

## A computer program for the analysis of protein complex formation

Steven Lay and Dennis Bray<sup>1,2</sup>

### Abstract

**Motivation:** We needed an efficient way to explore the binding reactions leading to protein complexes of known composition and structure.

**Results:** A new program is described that allows the user to define a set of protein elements and to link these elements into an oligomeric 'ball-and-stick' assembly in a graphical interface. Once the structure of the oligomer has been defined, the program then employs a novel algorithm to deduce the binding reactions and intermediate complexes needed to make the oligomer from its starting protein components. The program also finds the equilibrium state of the system, using either default starting concentrations and  $K_d$  values or data supplied by the user.

**Availability:** OLIGO runs on any colour Apple Macintosh and is available without charge by anonymous FTP from: <ftp://sequoia.amtp.cam.ac.uk/pub/BCT/oligo-10.sea.hqx>

**Contact:** E-mail: [s.w.lay@damtp.cam.ac.uk](mailto:s.w.lay@damtp.cam.ac.uk) and [d.bray@zoo.cam.ac.uk](mailto:d.bray@zoo.cam.ac.uk)

### Introduction

Many, if not most, protein molecules in a living cell are part of multiprotein assemblies held together by non-covalent bonds. Well-characterized examples include multimeric enzymes, proteasomes, receptor complexes, focal contacts, and the protein assemblies responsible for DNA replication (Alberts *et al.*, 1994; Tjian and Maniatis, 1994; Mochly-Rosen, 1995; Pawson, 1995; Faux and Scott, 1996). Although such complexes have a definite structure and composition, their formation depends on a series of diffusion-limited binding reactions influenced by the concentrations of participating protein molecules and their binding affinities for each other. Therefore, in order to know how much of the complete (functionally active) complex will be formed under any specified set of conditions, we need to know what binding steps lead to that complex and what are their individual  $K_d$  values. Unfortunately, such detailed information is available for few, if any, intracellular protein complexes.

What may be done, however, is to enumerate in theory a list of possible binding steps that could, in principle, lead from the individual unassociated protein molecules to the complete, functionally active complex. A list of this kind may then be used as a basis to explore the dependency of complex formation on the nature of the bindings steps and their individual  $K_d$  values. Producing such a list of binding steps and predicting their steady state, given specific values of starting concentrations and  $K_d$  values, by numerical integration is a straightforward albeit tedious and time-consuming exercise. We therefore felt that it would be useful to automate the procedure and for this reason developed the computer program, called OLIGO, described in this report. Our program runs on Apple Macintosh machines and allows the user to enter in symbolic form the structure of a small complex of proteins (containing up to eight or so protein species) in a graphical interface, and then to deduce the binding steps and intermediate protein complexes needed to produce that complex. When provided with suitable starting concentrations and  $K_d$  values, the OLIGO program also predicts the steady-state concentrations of the various species, including, most importantly, the concentration of the structurally complete complex.

### System

OLIGO runs on any colour Apple Macintosh computer with System 7.0 or higher and was developed with the Metrowerks C/C++ compiler. It runs in native mode on Macintosh Power PC machines.

### Algorithms

The OLIGO program allows the user to enter a set of proteins, and use these to construct an oligomer, modelled by a simple graph. In this context, a graph simply means a collection of nodes (proteins) and edges, which loosely represent a binding relationship between proteins. For the purposes of computation, the term 'bond' refers to these binding relationships and the generic term 'species' includes all oligomers and sub-oligomers, including those with only one protein and (hence) no bonds.

The program allows the user to enter the bonds in one of two formats. A normal, dissociable bond may be broken by the system when searching for binding relationships between

Department of Applied Mathematics and Physics, Silver Street, Cambridge CB3 9EJ and <sup>1</sup>Department of Zoology, Downing Street, Cambridge CB2 3EJ, UK

<sup>2</sup>To whom correspondence should be addressed

proteins and sub-oligomers. A fixed bond may not be broken by the system. An oligomer made of two similar proteins joined by a fixed bond effectively represents a dimer.

When the user has finished constructing an oligomer in the graphical window, the program checks that the resulting structure is connected by traversing the oligomer recursively, marking each protein visited. A legal oligomer (one represented by a connected graph) can then be deconstructed to form a list of sub-species and binding reactions. Deconstruction is achieved by means of a recursive procedure which breaks up the oligomer into smaller pieces, adding each sub-species in turn to the list before accepting the original oligomer.

#### *Oligomer list*

The list of sub-species that can be formed from the oligomeric structure specified by the user is obtained by the deconstruction algorithm as follows. To add a species, say *S*, to the list, it is first tested against the existing list for conflicts. Any conflicts found are handled in the manner described below. The structure of *S* is then traversed, splitting it into two by extracting the smallest possible sub-species containing the current protein (say *P*) at each step, leaving a 'remainder' species (say *Q*) which may or may not be legal (i.e. connected). Both *P* and *Q* may be added to the list of species provided there are no conflicts. At each step, this procedure is recursively invoked for *P* and, if legal, *Q*. Finally, *S* is added to the list of oligomers.

The 'conflicts' referred to above occur if a particular species produced in the deconstruction process, say *S*, has the same number, composition and topological arrangement of proteins as an existing member of the species list, say *C*. Various possibilities now occur, each of which is handled differently.

- (i) If *S* and *C* have an identical arrangement of bonds, then sub-species *S* is ignored (it is already in the list).
- (ii) If *S* has fewer bonds than *C*, but its proteins are arranged in such a way that *S* can be converted to *C* by adding one bond, then sub-species *S* is ignored. The biochemical justification for this rule is that bonds will form between two proteins close together if they can.
- (iii) Conversely, if *S* differs from *C* simply by having one additional bond then, for the reason just enunciated, the new species (*S*) is added in place of the smaller species already in the list (*C*).
- (iv) If *S* and *C* are identical except that a fixed bond in one is matched by a dissociable bond in the second, then the conflict is reported, and the deconstruction procedure is terminated.

#### *Reaction list*

The list of reactions used by OLIGO is generated

automatically from the list of species and is updated each time the latter changes. A simple exhaustive search for all the possible interactions between species is then performed (the search space is generally fairly small because only combinations that have as their product one of the species existing on the list are included). The set of reactions generated by the program are a maximal set—the actual rate constants will normally be interrelated due to free-energy considerations. These dependencies are automatically calculated each time the user specifies a value for a rate constant (effectively 'pinning' that value). A rate constant whose value is implied by these dependencies may not be changed by the user.

The list of reactions is then used to equilibrate the system by a customized algorithm that traverses the list of reactions, solving each individual reaction in turn (as if run in isolation to equilibrium). This procedure is repeated until the reaction product of each reaction is within 0.1% of its final value, when the integration is terminated and the concentrations of species on the list displayed. Some sets of binding equilibria were also examined by numerical integration using the Livermore solver for ordinary differential equations, with automatic method switching for stiff and non-stiff problems (LSODA) (Petzold, 1983).

### **Implementation**

#### *Tar complex*

The use of OLIGO will be illustrated by following the steps needed to analyse a cluster of proteins termed the Tar complex which is involved in bacterial chemotaxis. The core of this oligomer contains three species of protein molecules: the aspartate receptor Tar (*T*), the cytoplasmic protein CheW (*W*), and the cytoplasmic histidine kinase CheA (*A*). It has the empirical formula TTWWAA (Gegner and Dahlquist, 1991; Gegner *et al.*, 1992).

On opening the OLIGO program, the user selects *New protein* from the *Project* menu (or enters *command N*) and completes the dialogue box. A single-letter symbol and a colour are selected for the new protein (e.g. '*T*', red) and there is an option to enter a longer name or descriptive title, such as 'Tar receptor'. As each protein is entered, it adds to a list of proteins species displayed in the project window. The project may be saved at any time in the usual way by entering *command S* and designating a name, which then appears in the project window (Figure 1).

When the three proteins, *T*, *W* and *A*, have been entered in this manner, the user selects *New Oligo* from the *Project* menu, thereby opening a window in which the new oligomeric complex will be built. A toolbar in this window displays set of proteins (in this case *T*, *W* and *A*) as filled circles of the assigned colour in a scrollable box. The Tar receptor is selected by scrolling to the symbol for *T* using the small

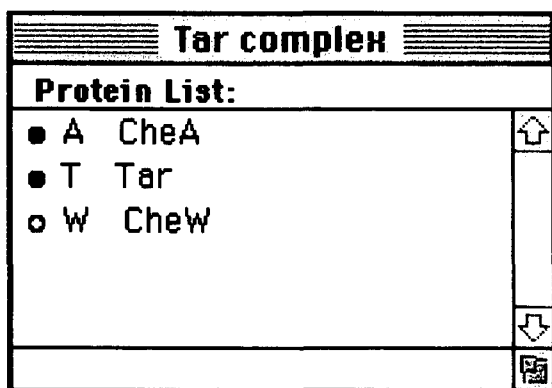


Fig. 1. Protein species used to construct the Tar complex. The name and associated colour of each protein are designated in a dialogue box (not shown).

arrows beneath the symbol, and then clicking on this symbol by means of the mouse. The cursor changes to a 'potato cutter' which can be used to deposit two symbols representing the Tar proteins at desired locations in the window. The same procedure is then employed to deposit two CheWs and two CheAs. The positions of protein symbols in the window can be adjusted by means of a pointer and a hand tool.

The topological structure of the Tar complex is then defined by two bonding tools. One tool establishes a 'fixed' or permanent bond between a pair of proteins that will remain together during the formation of the oligomer. The second bonding tool forms a 'dissociable' or weak bond between two proteins. In either case, the user first selects the bonding tool by clicking with the mouse button and then moves the cursor to the first protein. The mouse button is clicked and held down, and the cursor dragged across to the second protein and released. A visible line created by this procedure represents the bond between the two proteins and may be either a thick line (representing a fixed bond) or a thin line (dissociable bond).

Both Tar and CheA are tightly bound dimers under normal conditions so the fixed bonding tool is used to produce T-T and A-A. Links between T and W, W and W, and W and A are then established by the dissociable bonding tool. Bonds are preserved as the proteins are moved within the window by means of the pointer, thereby allowing the oligomer to be shaped to the desired configuration (Figure 2).

The user signals completion of a satisfactory oligomeric structure by selecting *Add Oligo* in the *Project* menu. The cursor now changes to a watch face while the program deconstructs the Tar complex into its binding steps by the procedure described above, and then returns to the normal cursor. (The Tar complex is relatively simple and deconstruction takes less than a second on most Macintoshes.) The outcome of the deconstruction process may then be viewed in one of two windows, the *Oligo List* window and the *Reaction List* window, which are opened by making the appropriate selection from the *Project* menu.

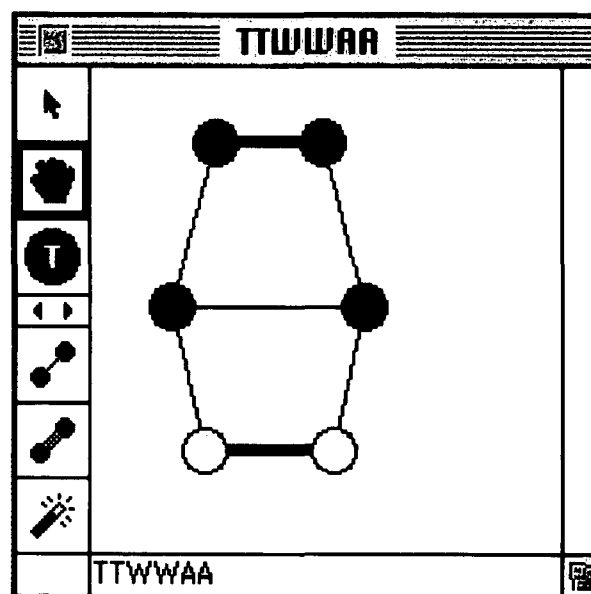
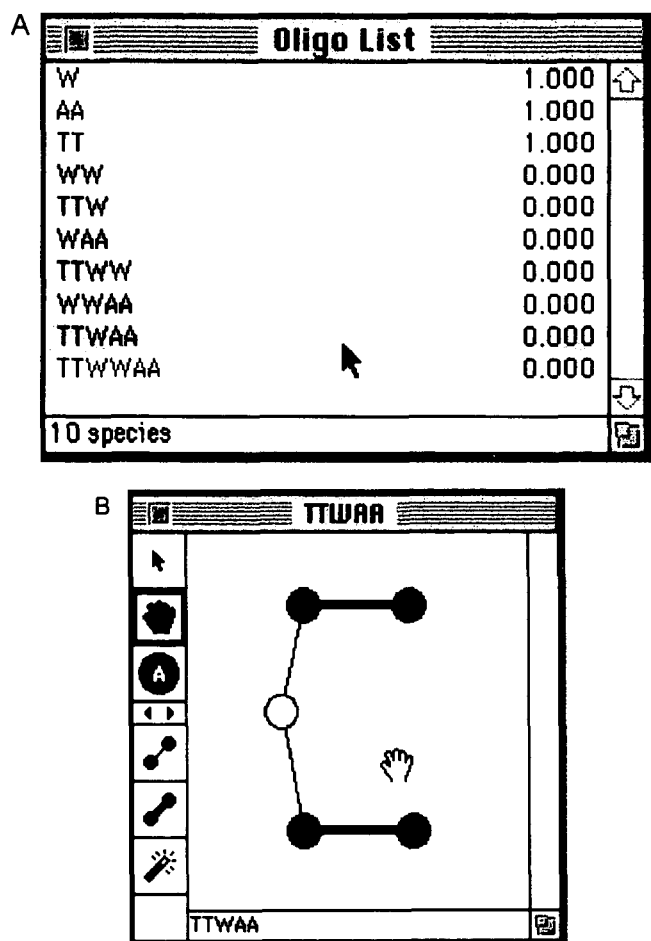


Fig. 2. Window displaying the structure of the Tar complex. The oligomer was assembled by clicking and dragging symbols representing the three protein species from the toolbar, and then linking them together by means of the two bonding tools, as described in the text.

The *Oligo List* window displays a list of all of the protein species produced in the deconstruction, from the starting proteins (shown in blue) to the target oligomeric complex (shown in red) (Figure 3A). Note that the starting form of the aspartate receptor is represented in this list as T-T and that of CheA as A-A because of the fixed bonds made previously between these pairs of proteins. The composition and linkage of any protein complex in this list may be examined by double-clicking on its symbol to open a separate window (Figure 3B). Each protein species is listed with a value representing its concentration in micromolar. By default, this is 1  $\mu\text{M}$  for each of the initial (non-bound) proteins (so that W appears as 1  $\mu\text{M}$ , and T-T and A-A each appear as 0.5  $\mu\text{M}$ ) and zero for all other species. Any of these starting values may be altered by clicking on the entry and then selecting *Concentration* from the *Edit* menu. The name of the oligomer may also be changed by selecting *Rename* in this menu: in the present instance, we change 'AATTWW' to 'TTWWAA' so as to conform to previous usage (Bray and Bourret, 1995).

The *Reaction List* window displays a list of binding equilibria, each corresponding to a different dissociable bond in the oligomer (Figure 4). Reactions are arranged in increasing order of product size and shown with default values of  $K_d$ . Any individual  $K_d$  value may then be changed by clicking its entry in this table and selecting *Kd* in the *Edit* menu. As  $K_d$  values are altered in this manner, certain values become constrained for thermodynamic reasons and are automatically locked by the program. Thus, of the 13 reactions leading to TTWWAA, only seven are independent variables and the remaining six are determined by the others.



**Fig. 3.** Intermediate oligomeric complexes. (A) A list of all protein species involved in the formation of the Tar complex is shown, from the starting proteins to the final complex. The list is shown prior to equilibration, with the starting proteins all at  $1 \mu\text{M}$  concentration. As described in the text, both the names of the proteins and their starting concentrations can be changed by the user. (B) Structure of one of the intermediate complexes. The species TTWAA was selected by double clicking [as shown in (A)], causing it to be displayed in a second window.

Once starting concentrations and  $K_d$  values have been entered, then the equilibrium state of the system can be derived by selecting *Equilibrate* in the *Project* window (or entering *command B*). A dialogue box then displays the progress of the numerical integration as the 'distance' of the state of the system from true equilibrium. The latter is obtained by calculating the distance from equilibrium of each reaction in the list using the present concentrations of reactant and product and the  $K_d$  specified by the user. Equilibration is complete when all reactions in the list concentrations are  $<0.1\%$  of their equilibrium value and for TTWWAA typically takes 2–5 s, depending on the machine. Completion of integration is signalled by an audible beep and the display, in the *Protein* window, of an updated set of concentrations. For the starting values displayed in Figures 3A and 4, the equilibrium values obtained were TT ( $0.495 \mu\text{M}$ ), WW

( $0.495 \mu\text{M}$ ), TTWAA ( $0.011 \mu\text{M}$ ) and the Tar complex, TTWWAA ( $0.494 \mu\text{M}$ ), with all other species being  $<1 \text{ nM}$ .

#### Scope and accuracy

OLIGO has been used to generate lists of binding reactions and intermediate species for  $>100$  simple oligomers, including linear, branched and cyclic structures, a selection of which is illustrated in Figure 5. In some of these protein complexes, more than one intermediate with the same empirical formula is generated in the deconstruction. These are designated #1, #2, and so on, in the Reaction list, and their structures may be displayed in the usual way. As the size and complexity of the oligomer increase, so the time required to deconstruct its binding reactions and to find the equilibrium state of the set of reactions becomes longer. Thus, the 'Necker cube'  $a_6$  oligomer shown in Figure 5 required 20 s to deconstruct on a Power PC, with equilibration requiring 2–3 s. These times escalate rapidly on addition of further proteins to the complex but, so far as we are aware, this limitation is solely that of computational time. We see no reason why the same deconstruction and integration algorithms should not be used on larger, faster machines to analyse larger and more complicated oligomeric complexes.

The accuracy of the integration process is computed automatically for each of the reactions by the procedure described above and the largest error (expressed as a percentage deviation of the reaction product from a true equilibrium concentration) is displayed continuously in the integration dialogue box, thereby allowing the course of integration to be monitored. For several small oligomers, the values achieved by the OLIGO program have also been compared directly to the values obtained by LSODA (Petzold, 1983) with essentially no difference between the computed values. Note that the use of LSODA in this context requires substantially greater work on the part of the user, since it necessitates the calculation of the binding reactions and then manual entry of differential equations for each of the intermediate species.

#### Discussion

The need for a simple, easy-to-use, way to enumerate the binding reactions leading to small protein complexes arose in a study of bacterial chemotaxis (Bray and Bourret, 1995). The phenotype of certain mutants could only be explained by postulating an effect on the assembly of the Tar complex mentioned above. Testing this hypothesis required the tedious elaboration by hand of binding reactions leading to different postulated structures for the Tar complex, a process that can now be performed in a fraction of the time on a computer using the OLIGO program. Although the Tar complex is presently the best understood protein complex of its kind, it will not be long before others are characterized to an

Reaction Window	
W + W $\rightleftharpoons$ WW	1.00E-9
TT + W $\rightleftharpoons$ TTW	1.00E-9
AA + W $\rightleftharpoons$ WAA	1.00E-9
TT + WW $\rightleftharpoons$ TTWW	1.00E-12
TTW + W $\rightleftharpoons$ TTWW	1.00E-12
AA + WW $\rightleftharpoons$ WWAA	1.00E-12
WAA + W $\rightleftharpoons$ WWAA	1.00E-12
TT + WAA $\rightleftharpoons$ TTWAA	1.00E-9
TTW + AA $\rightleftharpoons$ TTWAA	1.00E-9
TT + WWAA $\rightleftharpoons$ TTWWAA	1.00E-12
TTW + WAA $\rightleftharpoons$ TTWWAA	1.00E-15
TTWAA + W $\rightleftharpoons$ TTWWAA	1.00E-15
TTWW + AA $\rightleftharpoons$ TTWWAA	1.00E-12

13 reactions

Fig. 4. List of binding reactions needed to make the Tar complex. Reactions are listed in order of size of their reacting species, with the final 'target' oligomer highlighted in red. The  $K_d$  value of each binding reaction is indicated in molar units to the right of each reaction and may be changed by the user, as described in the text.

equivalent level of detail. At that stage, we anticipate that the OLIGO program will become an invaluable tool to analyse their binding reactions and intermediate complexes.

So far as we are aware, the deconstruction algorithm employed by OLIGO to generate binding reactions and intermediate species is without precedent. The second step—that of finding the steady-state equilibrium of a system once the reactions and their  $K_d$  values and starting concentrations have been defined—could in principle be achieved by a number of existing methods. The most obvious approach is to use a standard numerical integration procedure, and we found it useful to employ the LSODA package for specific problems. However, the problem at hand—that of finding the steady state of a set of solely binding steps of the form  $A + B = AB$ —is highly stereotyped and tends to involve rate constants that differ greatly in magnitude, which therefore makes numerical integration extremely slow. Moreover, since we did not need to determine progression over time of the system, we devised and incorporated into the OLIGO program a simple algorithm that rapidly finds the equilibrium position of the reactions by an iterative process of solving each reaction equilibrium in turn. An alternative approach that could be used to tackle this problem was employed previously to deduce the steady state of ionic equilibria in solution (Storer and Cornish-Bowden, 1976; Perrin, 1966).

OLIGO is an easy-to-use, robust program designed for a specific task: the enumeration of the set of binding reactions needed to make a specific oligomeric complex and calculation of the steady state of these reactions. In scope and application, it may be regarded as a prototype which is offered here to readers in order to encourage their interest and

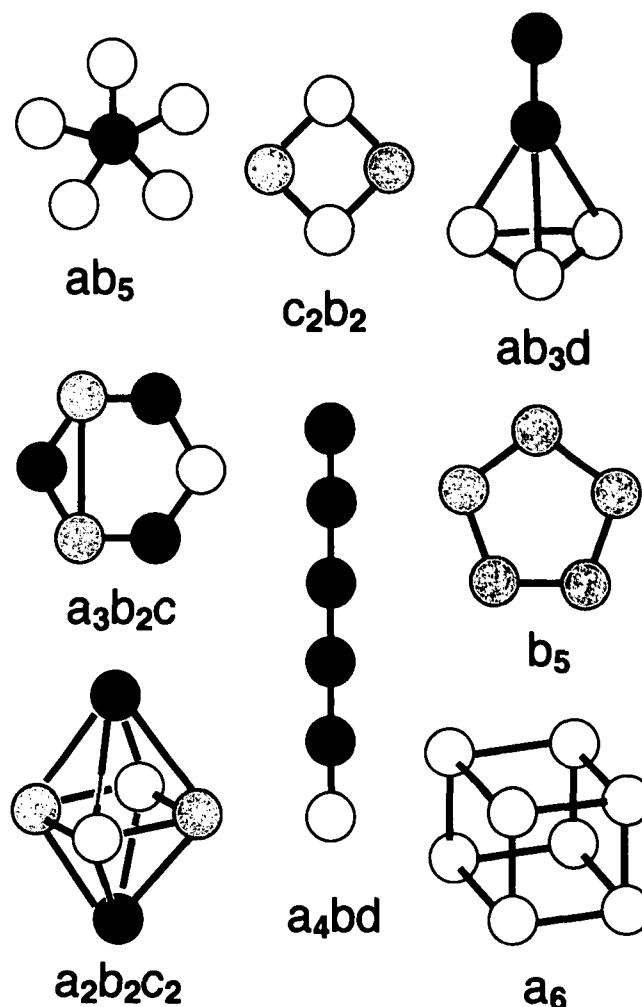


Fig. 5. Selection of oligomeric structures analysed by the OLIGO program.

potential use. Evidently, the program could be expanded and made far more sophisticated by adding graphical displays of output, automating the equilibration over ranges of starting concentrations, and so on. The deconstruction and equilibration algorithms could also be ported to larger machines to facilitate analysis of larger and more complicated oligomeric species. We encourage potential users to test the program and to relate to us their experiences, and we welcome suggestions for further development of the program.

## References

- Alberts, B., Bray, D., Lewis, J., Raff, M., Roberts, K. and Watson, J.D. (1994) *Molecular Biology of the Cell*, 3rd edn. Garland Publishing Inc., New York.
- Bray, D. and Bourret, R.B. (1995) Computer analysis of the binding reactions leading to a transmembrane receptor-linked multiprotein complex involved in bacterial chemotaxis. *Mol. Biol. Cell*, **6**, 1367–1380.
- Faux, M.C. and Scott, J.D. (1996) Molecular glue: Kinase anchoring and scaffolding proteins. *Cell*, **85**, 9–12.
- Gegner, J.A. and Dahlquist, F.W. (1991) Signal transduction in bacteria:

- CheW forms a reversible complex with the protein kinase CheA. *Proc. Natl Acad. Sci. USA*, **88**, 750–754.
- Gegner, J.A., Graham, D.R., Roth, A.F. and Dahlquist, F.W. (1992) Assembly of an MCP receptor, CheW, and kinase CheA complex in the bacterial chemotaxis signal transduction pathway. *Cell*, **70**, 975–982.
- Mochly-Rosen, D. (1995) Localization of protein kinases by anchoring proteins: a theme in signal transduction. *Science*, **268**, 247–251.
- Pawson, T. (1995) Protein modules and signalling networks. *Nature*, **373**, 573–580.
- Perrin, D.D. (1966) Multiple equilibria in assemblages of metal ions and complexing species: a model for biological systems. *Nature*, **206**, 170–171.
- Petzold, L.R. (1983) Automatic selection of methods for solving stiff and nonstiff systems of ordinary differential equations. *J. Sci. Stat. Comput.*, **4**, 136–148.
- Storer, A.C. and Cornish-Bowden, A. (1976) Calculation of the true concentrations of species present in mixtures of associating ions. *Biochem. J.*, **159**, 1–5.
- Tjian, R. and Maniatis, T. (1994) Transcriptional activation: A complex puzzle with few easy pieces. *Cell*, **77**, 5–8.

*Received on January 17, 1997; revised on March 6, 1997; accepted on March 13, 1997*