CABIOS

Vol.7. no.1. 1991
Pages 89 – 93

MetaModel: a program for modelling and control analysis of metabolic pathways on the IBM PC and compatibles

Athel Cornish-Bowden and Jan-Hendrik S.Hofmeyr[1]

Abstract

MetaModel is a user-friendly program for calculating steady-state fluxes and metabolite concentrations of metabolic systems on the IBM PC and compatible computers. For any steady state that is obtained, one can then calculate a matrix of elasticity coefficients at that steady state, or a matrix of control and response coefficients. It thus offers a simple way to calculate the control structure of a pathway: it provides not only an educational tool that allows the student to verify empirically the classic summation relationships of metabolic control analysis but also a research tool for addressing 'what if?' questions about the behaviour of metabolic systems. Results can not only be printed or stored in a file, but can also be written to a special file that can be read by popular spreadsheet programs, thereby giving access to rapid, flexible and powerful methods for subsequent analysis and plotting of these results.

Introduction

Several theoretical approaches to the understanding of metabolic control are currently under active discussion in the literature: the most important of these are metabolic control analysis, developed from the work of Kacser and Burns (1973) and of Heinrich and Rapoport (1974), biochemical systems theory (Savageau, 1976), and the theory of Crabtree and Newsholme (1987); many recent contributions to these approaches may be found in Cornish-Bowden and Cárdenas (1990). There has been comparatively little attempt to harmonize these approaches, and they have tended to be expressed in terms that exaggerate their differences, but a study by Groen and Westerhoff (1990) suggests a much greater degree of equivalence between them than is apparent to the casual eye. However, their large algebraic content makes it difficult to appreciate the extent to which these theories agree with one another and with older ideas about metabolic control derived mainly from experimental observation. On the other hand, thorough experimental investigation of the control structure of metabolic systems is almost impossible because one cannot make the changes in parameters that are needed to address particular hypotheses: how, for example, does one make an experimental test of the idea that the allosteric properties of phosphofructokinase are essential to the control of glycolysis in the living rat? Can one obtain a rat that is normal in every respect except that its liver contains a form of phosphofructokinase with all of the normal properties except for a modified elasticity towards ATP?

Understandable (and well justified) reluctance to put total trust in an algebraic analysis of the control structure of a metabolic system, coupled with the near impossibility of investigating it experimentally, argues strongly for the need for efficient methods of simulating the behaviour of such systems in the computer. Programs for doing this have existed for many years, starting with the heroic work of Garfinkel, Hess and co-workers (Chance *et al.*, 1960; Garfinkel and Hess, 1964), but these earlier programs were written by experts for experts to use on mainframe computers far removed from modern practice. How far removed may be judged from two points noted by Garfinkel and Hess (1964): 'the Univac I and II computers cannot easily represent quantities greater than 1', and 'the first time the complete glycolysis system was tried on the Univac II, ½ hour of computer time was needed to simulate 75 ms of real time'. Access to the Univac II was restricted to one session per week, and the work described in Chance *et al.* (1960) required > 500 h of Univac I time.

Recently, various metabolic modelling programs have been written for use on IBM PCs and compatibles, but their use has been described in the research literature only in the form of brief notes as an appendix to a symposium (Letellier *et al.*, 1990), though in some cases the principles of the algorithms have been described (Hofmeyer and van der Merwe, 1986; Mazat and Reder, 1988; Irvine and Savageau, 1990). Here we shall describe the use of MetaModel 2.0, which is derived from MetaModel, a FORTRAN program written for mainframe computers by Cornish-Bowden, and METAMOD, originally written by Hofmeyr in BASIC for the BBC microcomputer and described by Hofmeyr and van der Merwe (1986). [This last paper and some others refer to a chapter by Cornish-Bowden (1986): to avoid fruitless searches we note that the book for which this chapter was accepted in 1985 never appeared, and will not appear.] In many respects MetaModel 2.0 can be regarded as an updated version of METAMOD, and for that reason we shall not repeat information given by Hofmeyr and van der Merwe (1986); however, we note that some of this information is quite important and the interested reader should

Centre de Biochimie et de Biologie Moléculaire, Centre National de la Recherche Scientifique, 31 chemin Joseph-Aiguier, Boîte Postale 71, 13402 Marseille Cedex 9, France and [1]Department of Biochemistry, University of Stellenbosch, Stellenbosch 7600, South Africa

refer to the earlier paper for details. More general background for modern methods of simulating metabolic behaviour is given by Hofmeyr (1986).

System and methods

MetaModel 2.0 was written in Turbo Pascal 5.0, and developed on computers compatible with the IBM XT and AT; it will run under MS-DOS 2.11 or later on any computer of this type with sufficient memory. We have used MetaModel 2 only on machines with hard disks, but in principle it should run, possibly with some inconvenience, without a hard disk. The current version requires 191 kbytes of memory, and although this requirement may increase somewhat as the program is updated it is unlikely to change very much. When running on an XT-type machine with 512 kbytes of memory and no other programs resident in memory, MetaModel 2.0 has sufficient memory for far more enzymes than the maximum of 15 that the screen formatting constraints allow. With modern machines, memory shortage is unlikely unless several other programs are resident concurrently. In this event the program will still run for models requiring < 15 enzymes, because enzyme information is stored in dynamic variables that are created with pointers as required.

MetaModel 2.0 makes no use of graphics routines, and we do not anticipate any difficulty in running it with any common graphics card. It was developed on machines with Hercules monochrome cards (both true Hercules and Hercules-compatible). It has no requirement for colour but it allows for colour in the sense that the display colours can be adjusted during operation and the choices can be saved so that they are used automatically in subsequent runs. In our experience such adjustment is likely to be necessary even with monochrome systems because supposedly compatible machines are not by any means equivalent in practice. The cursor scan lines likewise differ between machines, and cursor settings that are correct for one machine may be intolerable on another: MetaModel 2.0 allows these settings to be adjusted and saved.

The lack of built-in graphics capability was a deliberate choice, avoiding most of the problems associated with incompatibilities between graphics cards, and offering a substantial saving in memory requirements. In addition, we believe that most users will find the capacity to produce ASCII output (delimited by commas and quotes in the default state, but these can be altered) capable of being processed by commercial spreadsheet packages, such as Quattro, Lotus 1-2-3, etc., more valuable than built-in but limited graphical processing would be.

MetaModel does not require access to a virtual (RAM) disk but can use one if it exists for more rapid operation. If the installation program detects a disk drive with drive letter in the range D–H it requests permission to use a designated drive as a scratch pad.

MetaModel 2.0 is available on a single 5.25 inch 360 kbyte disk. In addition to the program itself, this also includes numerous files used for providing context-sensitive help, and one or more sample data files. The help files are stored as separate text files: this allows them to be edited separately from the program itself, and to be deleted when disk space is limited. The program itself displays all captions and so on in English, but some help files exist also in French and Spanish: users interested in these should indicate it when ordering the program because there is not sufficient space on the disk to include all of the files that exist. MetaModel 2.0 is accompanied by a printed user's manual and is available for a small charge to cover copying and distribution costs. It may be transferred to other machines, given away, copied, etc., without restriction, except that it may not be incorporated into another package distributed for payment.

Implementation

MetaModel was designed as a user-friendly program intended originally for teaching the ideas of control analysis to students with some knowledge of elementary enzyme kinetics but no previous exposure to computers. The earlier mainframe and BBC microcomputer versions were used for teaching such students over a period of several years, and proved to be learned very rapidly. It is fully menu-driven, but experienced users can greatly accelerate operation, avoiding the need to return endlessly to the same menus to rekey the same information: this is achieved by means of a Quick change menu entry that allows various kinds of repetitive calculations to be predefined.

A mouse is not needed to run the program, and no mouse functions have been consciously included. Nonetheless, observation of its behaviour on a machine with a Microsoft mouse indicates not only that it can be run without difficulty when such a mouse is installed, but also (to our surprise) that one can easily use the mouse to select menu entries.

Main menu

The main menu is illustrated in Figure 1. Of the choices available at the start, Find model is used to recover a model saved previously, Input new model to define a new model, Delete model to delete a model from the list of saved models, Options to adjust the screen colours or cursor scan lines, vary the characteristics of the built-in editor, or carry out various other tasks that will be rarely needed, and Exit to leave the program.

Once a model has been entered or recovered and fixed concentrations have been defined, other menu items become available, as also illustrated in Figure 1. The most important of these is Steady state, which allows calculations to begin, but one may also mention Summary mode and Quick change, which are both useful when proceeding through a repetitive series of calculations: Summary mode allows the results of such calculations to be displayed or printed in a table with comparable results arranged in the same columns, whereas Quick change allows fixed concentrations to be changed without proceeding through the normal sequence of menus.

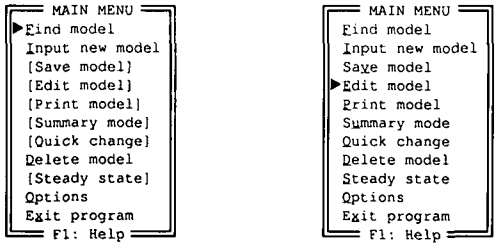

Fig. 1. The main menu as it appears when starting the program (left) and as it appears after a model has been defined (right). Options that are not available because a necessary precondition has not been fulfilled are shown in square brackets. Menu items can be selected either by pressing the highlighted letters (here underlined), by using the arrow keys to move the triangle to the appropriate line and pressing ENTER, or by means of a Microsoft mouse.

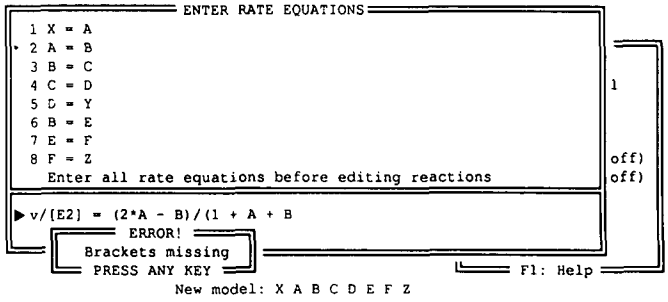

Fig. 2. Screen during input of the model SEQFB described by Hofmeyr and van der Merwe (1986). All reactions have been defined, and the user has attempted to enter the second rate equation, but this has resulted in an error message because the parenthesis in the denominator has not been closed. The bottom line lists all of the metabolites in the model as they become known to the program.

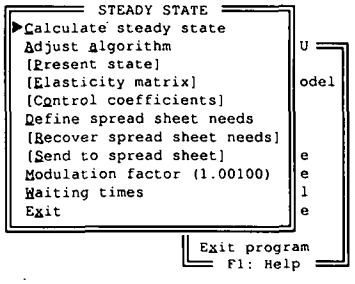

Fig. 3. Steady-state menu as it appears before a steady state has been calculated. In this menu unavailable items are shown in square brackets but the keys to activate them are still highlighted: this allows MetaModel to generate a message indicating why they are not available.

Inputting a new model

MetaModel 2.0 is designed to the greatest extent possible to allow metabolic schemes to be input in the form most natural for writing chemical equations. To define the hexokinase reaction, for example, it is sufficient to type

$$\text{Glc} + \text{ATP} = \text{G6P} + \text{ADP}$$

and rate equations are likewise typed in a style corresponding to normal kinetic practice, e.g. a rate expression might be written as follows:

$$(5\star A - 0.1\star B)/(1 + 2\star A + B + 0.3\star A\star B + 0.2\star C\wedge 2).$$

Left-hand sides of rate equations appear as prompts and are not typed by the user; the normal rules of algebra are followed, in the sense that any legal rate expression must be algebraically correct; however, the rules are more restrictive, in that multiplication signs (★) may not be omitted, brackets cannot be nested, and only a single division sign (/) is permitted: A/5, for example, will normally have to be written as 0.2★A. Exponents, when needed, are preceded by∧ as in the last term in the example above.

Input errors are trapped as early as possible: on pressing the incorrect key if one attempts to type a character such as & that cannot occur in a rate expression at all, or on attempting to enter the line if a legal character appears in an illegal position, such as a minus sign in the denominator of a rate expression. Figure 2 illustrates a typical screen as it might appear after attempting to enter a line containing an error. Once a model is defined it can be used for steady-state calculations; it can also be edited, saved or recalled in a later run for further analysis.

Metabolite names do not have to be listed separately: it is sufficient to include a name in a reaction or a rate equation to make it known to MetaModel. Any metabolites that appear in rate expressions but not in reactions are assumed to be external effectors. MetaModel compiles its own list of metabolites and displays it at the bottom of the screen during input (see Figure 2) and editing of pathways, so that any unintended names that result from typing errors and so on can be recognized immediately. Pascal purists may feel that this is an unfortunate departure from the discipline of requiring names to be defined before they are used; however, MetaModel is a program for users rather than for programmers, and user friendliness has taken precedence over user discipline.

Metabolites may be defined as fixed (normally appropriate for reservoirs, sinks and external effectors), variable (most intermediates) or dependent. This last category applies to one intermediate in each conservation equation that one might want to define. For example, to define the adenine nucleotides as a system in which the adenine moiety is conserved, one could enter an equation

$$\text{ATP} + \text{ADP} + \text{AMP} = 5$$

and then set the status of one of the three nucleotides (it does not matter which) as dependent. More information about metabolite status and conservation equations may be found in Hofmeyr and van der Merwe (1986).

Steady-state menu

The steady-state menu is illustrated in Figure 3 as it appears before a steady state has been calculated. The principal function available is Calculate steady state, which causes all rates

and intermediate concentrations to be calculated for the steady state corresponding to the particular fixed concentrations that have been specified. Once a steady state has been calculated, the additional functions Elasticity matrix and Control coefficients become available. In each case a complete matrix is calculated, and the control matrix includes response coefficients as well as control coefficients. The program recognizes if some enzymes are linked in a linear chain (so that their fluxes are necessarily equal in the steady state), and does not include redundant columns in the control matrix. Nonetheless, it can easily happen that the matrix is wider than the screen: in this event only seven columns are displayed simultaneously but the others can be reached by means of the arrow keys.

Export of output to spreadsheet programs

By means of the entry Define spreadsheet needs in the steady-state menu, one can define the variables that are to be exported to a file that can be read by commercial spreadsheet programs such as Quattro or Lotus 1-2-3. As the length of line that such programs can accept is limited, it is not normally possible to export all calculated results in this way, especially as they do not simply ignore characters after the last one they can read, but either refuse the entire line or even crash. Provided that the requirements of the spreadsheet program are set correctly, MetaModel will avoid creating lines that are too long.

As the files produced in this way are intended to be read by other programs, they are not very legible to the eye: entries are separated only by commas and no spaces are used. For other purposes one may prefer the results of successive calculations to be displayed legibly on the screen or printed in regular columns: this can be done by selecting Summary from the main menu to define the columns required.

Algorithm

The principles of the algorithm are virtually unchanged from those described for algorithm SOLVE by Hofmeyr and van der Merwe (1986), and thus do not require repetition here. Suffice it to remark that Cornish-Bowden's alternative algorithm CHANGE has not been implemented in the current program, as we have found that when SOLVE is well tuned it gives superior performance under virtually all conditions, so that there is no advantage in burdening the program with an unnecessary alternative. The current version of SOLVE does not allow concentrations to become negative, and so the logarithmic transformation allowed as an option in METAMOD is redundant and has been removed.

Calculation of the steady state for the model SEQFB defined in Hofmeyr and van der Merwe (1986) on a Sanyo 17 (an AT-type machine with mathematical co-processor and a clock speed of 8 MHz) required seven iterations and ~ 3.5 s, for the same concentrations ($X = 10$, $Y = 2$ and $Z = 1$) of the fixed metabolites as used in the earlier paper, and starting from a default concentration of 1 for each intermediate. The number of iterations is essentially the same as before, but the execution time is faster by a factor of 20, presumably reflecting the change of machine. Calculation of the 72 elasticities required ~ 1 s, and calculation of the 72 distinct control coefficients and 27 distinct response coefficients required 18 s.

The normal version of MetaModel does not use a 8087/80287/80387 co-processor, because the small saving in time (~ 10% when calculating the 99 control and response coefficients) on machines with a co-processor hardly justifies the 110% increase in time implied for machines without a co-processor: on a Thomson TO16 (an XT-type machine without co-processor and with a clock speed of 10 MHz) calculation of the 99 control and response coefficients required 49 s with the normal version, but 101 s with the version that detects and uses or emulates a co-processor. However, this version can be supplied if specifically requested.

Discussion

The implementation of MetaModel on the IBM PC and compatibles makes it easy for users with no previous experience with computers to set up models of metabolic pathways, calculate their steady states and analyse their control structures. Moreover, the capacity to export files of results to spreadsheet programs permits a highly flexible approach to subsequent analysis and graphical display. The program thus offers a flexible tool for studying the behaviour of metabolic pathways. It may also be used for teaching the principles of metabolic control analysis, allowing students to verify fundamentals such as the summation relationships by treating them as empirical laws to be observed rather than simply as results from only partially understood algebra.

References

Chance,B., Garfinkel,D., Higgins,J. and Hess,B. (1960) Metabolic control mechanisms. V. A solution for the equations representing interactions between glycolysis and respiration in ascites tumor cells. *J. Biol. Chem.*, **235**, 2426–2439.

Cornish-Bowden,A. and Cárdenas,M.L. (1990) *Control of Metabolic Processes*. Plenum Press, New York.

Crabtree,B. and Newsholme,E.A. (1987) The derivation and interpretation of control coefficients. *Biochem. J.*, **247**, 113–120.

Garfinkel,D. and Hess,B. (1964) Metabolic control mechanisms. VII. A detailed computer model of the glycolytic pathway in ascites cells. *J. Biol. Chem.*, **239**, 971–983.

Groen,A.K. and Westerhoff,H.V. (1990) Modern control theories: a consumers' test. In Cornish-Bowden,A. and Cárdenas,M.L. (eds), *Control of Metabolic Processes*. Plenum Press, New York, pp. 101–118.

Heinrich,R. and Rapoport,T.A. (1974) A linear steady-state treatment of enzymatic chains. *Eur. J. Biochem.*, **42**, 89–102.

Hofmeyr,J.-H.S. (1986) Steady-state modelling of metabolic pathways: a guide for the prospective simulator. *Comput. Applic. Biosci.*, **2**, 5–11.

Hofmeyr,J.-H.S. and van der Merwe,K.J. (1986) METAMOD: software for steady-state modelling and control analysis of metabolic pathways on the BBC microcomputer. *Comput. Applic. Biosci.*, **2**, 243–249.

Irvine,D.H. and Savageau,M.A. (1990) Efficient solution of ordinary nonlinear

differential equations expressed in S-system canonical form. *SIAM J. Numer. Anal.*, **27**, 704–735.
Kacser,H. and Burns,J.A. (1973) The control of flux. *Symp. Soc. Exp. Biol.*, **27**, 65–104.
Letellier,T., Mazat,J.-P., Irvine,D.H., Savageau,M.A., Voit,E.O., Mendes,P., Hofmeyr,J.-H.S., Cornish-Bowden,A. and Atkinson,D.E. (1990) Appendix B, Computer Programs. In Cornish-Bowden,A. and Cárdenas,M.L. (eds), *Control of Metabolic Processes.* Plenum Press, New York, pp. 433–436.
Mazat,J.-P. and Reder,C. (1988) Aspects géometriques de la théorie du contrôle du métabolisme. In Mazat,J.-P. and Reder,C. (eds), *Contrôle du Métabolisme Cellulaire.* Bordeaux, pp. 77–96.
Savageau,M.A. (1976) *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology.* Addison-Wesley, Reading, MA.

Received on August 27, 1990; accepted on October 17, 1990