

# MULTIPLE SITE BINDING: An MLAB Example

Gary D. Knott, Ph.D.

Civilized Software Inc.  
12109 Heritage Park Circle  
Silver Spring, MD 20906 USA  
Tel: (301)962-3711  
Email: csi@civilized.com  
URL: www.civilized.com

## Introduction

The study of how a ligand material, such as a hormone, binds to one or more kinds of molecular complexes, called sites, is of fundamental importance in biochemistry. Sites are often embedded in cell membranes, and the binding serves to control the behavior of the cell itself. Typically we are interested in the number of distinct kinds of sites and their frequency of occurrence, and also the equilibrium constants for each ligand-site binding reaction which indicates the absolute strength of each such binding reaction.

For example, quantitative analysis of hormone-receptor binding can be easily performed using appropriate software such as MLAB. MLAB is a computer program whose name is an acronym for “modeling laboratory”; it is an interactive system for mathematical modeling, originally developed at the National Institutes of Health. MLAB can fit multiple non-linear models to data points obtained from standard displacement assays. Typical assays involve measuring the competition between radiolabelled and cold ligand in detergent-solubilized membrane preparations or on whole cells. Affinity constants and limit values of binding protein concentrations for single or multiple sites can be computed by fitting saturation curves in MLAB. Output can include Scatchard plots obtained by a suitable transformation.

The general form of experimental data consists of data points of the form  $(F_0, B)$  where  $F_0$  is the concentration of supplied ligand material and

$B$  is the concentration of ligand bound to sites, measured after equilibrium is reached.

MLAB can be used to analyze down and up regulation in receptor number on the cell surface, where the binding capacity or receptor number in liter/pmole is the limiting  $x$ -intercept on the Scatchard plot. The program is also useful in determining whether changes in binding by different ligand agonists/antagonists can be attributed to reduced binding affinity of the defined site/sites (seen as changes in the affinity constant), or to the total loss of binding ability of a class of sites (seen as changes in the binding capacity, and changes in the fit from an  $n+1$  to  $n$  site model). Similarly, comparison studies of a single ligand against different binding proteins that vary by site-directed mutagenesis, or alternative splicing can be useful in exploring mechanisms of ligand/receptor interaction, including amino acid and charge requirements.

In addition, post-translational processing and folding of the nascent protein in the endoplasmic reticulum can be studied by estimating the number of active receptor molecules that have reached the cell surface. These studies are of current interest to biochemists, and an accurate method of quantification is important since conclusions about the biological system, which in turn determine the future direction of research, are based on the calculated binding parameters.

In addition to saturation and Scatchard analyses, MLAB can be used in many other curve fitting applications such as kinetic, Michaelis-Menten, Lineweaver-Burke plots, and various statistical analyses. Kinetic analysis involves fitting differential equation models, which is a unique capability of MLAB.

## The MLAB Mathematical Modeling System

MLAB has hundreds of useful functions, e.g., the discrete Fourier transform function `dft` and the parametric spline interpolation function `splinep`. One of the central components of the system is a curve fitting program which will adjust the parameters of a model function to minimize the weighted sum of the errors raised to a specified power. A repertoire of mathematical operators and functions, routines for solving differential equations, a collection of routines for onscreen drawing and for hardcopy plotting, and mechanisms for saving data between sessions provide a powerful and convenient environ-

ment for data manipulation, arithmetic calculations, and for building and testing models.

The user communicates with MLAB by typing commands which are executed at once or by providing a script to be executed. Should the user have questions, typing `HELP` will put the on-line system documentation at his disposal. The MLAB language is defined in the MLAB reference manual.

One of MLAB's main uses is to fit models to data. Curve-fitting is a useful analytical tool in many diverse disciplines. The basic notion is easily described. Given data, say various points in the plane  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ , and a function  $y = f(x)$  where  $f$  involves some parameters, say  $a$  and  $b$ , as for example  $f(x) = ax^b + 1$ , we may wish to calculate values for the parameters  $a$  and  $b$  so that the function  $f$  well-predicts the observed data, that is, so that  $f(x_i) = y_i$  for  $1 \leq i \leq n$ . In this case, we say we have fit the model  $f$  to the data by estimating the parameters  $a$  and  $b$ . The end result is merely the values obtained for the initially unknown parameters. The same principles apply in higher dimensions with arbitrarily many parameters. MLAB can simultaneously fit multiple non-linear model functions, some or all of which may be implicit functions, or may even be defined by a system of differential equations.

The curve-fitting and graphics display facilities of MLAB make it an ideal tool for the estimation of equilibrium constants from data, which typically consist of observed amounts of ligand bound for various specified amounts of ligand provided for binding.

## The Mathematics of Multiple Site Binding

Suppose we have a ligand,  $F$ , which binds to each of  $N$  *independently-acting* sites,  $S_1, S_2, \dots, S_N$ , which are present in the various concentrations of  $S_{10} \mu\text{M}, S_{20} \mu\text{M}, \dots$ , and  $S_{N0} \mu\text{M}$  respectively. Each reaction  $F + S_i \rightleftharpoons B_i$  forms a bound complex  $B_i$ , and we shall define  $\kappa_i$  to be the associated equilibrium constant. Let  $F_0$  denote the concentration of ligand present initially, and let  $F$  be the concentration of free ligand at equilibrium. Similarly, let  $S_i$  denote the concentration of free sites of type  $i$  at equilibrium, and let  $B_i$  denote the concentration of bound site  $i$  complex at equilibrium. Note we use the symbols  $F, S_i$ , and  $B_i$  for the numerical quantities in  $\mu\text{M}$  units of these materials, as well as for the names of the materials themselves. Then:

$$\kappa_i = B_i / (FS_i) \quad \text{and} \quad B_i + S_i = S_{i0} \quad \text{for} \quad 1 \leq i \leq N, \quad \text{and}$$

$$F = F_0 - B_1 - B_2 - \dots - B_N.$$

Often there is a fictitious  $(N + 1)$ -st site,  $X$ , which binds  $F$  molecules, which is introduced to describe the *non-specific* binding of  $F$  molecules with weak affinity to many locations, other than the  $N$  specific sites of interest. Let  $\kappa_0$  be the equilibrium constant for the fictitious non-specific binding reaction  $F + X \rightleftharpoons Y$ . Then we have:

$$\begin{aligned} \kappa_i &= B_i/(FS_i) \quad \text{and} \quad B_i + S_i = S_{i0} \quad \text{for} \quad 1 \leq i \leq N, \\ \kappa_0 &= Y/(FX) \quad \text{and} \quad Y + X = X_0, \quad \text{and} \\ F &= F_0 - B_1 - B_2 - \dots - B_N - Y, \end{aligned}$$

where  $Y$  is the concentration of non-specifically bound ligand,  $X_0$  is the concentration of the fictitious non-specific binding site, and  $X$  is the concentration of the free fictitious site at equilibrium. Then  $B_i = \kappa_i S_{i0} F / (1 + \kappa_i F)$  for  $1 \leq i \leq N$ , and  $Y = \kappa_0 X_0 F / (1 + \kappa_0 F)$ .

Now, to capture the notion of non-specific binding as a weak sticking of  $F$  molecules almost everywhere, let  $\kappa_0$  tend to zero and let  $X_0$  tend to infinity such that  $\kappa_0 X_0 = c$ , where  $c$  is a fixed constant. Then  $Y \rightarrow cF$ , and we have:

$$B_i = \kappa_i S_{i0} F / (1 + \kappa_i F) \quad \text{for} \quad 1 \leq i \leq N, \quad Y = cF, \quad \text{and} \quad F = F_0 - B_1 - \dots - B_N - Y.$$

Note that the mathematical form  $B_i = S_{i0} F / (1/\kappa_i + F)$  is difficult to handle when  $\kappa_i$  approaches 0, or when  $1/\kappa_i + F$  approaches 0, so that the form given above is more convenient.

Usually values of  $F$  and/or  $B_1 + B_2 + \dots + B_N + Y$  are measured for different values of  $F_0$  and we wish to use this data, which is generally of the form  $(F_0, F)$  or  $(F_0, B_1 + \dots + B_N + Y)$ , to estimate  $\kappa_1, \kappa_2, \dots, \kappa_N, c$ , and, if not already known,  $S_{10}, S_{20}, \dots, S_{N0}$ . When only the total bound concentration  $B_1 + B_2 + \dots + B_N + Y$  is measured, rather than  $B_1, B_2, \dots, B_N$  and  $Y$  separately, only a few sites can be distinguished by curve-fitting. In order to determine how many kinds of sites appear to be present, we must try each of the 1-site, 2-site, etc. models and choose among them on the basis of how well the data is fit. If the equilibrium constants obtained for two species of sites are close, then there are no grounds for considering them to be distinct species, based on this analysis alone.

Often people use the model  $B(F) = \kappa_1 S_{10} F / (1 + \kappa_1 F) + \dots + \kappa_N S_{N0} F / (1 + \kappa_N F) + cF$  and fit it to data points of the form  $(F, B)$  where  $F$  is the free ligand concentration at equilibrium and  $B$  is the total bound ligand concentration at equilibrium, with one such point for each experiment. *The difficulty with this approach is that we must compute  $F$  as  $F_0 - B$  so that there is correlated error in **both** the dependent and independent values used to form the data points.* Our approach uses data points of the form  $(F_0, B)$ , and since  $F_0$  can be accurately determined, we avoid the aforementioned difficulty of error in the independent variable.

For example, the general two-site model with non-specific binding can be defined in MLAB with the following dialog. Here and hereafter, the text shown following the MLAB prompt asterisk is an MLAB command statement entered by the user.

```
* FUNCTION A(F0) = B(F(F0))
* FUNCTION B(F) = k1*S10*F/(1+k1*F) + k2*S20*F/(1+k2*F)
* FUNCTION F(F0) = ROOT(Z,0,F0,F0-B(Z)-Z*(1+c))
* FUNCTION Y(F0) = c*F(F0)
```

Here  $A(F_0) = B_1 + \dots + B_N$  for  $N = 2$ .

These commands exemplify the MLAB FUNCTION statement, which is used to define a function or differential equation. Note that arguments of functions must be explicitly specified. Variables, such as  $k_1$  and  $k_2$ , which appear in the body of a function, but not in its argument list, are called parameters. Parameters must be assigned values before an associated function can be evaluated.

ROOT is an operator which is built-in in MLAB.  $\text{ROOT}(Z, A, B, E)$  is a value between  $A$  and  $B$  which, when taken as the value of the dummy variable,  $Z$ , makes the expression,  $E$ , which involves  $Z$ , equal to zero. Thus  $\text{ROOT}(Z, A, B, E)$  is a solution of  $E(Z) = 0$ . The model given above, involving a so-called *implicit* function, deserves careful study; it is easily extendible to more than two sites. The amount  $F$  of free-ligand at equilibrium as a function of  $F_0$  satisfies  $F_0 - F = B(F) + c \cdot F$ .

## An Example

Suppose we have measured  $F$  in  $\mu\text{M}$  units as a function of  $F_0$  in  $\mu\text{M}$  units, as follows:

$F_0$	$F$
.58668	.036
1.1734	.096
2.3467	.385
2.9334	.61
4.1068	1.15
4.6934	1.46
5.8668	2.11
7.0402	2.73

and we have  $S_{10} = S_{20} = 1.7121 \mu\text{M}$ .

Then, we can introduce the appropriate constraints (which should always be used for this particular model), guess  $K_1$ ,  $K_2$ , and  $c$  and then estimate them, as follows.

```
* constraints q = {k1>=0,k2>=0,c>=0,S10>=0,S20>=0}
```

The **CONSTRAINTS** statement permits the user to specify successive linear inequalities or equations involving the parameters (or potential parameters). Now we may specify values for the parameters, guessing when necessary.

```
* k1 = 10; k2 = 1; c = 0; S10 = 1.7121; S20 = S10;
```

Here the **ASSIGNMENT** statement is exemplified. In this case all the above assignments are assigning values to scalar variables, but the **ASSIGNMENT** statement is used to assign values to matrices as well. This can be seen in the next **ASSIGNMENT** statement which defines a matrix,  $D$ .

```
* D = Kread(8,2)
.58668 .036
1.1734 .096
2.3467 .385
2.9334 .61
```

```

4.1068 1.15
4.6934 1.46
5.8668 2.11
7.0402 2.73

```

The KREAD operator takes optional array size arguments; in this case an 8 row by 2 column matrix is specified, and reads in numbers from the keyboard to construct an appropriately-dimensioned matrix as the result. An analogous function is used to read numbers from a file. This matrix is then, in this case, assigned to D. Note the entry of the numbers which follow. Now we may examine D by “typing it out” using the TYPE statement.

```

* type D
D: 8 by 2 matrix
1: .58668 .36E1
2: 1.1734 .96E-1
3: 2.3467 .385
4: 2.9334 .61
5: 4.1068 1.15
6: 4.6934 1.46
7: 5.8668 2.11
8: 7.0402 2.73

```

We have established a model function,  $F$ , and entered data,  $D$ . We expect that  $f(D[i,1]) \approx D[i,2]$  would hold, if only the parameters  $\kappa_1$ ,  $\kappa_2$ , and  $c$  were set to their “correct” values. The following FIT statement requests MLAB to estimate  $\kappa_1$ ,  $\kappa_2$ , and  $c$  by assigning them values which minimize the sum-of-squares objective function  $S(\kappa_1, \kappa_2, c) = \sum_{i=1}^8 (F(D[i,1]) - D[i,2])^2$ .

```

* maxiter = 30; TOLSOS = .001
* fit(k1,k2,c), F to d, constraints q
final parameter values
value          error          dependency    parameter
13.86352736    2.331373145    0.665073064    K1
0.5321372226    0.06602915638    0.9432600296    K2
0.5874015019    0.02229743988    0.9171690302    C
5 iterations

```

```

CONVERGED
best sum of squares = 9.78763e-04
root mean square error = 1.39912e-02
deviation fraction = 6.82654e-03
R squared = 9.99854e-01
no active constraints

```

The behavior of the FIT statement depends upon the supplied constraints  $q$ , as well as upon the MLAB control variables: `maxiter`, the maximum number of iterations and `tolsos`, the requested convergence factor.

MLAB uses a carefully-tuned version of the Marquardt-Levenberg magnified-diagonal algorithm which is, in turn, a form of the Gauss-Newton procedure for minimizing a function which is in the form of a sum-of-squares. This process estimates the value of the parameter vector  $b = (\kappa_1, \kappa_2, c)'$  by successive approximations  $b^{(0)}, b^{(1)}, \dots, b^{(n)}$ , where  $b^{(0)}$  is the vector of initial guesses for  $\kappa_1, \kappa_2$ , and  $c$ , and  $b^{(j+1)} = b^{(j)} + \beta^{(j)}$ , where

$$\begin{aligned}
\beta^{(j)} &= (X'V^{-1}X + \varepsilon G)^{-1}X'V^{-1}(y - (f(x_1; b^{(j)}), \dots, f(x_8; b^{(j)})))', \quad \text{with} \\
X_{st} &= \partial f(x_s; b^{(j)})/\partial b_t \quad \text{and} \\
G_{st} &= \text{if } s = t \text{ then } (X'V^{-1}X)_{st} \text{ else } 0 \quad \text{and} \\
x_s &= D[s, 1] \quad \text{for } 1 \leq s \leq 8, \quad \text{and} \\
y &= (D[1, 2], \dots, D[8, 2])',
\end{aligned}$$

where  $V$  is the estimated covariance matrix of the observations. In our example,  $V = I$ , the identity matrix. In general  $V$  is determined from weight-values supplied by the user.

An iteration consists of computing  $b^{(j+1)}$  from  $b^{(j)}$ . Note that this requires the partial derivatives of the model function with respect to the parameters evaluated at  $b^{(j)}$ , since these values form the matrix  $X$ . In MLAB, these derivatives are automatically computed symbolically and evaluated to form  $X$ . The convenience thus obtained is considerable and the parameter estimation process is provided with more accurate derivative values. For example the derivative of  $F$  with respect to  $\kappa_1$  can be explicitly displayed in MLAB as follows.

```

* type F diff k1
FUNCTION F DIFF K1(F0) = EVAL(Z,ROOT(Z,0,F0,(F0-B(Z))-Z*(1+c)),
    -B DIFF K1(Z)/(B DIFF F(Z)-(1+c)))

```

Indeed derivatives are full-fledged members of the class of functions and can be used in graphics or curve-fitting in MLAB just as can any other user-defined function.

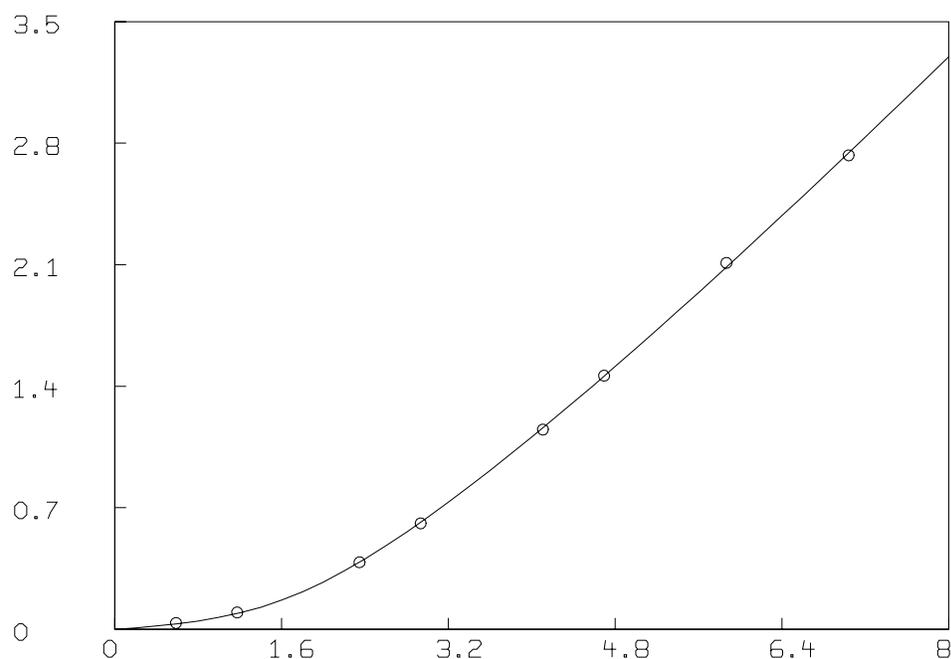
The material typed out above shows that the vector of parameters  $(K_1, K_2, c)$  has been estimated to be  $(13.8633 \pm 2.3315, .532151 \pm .0660296, .587396 \pm .0222973)$ , with reasonably small dependency values, and with an RMS error of about .014, which should be comparable with the experimental error in our data,  $D$ . The sum-of-squares was reduced from an initial value of 2.64314 to .000978763 at the final parameter values.

In order to visually see how our model with its parameters set to their best-estimated values corresponds to the data, we may draw a graph of the data points and the model function. Although we are drawing only the simplest and most direct kind of picture here, it should be noted that MLAB provides facilities for many types of point-symbols and types of lines, axes with arbitrarily-placed numeric labels in various formats, titles in the form of text strings in arbitrary sizes and various fonts with subscripts and superscripts, color, and a number of other special features. It is quite possible to prepare more or less elaborate publication-quality graphs with a modest amount of effort. Indeed this is one of MLAB's most-used facilities. The desired graph can be constructed as follows.

```

* draw D, linetype none,pointtype circle
* draw c2 = points(F,0:8:.2)
* VIEW

```



The first `DRAW` command above plots the data points, while the second `DRAW` command constructs a curve called `C2`, which is a graph consisting of solid straight-lines connecting the points which are the rows of the 2-column matrix which is the value of the expression `points(F,0:8:.2)`. This matrix has the values 0 through 8 in steps of .2 in its first column and corresponding values of the function  $F$  evaluated at 0 through 8 in steps of .2 in the second column. The `POINTS` operator is very useful for graphing functions. Both these curves are drawn in the default MLAB 2D-window called `W` (since no other window is specified) which has predefined labeled axes already present. The picture finally appears when its display is requested with a view statement and a plot can be obtained if desired using the `PLOT` statement.

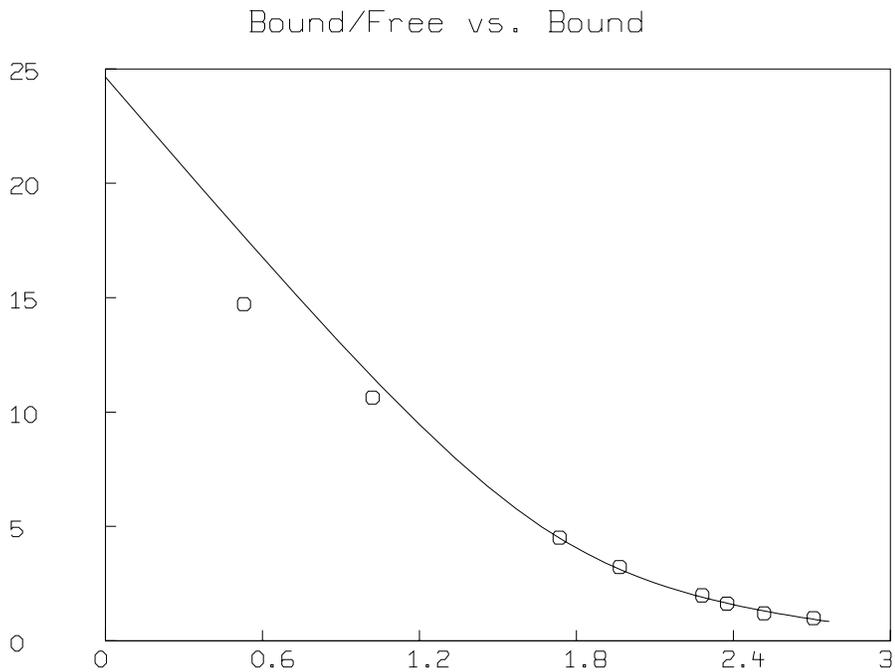
Often a Scatchard graph of Bound/Free vs. Bound (i.e.,  $A(F_0)/F(F_0)$  vs.  $A(F_0)$ ) is desired. Although such a formulation should not be used for curve-fitting due to the non-normal error introduced by computing Bound/Free, it is quite straightforward to draw the data and model Scatchard plots as follows. Note the current picture in `W` is saved and restored.

\* `SAVE W IN GW`

```

* DELETE W
* FUNCTION R(X,Y)=IF Y=0 THEN (k1*s10+k2*s20) ELSE X/Y
* MF = F ON 0:8:.2
* M = B ON MF
* M = M&'(R ON M&'MF)
* DRAW M;
* MF = D COL 2
* M = (D COL 1)-(1+C)*MF
* M = M&'(R ON M&'MF)
* DRAW M, LINETYPE NONE, POINTTYPE "o"
* TOP TITLE "Bound/Free vs. Bound"
* VIEW

```



```

* DELETE W, MF, M
* USE GW

```

The & operator denotes column concatenation, while the ON operator, as in  $F \text{ ON } H$ , applies the function  $F$  to each row of the matrix  $H$  treated as an argument list for  $F$  and returns the column vector of result values.

It is an enlightening exercise to fit the Scatchard model to the corresponding transformed data and to then draw a graph of the original data points and the function  $f$  using the parameter values obtained.

You can see that MLAB is an extremely flexible and general tool for curve-fitting. Moreover, it has a broad range of other useful functions, only a few of which have been alluded to here.

Further information about MLAB is available from Civilized Software, Inc., 12109 Heritage Park Circle, Silver Spring, MD 20906, Tel. (301) 962-3711, Email: [csi@civilized.com](mailto:csi@civilized.com).