STEADY STATES OF GENERAL MULTI-ENZYME NETWORKS AND THEIR ASSOCIATED PROPERTIES. COMPUTATIONAL APPROACHES

J. A. BURNS
Edinburgh, Scotland

This paper discusses the theoretical basis for the determination of the steady-state characteristics of a multi-enzyme network, given initial values for the enzyme and metabolite concentrations, and rate equations for each enzyme. The computation depends only to a small extent on integrating the rate equations, mainly on adjustment of the matrix of fluxes through each component of the system, until they are equal to each other within a prescribed tolerance limit.

The program has been devised to be as flexible as possible to permit the user to investigate the effects of changing the concentration or kinetic properties of an individual enzyme. Alternatively, the concentrations of enzymes required to maintain a given flux may be computed. The program will deal with a network of up to about 30 enzymes.

1. Introduction

What follows is a brief outline of our attempts in Edinburgh [1,2] to implement a program capable of dealing with a variety of questions centred on the steady state properties of multi-enzyme networks.

Our interest in a program specifically for the steady state is due to several reasons. Firstly that the experimental measurements do not deal with transients, the methodology being to observe the response of selected fluxes and metabolites, measured during steady logarithmic growth of *Neurospora,* to the substitution of alternative enzymic forms. Secondly, since the steady state(s) of a network are, if they exist, independent of the initial conditions, it becomes possible to define theoretical quantities of interest such as "sensitivity-coefficients" and "optimal allocation of enzyme". Finally, it should be noted that programs dealing with the complete dynamic problem [3,4] are very time-consuming; by utilising the special properties of the steady state we may hope to investigate a fruitful set of defined theoretical questions in the modest amount of computer time available in our environment.

2. Theoretical

Consider a multi-enzyme system consisting of N enzymes having concentrations E_i and n variable metabolic pools having concentrations S_j; the ith enzyme carries at any instant a net-flux/unit volume, F_i, determined by a rate expression of the type

$$F_i = F_i(E_i, P_{i1}, ..., P_{ir}, S_a, S_b, S_c, ...) , \qquad (1)$$

where F_i is an algebraic expression of the King-Altman form. The $P_{i1}, ..., P_{ir}$ are the r kinetic constants necessary to describe the behaviour of the ith enzyme E_i in the presence of metabolites $S_a, S_b, S_c, ...$ having strong interactions with it, this includes of course allosteric type interactions. Eq. (1) assumes enzymic intermediates to be in a steady state. Under almost all circumstances eq. (1) can be written as

$$F_i = E_i [f_i(P_{i1}, ..., P_{ir}, S_a, S_b, S_c, ...)] , \qquad (1a)$$

so that for fixed pool concentrations

$$F_i \propto E_i .$$

 North-Holland Publishing Company – Amsterdam

The instantaneous rate of change of the jth pool is a linear sum of the fluxes connected to it, thus

$$\dot{S}_j = \Sigma\, \lambda_{ji} F_i \,. \qquad (2)$$

Most of the stoichiometric coefficients λ_{ji} are of course equal to zero. The steady state solution of eq. (2) is $\bar{S}_j$ such that all $\dot{S}_j = 0$, this solution will be correct even though the King-Altman assumptions may distort the dynamics of the approach to $\bar{S}_j$. In fact so long as we are interested only in the final steady-state solution we can alter eq. (2) so that we get to the same solution by completely altered "false dynamics".

An example of this is

$$\dot{S}_j = \varphi_j(S_1, ..., S_n)\, \Sigma\, \lambda_{ji} F_i \,, \qquad (2a)$$

where $\varphi_j > 0$ for all arguments.

It is clear that if all $\bar{S}_j$ are to be independent of each other and of the initial conditions S_j^0, then we require that:

i) There be more enzymes than pools, i.e., $N > n$.
ii) The rows in the matrix $[\lambda_{ji}]$ of eq. (2) be linearly independent.

The following theory considers only the case where both these conditions are satisfied:

i) is considered reasonable by Garfinkel (discussion at Summer School);
ii) amounts to abolishing artificial situations, such as, for instance, the sum of ATP and ADP depending only on the initial conditions, and replacing them with a situation which allows the metabolites to move independently of each other.

The $\bar{S}_j$ are functions of all parameters such as P_{ik} (the kth parameter belonging to enzyme i), E_i (the quantity of enzyme i), X_P (the constant concentration of the Pth external metabolite), etc. The form of the function is unknown but the computer can be used to explore the dependence of $\bar{S}_j$ on the parameter values. We can observe the relative influence of different parameters on $\bar{S}_j$ around any point in parameter space by computing, for instance,

$$C_{E_i}^{\bar{S}_j} = \frac{E_i}{\bar{S}_j} \frac{\partial \bar{S}_j}{\partial E_i} \qquad (3)$$

which measured the influence of the ith enzyme on the jth pool. Alternatively we can find

$$C_{P_{ik}}^{\bar{S}_j} = \frac{P_{ik}}{\bar{S}_j} \frac{\partial \bar{S}_j}{\partial P_{ik}}$$

which measures the influence of the kth parameter of the ith enzyme upon the jth pool.

This can be extended to consider the influence of a parameter on any function of the $\bar{S}_j$, of particular interest being the steady state fluxes $\bar{F}_l$ which are known functions of the $\bar{S}_j$, thus

$$\bar{F}_l = \bar{F}_l(E_l, P_{l1}, ..., P_{lr}, \bar{S}_a, \bar{S}_b, \bar{S}_c, ...)$$

and

$$C_{E_i}^{\bar{F}_l} = \frac{E_i}{\bar{F}_l} \frac{\partial \bar{F}_l}{\partial E_i} = \frac{E_i}{\bar{F}_l} \left\{ \frac{\partial \bar{F}_l}{\partial \bar{S}_a} \frac{\partial \bar{S}_a}{\partial E_i} + \frac{\partial \bar{F}_l}{\partial \bar{S}_b} \frac{\partial \bar{S}_b}{\partial E_i} + ... \right\} .$$

The interest in a quantity such as $C_{E_i}^{\bar{F}_l}$ is that if we evaluate it for all enzymes E_i, $i = 1, ..., N$, we can see which one, if any, has the major "rate-controlling" effect on the flux $\bar{F}_l$. Quantities of the type C_y^x we call sensitivity coefficients.

3. Aims of program

The aim of the program is to receive a set of rate equations (1) together with rules for combining them to form the rate of change of the pools (2), (1) and (2) being written in ordinary algebraic notation.

The user should be able to manipulate parameters, enzyme quantities, external metabolites, initial conditions etc., and cause the steady-state solutions to be found, together with the sensitivity coefficient of any function of the steady state with respect to any parameter, i.e., C_y^x.

A number of other features seem important and have been included, in particular the ability to switch on or off suitably defined allosteric interactions, the ability to command a systematic exploration of parameter space, the ability to easily alter the enzyme network and also to have available at any time a number of alternative networks for different studies.

4. Mathematical basis of program

One general way of getting into the neighbourhood of a steady-state solution $\bar{S}_j$ is to integrate the differential equations of motion (2), thus simulating the natural movement of the S_j. However, a criterion for when to stop, or indeed certainty as to whether one is in the region of a steady state at all, are not simple matters.

The criterion of closeness to the desired steady state is as follows; after every half-hour of simulation (metabolic system time) the values of $(N-n)$ of the F_i fluxes are accepted and the values F' which the remaining n must have in order to satisfy eqs. (2) with all $\dot{S}_j = 0$ are calculated. In general these will be different from their existing values and

$$\rho = \Sigma \left| \frac{F' - F}{F} \right|$$

is a measure of the distance of a constructed exact steady state, with slightly altered enzyme concentrations, from the desired exact steady state with the prescribed enzyme concentrations. When the rate equations can be expressed in the form (1a) ρ measures the sum of the fractional enzyme changes needed to move from the constructed to the prescribed steady state. Thys, if $\rho < 0.001$ we can say that we have found a mathematically exact steady state such that no enzyme differs from its prescribed value by more than 1 part in 1000.

Simulation, which is costly, is only employed while $\rho > 0.20$. As soon as $\rho < 0.20$ the matrix $[\partial \bar{S}_j / \partial E_i]$ is formed, by numerical approximation, for the constructed steady state and is used to estimate the $\bar{S}_j$ obtaining at the correct enzyme values. This is repeated until ρ is reduced to the required accuracy. This process, similar to Newton-Raphson iteration, converges rapidly for all systems so far tried and it is not necessary to recompute $[\partial \bar{S}_j / \partial E_i]$ very frequently. The sensitivity coefficients C_y^x can all be found from the matrix without appreciable further computation.

The time to recompute the $[\partial \bar{S}_j / \partial E_i]$ matrix as we move about the parameter space is not great, compared with the time spent on simulation, for systems up to 30 enzymes. Also the necessary matrix inversion is numerically well-behaved provided that we already have a reasonable estimate for $[\partial \bar{S}_j / \partial E_i]$, which will be the case if parameter movements are not too violent. Initial estimates of both $\bar{S}_j$ and $[\partial \bar{S}_j / \partial E_i]$ are more costly, being formed by simulation. They are computed only once and kept available should the current values be lost. Clearly the integration method employed can have a low accuracy and still suffice. At present 4th order Kutta-Merson is used because it is convenient and reliable. For "stiff" systems considerable improvement would result from the employment of a more suitable integration method.

5. Program design

If a "compile more code" facility is available in conjunction with a good high level language the most flexible method is to compile a pack which contains all the necessary routines, name lists, etc., and use this as a special compiler. Input is then merely a routine containing a source language description of the network in ordinary algebraic notation followed by the required commands (routine calls). This method allows the user all the flexibility of the available high level language and enables him to perform subsidiary computations and suitable tabular or graphical outputs without difficulty.

Unfortunately such facilities are not available in Edinburgh at the present time, and we have had to use, as a distinct "second best", a data-directed program technique. This consists of compiling the pack and the network description and holding the resultant binary program on magnetic tape; input for the program consists of a simple command language interspersed with the necessary numeric data. The commands, which can easily be added to, cause appropriate routines to be called. This has proved robust and simple to use for the non-programmer and seems a good technique for the more primitive computer environments, particularly since there is practically no overhead before genuine computation starts.

Another aspect which seems important is that, since numerical techniques are never infallible, adequate provision must be made for the program to recover itself should it encounter difficulties. For instance, if a steady state has not been found within a given computer time, the program should output a diagnostic showing which stage of the computation is

giving difficulty and then continue from the next independent starting point.

6. Discussion

Using the techniques outlined in this report a program was written to deal with networks containing up to 30 enzymes. The program is heavily machine dependent, in order to gain in efficiency, and is therefore unfortunately "not for export"; it is hoped, however, that the techniques discussed here will prove useful. It is notoriously difficult to discuss "efficiency" without a "benchmark" network but for a 15-enzyme system it takes of the order of 10 sec of computation to find a steady state on a machine having a 15 μsec time for floating point multiplication.

References

[1] H.Kacser, Symposium on Quantitative Biology, ed. A.Locker (Springer-Verlag, Heidelberg), in press.
[2] J.A.Burns, Symposium on Quantitative Biology, ed. A.Locker (Springer-Verlag, Heidelberg), in press.
[3] E.M.Chance, Computers and Biomed. Res. 1 (1967) 251.
[4] D.Garfinkel, FEBS Letters 2, Suppl. (1969) S9.