaks, and the type of calculation required are punched onto an instruction tape.

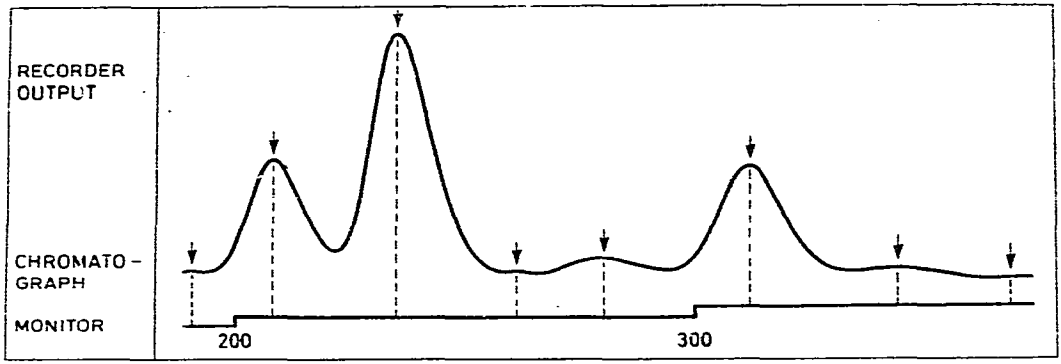
Recently we have developed the use of a Ferranti free-scan digitiser for producing instruction tapes. The chart is laid on the digitiser table and a moveable cursor placed on the chart. Having zeroed the displays at an arbitrary origin, the cursor continuously displays the x–y co-ordinates of its position. The machine is programmed to output the x co-ordinate to a paper tape punch when a button is pressed. The machine also has a keyboard for manually punching characters. Calibration is provided by outputting the x co-ordinates of the steps in the monitor line, then peak position parameters are obtained by movement of the cursor. Again peak names are keyed in at the appropriate times.

Other instructions on this tape, for the purposes of our particular application, include a key to sample numbers, tissues and genotypes, and a coded instruction nominating a data bank for comparison. A diagram of part of a chart is shown in Fig. 1, which also shows how its instruction tape would be coded. An example of a full output is shown in Fig. 2. The output, is headed with the run date, sample number, tissue and genotype. This is followed by a list of amino acids and unidentified peaks (*e.g.* PAA, PAB) as presented on the instruction tape, with their concentrations and intermediate calculations (area, ratio of areas of the peak and the internal standard, ratio $\times$ colour value and dilution factor), and with a percentage comparison with a nominated mouse strain and tissue held in a "library".

The whole operation is faster than measuring the peaks manually. It is less tedious, more variable work then manual measurement and it is far less error prone. The lengthiest part of the operation is preparing the instruction tapes. This has been mollified by using the digitiser.

THE SOFTWARE

All of the software is written in IMP, the local Edinburgh language. It is, of course, possible to write a similar program in FORTRAN or ALGOL.



NAME	AAA	BBB	CCC	DDD	EEE
START	190		265		
APPROX. TOP	208	235	280	312	346
END			265		370
CALCULATION TYPE	2	3	2	" TRIPLE PEAK "	3
INSTRUCTION LINES	AAA. BBB 190 208 235 265 / CCC. DDD. EEE 265 280 312 346 370				

Fig. 1. Diagram of a possible recorder output with the monitor line (upper part) and coding of peak names and position parameters into instruction lines (lower part).

The software is in two files. The first one contains the main program (MOUSEIO). The second, backup file is a collection of external routines, accessed direct from an on-line console, or from MOUSEIO, which consist of lists of colour values, and inbred mouse strain data, stored in OWNARRAYS, and of STREAM definitions for loading the program with a nominated data file and for nominating output devices (usually line printer or teletype). Thus colour values etc. can be updated, using an EDITOR programme and the routines recompiled without having to recompile the main program.

The main program routines

Create data. Instead of using the slow routine READ to get the large instruction and data file into the central processor, the ERCC external routine SMADDR is used which efficiently packs the whole file into a one dimensional byte integer array of single characters called DATA.

No checks are carried out by SMADDR so further routines are required.

Check instructions. This locates in the array DATA the positions of the start and finish of the instructions marked on the original instruction tape by + and * symbols, respectively. It then "reads" the sample numbers etc. in the upper block. Then the instructions proper are searched for parity errors, the presence of letters and numbers in their correct respective positions and the correct number of parameters on each line. Any faults are reprinted on a selected output device.

Check data. This locates the data block in the DATA array, it being marked

```
COMMAND:MOUSE(N231276,.TT,.TT)
STREAM02      .TT
STREAM01      .TT
SMFILE10      N231276
INSTRUCTIONS CHECK
END OF INSTRUCTIONS REACHED.
   0 FAULTS
DATA CHECK
END OF DATA REACHED
   0 FAULTS



STREAM02      .TT
STREAM01      .TT
SMFILE10      N231276
MOPVA(M)  DILUTION FACTOR=    0.200   COMPARISON=     3

AA.        %COMP   MMOLES/ML   RATIO*DF     RATIO        AREA        TOP

STHAGP        20   0.200000   0.200000   1.000001   5.990967        PN   2
TAU           70   0.190877   0.139337   0.696684   4.173811   3         1
URE           62   140.186029   0.100477   0.502387   3.009784             2
MET          261   0.090050   0.063148   0.315742   1.891602             3
ISL          149   0.072277   0.072786   0.363932   2.180308   -2        1
LEU          189   0.182253   0.264557   1.322786   7.924766    2        1
PRC                0.011210   0.011210   0.056049   0.335788    2   PN   1
HIS          114   0.059083   0.110045   0.550223   3.296371        PN   1
THE           96   0.004195   0.008139   0.040697   0.243817    0   PN   1
GYA           67   0.011471   0.014671   0.073353   0.439453        PN   3
PRA                0.081917   0.081917   0.409587   2.453823    1   PN   1
ASP           56   0.012739   0.011249   0.056243   0.336949        PN   2
THR          198   0.186880   0.297107   1.485533   8.899781    2        1
SER          182   0.310562   0.275712   1.378559   8.258897    2   PN   1
GLU          300   0.556060   0.721781   3.608908   21.620858       PN   2
CIT                0.074607   0.067893   0.339463   2.033711    1        1
GLY                0.347712   0.613899   3.069494   18.389245       PN   1
ALA          116   0.446709   0.471262   2.356312   14.116592   -2   PN   1
PAR                0.040200   0.040200   0.201002   1.204195    -1       1
CYS           23   0.011638   0.007516   0.037579   0.225138    4   PN   1
VAL          193   0.192090   0.244141   1.220705   7.313199    4        1
TYR           96   0.070974   0.095305   0.476526   2.854850    2   PN   1
PHE          120   0.060400   0.098037   0.490183   2.936673    1   PN   1
NH3          103   0.375942   0.379893   1.899466   11.379639       PN   3
ORN          131   0.077024   0.143541   0.717706   4.299754    0   PN   1
LYS          151   0.436938   0.584064   2.920323   17.495568   1        1
TRY           77   0.014250   0.018819   0.094095   0.563721        PN   3


BASES
 100  100  100  100  100  100  100  100  100  100  100  100  100  100  100
 100  100  100  100  100  100  100  100  100  100   99   99  100  100
 100  100  100  100  100  100  100  100  100  100
```

Fig. 2. Example of some output produced by the program. It identifies the sample and lists the results of amino acids in the order in which they were presented on the instruction tape.

by + characters at each end. Each "line" section of the array bounded by newline characters is checked for parity errors and correct number of digits per line. It also checks that the extra four monitor digits are present on every hundredth line. Faults

produce an output on a selected device giving the line number, the complete line of digits and a * symbol beneath the offending character. The program stops after completing CHECK DATA if there are any faults. Faults can be rapidly corrected on line using the EMAS EDITOR program.

Decode instruction line. Each line of instructions is "read" in turn and the variables "amino acid name", "start", "top 1", "top 2", etc. "end", and "calculation type", are allocated. The "start", "end" and "top" variables are the numbers of the lines on the original data tape.

Findpeak. The section of the DATA array between "start" and "end", is repacked into a real array called PEAK, the groups of four byte integers being converted first to real numbers (3527 becoming three thousand five hundred and twenty seven) and then to negative logarithms (proportional to concentration: Beer's law).

Area measurements, or calculation types. The analyst may make these as simple or as complex as he likes. The operations basic to all types are to take a mean baseline (mean of five points from "start"), to summate the points on the peak (*i.e.* the numbers in the array PEAK) and to subtract the mean base × the length of the peak. To avoid errors due to baseline noise, first the mean of the first five numbers in PEAK is calculated. Then the point with the greatest deviation from this mean is eliminated and the mean recalculated on the remaining four points.

(1) Calculation type 1. This routine takes a mean baseline at each side of the peak, summates all the points in the PEAK array, subtracts the rectangle formed by the length of the peak (between "start" and "end") and the lower of the two baselines, then subtracts the triangle formed by the projection of the lower baseline, the difference in heights of the baseline and the sloping base of the peak (see Fig. 3).

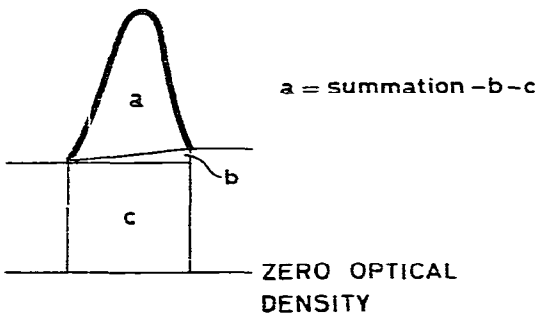


Fig. 3. Performance of a calculation type 1 on whole peaks.

This routine determines the area under any shape of peak or multiple peaks without making any assumptions of triangularity as in manual area measurements.

(2) Calculation types 2 and 3. These routines measure a peak with only one baseline, due to fusion with another peak. The single mean baseline is determined as before. The routine FINDTOP is called (see below) and summation is performed from the base to the top. The base rectangle is subtracted from the area, and the half-peak area is multiplied by a scaling factor (see Fig. 4).

Using calculation types 2 (left half of peak) and 3 (right half of peak), on a set of clean peaks with baselines on both sides gives an empirical mean correction factor due to peak asymmetry. Thus: whole area = left hand side area × 2.05;

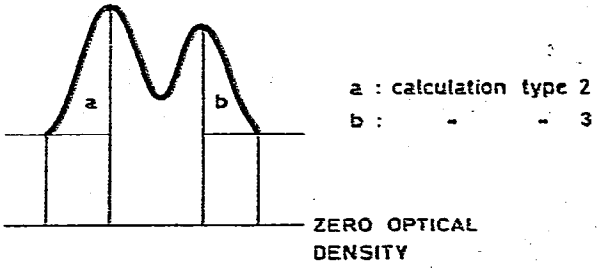
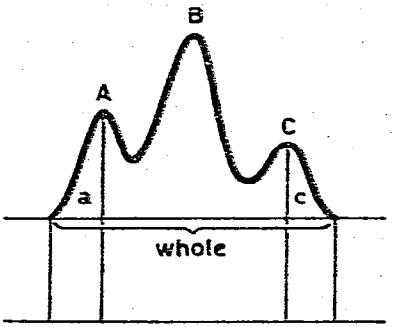


Fig. 4. Illustration of where calculation types 2 and 3 are used.

whole area = right hand side area × 1.95. The constants 2.05 and 1.95 are stored in the backup file for ease of updating. Calculation types 2 and 3 are called automatically by the main block of the program when it encounters an amino acid name consisting of two names connected by a comma: *e.g.* AAA, BBB in the instruction line (Figs. 1 and 2).

(3) Triple peaks. The area under a peak having no baselines can be determined as follows (see Fig. 5). First the area of the whole group of three peaks is measured with a type 1 calculation. Then the areas of peaks A and C are measured by type 2 and 3 calculations, respectively. Then B = whole —A —C. This task is performed automatically by the main block of the program when it encounters an amino acid name consisting of three names connected by comas: *e.g.* CCC, DDD, EEE in the instruction line (Fig. 1).



whole : calculation type 1
a : " " 2
c : " " 3

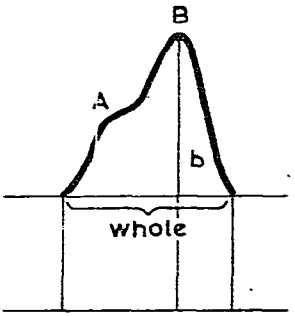
Area of B =
$$\text{whole} - (2.05 \times a) - (1.95 \times c)$$

Fig. 5. Determination of the middle peak of a triple peak.

(4) Poorly resolved peaks. It is sometimes useful to have an estimate of a peak area even when it is poorly resolved (Fig. 6) rather than no measurement at all. Calculation types 4 or 5 are used.

In this example a calculation type 5 measures the combined area (A + B) by a type 1 calculation and subtracts from it 1.95 times the "half" area of B (calculation type 3) to give the approximate area of a shoulder on the right hand side of a peak. For very close peaks, the position of the maximum of B will be shifted to the right and hence a smaller scaling factor than 1.95 will be appropriate (calculation type 3A). These "calculation types" are similar to other published methods[6].

Findtop. Calculation types 2 and 3 require the top of a peak to be located in the PEAK array. Merely choosing the highest point may cause large errors in noisy



whole : calculation type 1
b : " " 3

Area of A =
whole − (1·95 × b)

Fig. 6. Calculation type 5 for shoulder peaks. It is assumed that substance A does not contribute to the area of the right hand side of peak B.

chromatograms. FINDTOP creates a moving window of five points in the PEAK array. The window is moved one position at a time through the array and determines the sign of the slope along these five points. The regression equation for this special case simplifies as follows for five consecutive points a–e equally spaced on the x axis:

$$n = 2(a - e) + b - c$$

which appears to be similar to the Li-function of Back *et al.*[7]. The sign of n is the sign of the slope. A top is defined as the mid position of the window when two consecutive window positions to the left both give a positive slope and two to the right a negative slope. As a further safeguard the instruction line contains the approximate position of the top as measured on the chart. Only ten lines (scans or points) on either side of this approximate top are searched for a top by FINDTOP. Failure to find a top here aborts that line of instructions. The output (see Fig. 2) informs the analyst of the difference between the approximate and real tops, such that gross discrepancies can be discarded.

Smooth. All calculation types call the routine SMOOTH after the tops have been located but before summation. Our routine only removes grossly aberrent points. If noise were defined as a point on a slope whose numerical value did not lie between the values of its neighbours, the value of the offending point could be replaced by the mean of the neighbouring values. An example, is shown in Fig. 7a, where the window is travelling "up" a slope and encounters a point noisy in the "upward" direction. On the other hand Fig. 7b shows that noise in the "downward" direction would cause the wrong point to be smoothed. This method is therefore not used.

A cruder but more reliable definition of noise is, (for a line with positive slope and the window moving in the upward direction) a point which is lower than a point two places behind or higher than a point two places ahead. Fig. 8a shows that "upward" noise on an "up" slope is corrected as before (point d is smoothed because it is higher than point f). In the case of "downward" noise on an "up" slope, Fig. 8b illustrates that while SMOOTH is centred on point b this point is faulted because it is higher than point d. However, the correction, placing it at the mean of a and c, hardly moves point b. When the window moves on and centres on point d (Fig. 8c),

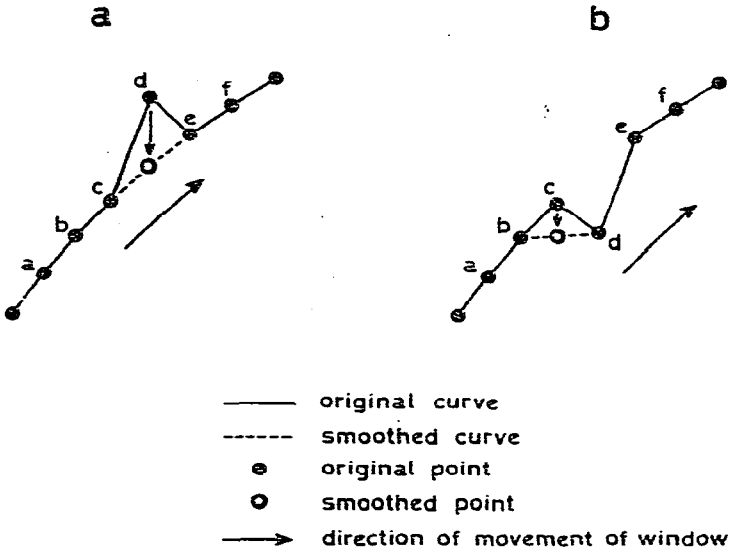


Fig. 7. Smoothing of a noisy point, defined as having a numerical value outside the range of its immediate neighbours .The noisy point would be smoothed correctly if it is "upward" noise on an "upward" slope (a), but in the case of "downward" noise on an "upward" slope (b) the wrong point is smoothed. This definition is therefore not used.

this point is correctly smoothed because it is lower than point b. More sophisticated methods such as fitting sets of 5–7 points to a quadratic equation[8] could be used, but our method is found to be reliable and safe.

SMOOTH is halted at maxima and minima as located by FINDTOP.

Main block of program. This calls routines in the appropriate order, signals faults, performs the rest of the calculation (concentration = (area of peak/area of internal standard) × colour value × dilution factor), performs subtractions for calculation types 4 and 5 and the triple peak type, and calls output routines.

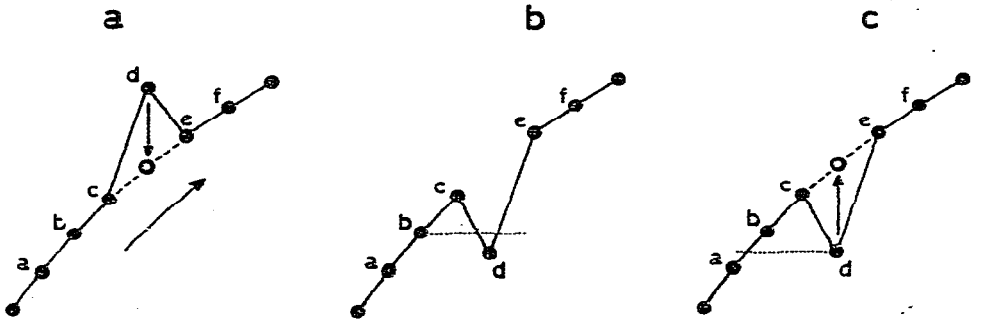


Fig. 8. The SMOOTH routine using a different definition of noise than that used for Fig. 7, described in the text, correctly smooths upward noise (a). With downward noise, correct smoothing takes place (c) at the expense of a negligible movement of a good point (b).

CONCLUSIONS

This program and its precursors have been in use in the laboratory regularly for two years. Technicians have been trained to operate all parts of the system. The saving in time over manual measurement is about three-fold and about four-fold when using the digitiser to produce instruction tapes. A typical chromatogram of four samples each with 34 peaks takes us about 270 min to measure manually. Using this computer method the time is reduced to about 100 min. Use of the digitiser to produce instruction tapes reduces this further to about 70 min. Technicians find that they can work with this system for longer continuous periods than with manual peak measurements. All conceivable faults are trapped by the program and signalled on the output (*e.g.* stray points grossly affecting the baseline mean, failure to find a top in the correct region) without interrupting the rest of the run. The only known cause of error is in the initial peak identification by the analyst. This condition is considered preferable to that in which identification rests in the hands of the integrator programmer.

The advantages of producing paper tape output off line have been that important tapes can be re-run as the program became more refined, that the programmer has a large central processor at his disposal and that output can be filed and further processed for output on, for instance, the ERCC graphplotter.

At this stage of development of this system, the rate-limiting step is the production of instruction tapes even using a digitiser. This is mollified in our application by the fact that up to four chromatograph runs are processed in a single batch. We use a six-channel Kent recorder accepting the output from four synthronized chromatograms. As stated, the data logger can scan up to twenty channels. Possibly the use of a video display system with interactive graphic would further improve the speed.

ACKNOWLEDGEMENTS

The general methodology of the approach and the idea of the monitor was devised by Dr. J. A. Burns, whose help is gratefully acknowledged. The advice of R. R. McLeod of Edinburgh Regional Computing Centre is also gratefully acknowledged.

REFERENCES

1 J. N. Owen and A. D. Dale, *J. Chromatogr.*, 107 (1975) 207.
2 M. A. Fox, *J. Chromatogr.*, 89 (1974) 61.
3 H. D. Spitz, G. Henyon and J. N. Sivertson, *J. Chromatogr.*, 68 (1972) 111.
4 S. E. Møller, *J. Chromatogr.*, 104 (1975) 63.
5 J. Lamaziere and R. Miglierina, *J. Chromatogr.*, 106 (1975) 191.
6 A. W. Boyne and W. R. H. Duncan, *J. Lipid Res.*, 11 (1970) 293.
7 H. L. Back, P. J. Buttery and K. Gregson, *J. Chromatogr.*, 68 (1972) 103.
8 R. Taylor and M. G. Davis, *Anal. Biochem.*, 51 (1973) 180.