Analysis of algebraic weighted least-squares estimators for enzyme parameters

Michael E. JONES

The School of Medicine, 6E-413, The Flinders University of South Australia, GPO Box 2100, Adelaide, South Australia 5001, Australia

An algorithm for the least-squares estimation of enzyme parameters K_m and V_{max} is proposed and its performance analysed. The problem is non-linear, but the algorithm is algebraic and does not require initial parameter estimates. On a spreadsheet program such as MINITAB, it may be coded in as few as ten instructions. The algorithm derives an intermediate estimate of K_m and V_{max} , appropriate to data with a constant coefficient of variation and then applies a single reweighting. Its performance using simulated data with a variety of error structures is compared with that of the classical reciprocal transforms and to both appropriately and inappropriately weighted direct least-squares estimators. Three approaches to estimating the standard errors of the parameter estimates are discussed, and one suitable for spreadsheet implementation is illustrated.

INTRODUCTION

The rectangular hyperbola is familiar to biochemists as a mathematical description of data arising from simple enzymic, binding and transport mechanisms. In the context of initial-velocity data derived from reactions following Michaelis–Menten kinetics, the dependence of initial velocity, v , on substrate concentration ($[S]$) is well approximated by:

$$v = V_{max}[S]/(K_m + [S])$$

where K_m and V_{max} are model parameters to be estimated.

Algorithms for estimating the parameters include the application of simple linear regression following the reciprocal transforms of Lineweaver & Burk ($1/v$ against $1/[S]$), Eadie & Hofstee (v against $v/[S]$) and Hanes ($[S]/v$ against $[S]$) ([1] and references cited therein). For binding data, the transform of Scatchard (bound/free against bound) is mathematically equivalent to the Eadie–Hofstee with the axes exchanged. These are convenient for the visual inspection of data, but are less efficient than an appropriately weighted direct least-squares fit for estimating parameters [2–5].

The reciprocal-transform methods have the advantage of being algebraic. That is to say, they have a predefined number of calculations, do not depend on iterative approximation and can be implemented by hand or on commercially available spreadsheet programs. By comparison, a direct weighted least-squares fit of the untransformed data employs mathematical techniques such as the Marquardt–Levenberg algorithm [6], which lie outside the interest of most practising experimentalists. In addition, non-linear minimization algorithms usually require initial parameter estimates, may not converge or may converge on an inappropriate local minimum. Their programming usually requires ‘conditional jumps’ and comparisons to detect lack of convergence and to decide when to stop. Such programs are necessarily lengthy. That listed by Roberts [1] extends to some 300 lines of FORTRAN code, although for some weightings more succinct algorithms have been described [7]. The non-parametric approach of Cornish-Bowden & Eisenthal [5], although robust in the presence of outliers, has not gained wide acceptance and cannot easily be adapted to spreadsheet calculations.

There is therefore a place for an algorithm which offers the desirable characteristics of these approaches. It should not require

initial parameter estimates, nor incorporate conditional jumps and comparisons. It should be programmable on a spreadsheet, yet offer the statistical efficiency of the more cumbersome direct least-squares algorithms. The algorithm below meets these criteria.

AN ALGORITHM TO ESTIMATE K_m and V_{max}

The data comprise n pairs (s_i, v_i) , $i = 1, 2, \dots, n$ substrate concentration s_i and observed initial velocity v_i . For each data pair let w_i be a weighting which is initially set at $w_i = 1.0$ for all i . Then a reweighted least-squares algorithm appropriate to Poisson-like error (RLS_p) is as follows.

1. Define the quantities $A = \sum w_i(v_i/s_i)^2$, $B = \sum w_i v_i^2/s_i$, $C = \sum w_i v_i^2$, $D = \sum w_i v_i/s_i$ and $E = \sum w_i v_i$.
2. Define intermediate estimates:

$$\hat{K}_m = (DC - BE)/(AE - BD)$$

$$\hat{V}_{max} = (AC - B^2)/(AE - BD)$$

3. Define the weights:

$$w_i = \hat{V}_{max} s_i / (\hat{K}_m + s_i)$$

4. Repeat steps 1 and 2 to give final estimates of K_m and V_{max} .

JUSTIFICATION OF THE ALGORITHM

For weighted direct least-squares algorithms, the estimates $\hat{K}_m$ and $\hat{V}_{max}$ of K_m and V_{max} , respectively are the values which minimize the weighted sum of squares of differences between the observed velocities, v_i , and the expected velocities, e_i defined as:

$$e_i = \hat{V}_{max} s_i / (\hat{K}_m + s_i)$$

The appropriate weights, w_i , are determined by the error structure of the data, and the objective function, Ω , to be minimized by the choice of $\hat{K}_m$ and $\hat{V}_{max}$ is defined as:

$$\Omega = \sum_i w_i (e_i - v_i)^2$$

The proposed algorithm derives from the observation [8,9] that, if the velocity estimates are subject to a constant coefficient of variation, so that $w_i = 1/e_i^2$, then an appropriately weighted

direct least-squares estimate would minimize the objective function defined by:

$$\Omega = \sum \left[\frac{e_i - v_i}{e_i} \right]^2$$

$$= \sum \left[1.0 - \frac{K_m v_i}{V_{\max} s_i} - \frac{v_i}{V_{\max}} \right]^2 \quad (1)$$

In the latter form the problem is clearly amenable to algebraic solution, because it amounts to a multiple linear regression in which the 'dependent' variable, 1.0, is determined by the 'explanatory variables' v_i/s_i and v_i . The coefficients of those variables in the regression analysis will therefore be the appropriately weighted least-squares estimates of $K_m/V_{\max}$ and $1/V_{\max}$, respectively.

Experimental data is not always subject to a constant coefficient of variation, however. The error structure in real data usually lies somewhere on a continuum between a constant absolute error (homoscedastic) at one extreme and a constant coefficient of variation at the other [8,10]. Between these two is an error for which the standard deviation is proportional to the square root of the expected value. For such an error structure, the appropriate weighting is $w_i = 1/e_i$, giving an objective function

$$\Omega = \sum (e_i - v_i)^2 / e_i$$

to be minimized. The proposed algorithm effects a minimization of this through a reweighted least-squares technique. Such an approach is conventionally used when the appropriate weighting in an otherwise linear regression problem depends on the expected value. In the current context, however, we use the technique in an initially non-linear regression where there is a weighting which makes the mathematics algebraically tractable, but which might not be experimentally justifiable. We recast the expression for Ω in the form:

$$\Omega = \sum (e_i - v_i)^2 / e_i \quad (2)$$

$$\approx \sum w_i \left[\frac{e_i - v_i}{e_i} \right]^2 \quad (w_i \approx e_i) \quad (3)$$

$$= \sum \left[\sqrt{(w_i)} - \frac{K_m}{V_{\max}} \frac{\sqrt{(w_i)} v_i}{s_i} - \frac{\sqrt{(w_i)} v_i}{V_{\max}} \right]^2 \quad (4)$$

A direct least-squares estimator minimizes Ω as defined by eqn. (2), by using a general non-linear minimization algorithm such as the Marquardt-Levenberg. The proposed algorithm minimizes Ω as defined by eqn. (4), in which the least-squares estimates of the coefficients $K_m/V_{\max}$ of $v_i \sqrt{(w_i)}/s_i$ and $1/V_{\max}$ of $v_i \sqrt{(w_i)}$ are derivable algebraically. In this context, the variable, w_i , is now really only a *part* of the weighting; we want a weighting of $1/e_i$, it is mathematically convenient to calculate with a weighting of $1/e_i^2$, and we utilize this to construct an overall weighting of w_i/e_i^2 which will approach $1/e_i$ when w_i is close to e_i . (If the variance structure is known to be otherwise, as in a case given by Askelöf *et al.* [10], where variance (Var) $\propto e_i^{1.6}$, then by putting $w_i = e_i^{0.4}$ the algorithm can be modified so that w_i/e_i^2 approaches $1/e_i^{1.6}$.)

It is conventional, when using a reweighting algorithm, to reweight iteratively, the values of the weights at one cycle depending on parameter estimates at the previous cycle, and continuing until further cycles make no change in the estimates. Such a strategy requires comparisons and conditional jumps, however, and these we seek to avoid. Iteratively reweighted least-squares algorithms often converge rapidly, however, and this suggests that an estimator based on a single reweighting should

perform well. Accordingly, four reweighted least-squares estimators have been constructed and compared with traditional estimators. All four make an initial estimate based on eqn. (4) above with $w_i = 1.0$. Two then reweight with weightings appropriate to a Poisson-like random error the standard deviation of which is proportional to the square root of the expected velocity. These are denoted RLS_p where the reweighting is carried out only once, and $IRLS_p$ where there is a constant number, ten, of iterative reweightings. The other two reweight with weightings appropriate to a homoscedastic error structure in which the standard deviation of the random error is constant and therefore independent of the expected velocity. These are denoted RLS_h for a single reweighting, and $IRLS_h$ for ten reweightings.

AN ALGORITHM FOR PARAMETER VARIANCES

As in all non-linear models, these variances must be regarded as an approximation only. An approach appropriate for estimators which assume a Poisson-like error, and which lends itself to spreadsheet calculation, is as follows.

5. Estimate the variance σ^2 in terms of the residuals errors:

$$\hat{\sigma}^2 = \frac{1}{n-2} \sum (e_i - v_i)^2 / e_i$$

where e_i is as previously described, and hence estimate the variance of v_i :

$$Var(v_i) = e_i \hat{\sigma}^2$$

6. Define the quantities $U = \sum v_i^2$, $W = \sum v_i^2 / s_i$, $X = \sum s_i / (\hat{K}_m + s_i)$, $Y = \sum s_i / (\hat{K}_m + s_i)^2$ and $Z = \sum s_i / (\hat{K}_m + s_i)^3$.

7. Calculate the partial derivatives:

$$\frac{\partial \hat{K}_m}{\partial v_i} = v_i [X/s_i - Y(1 + \hat{K}_m/s_i)] / [YW - Z(\hat{K}_m W + U)] \quad (5)$$

$$\frac{\partial \hat{V}_{\max}}{\partial v_i} = \left[\frac{v_i}{s_i} + \frac{WZ \partial \hat{K}_m}{Y \partial v_i} \right] / \hat{V}_{\max} Y \quad (6)$$

8. Calculate the parameter variances:

$$Var(\hat{K}_m) = \sum_i \left(\frac{\partial \hat{K}_m}{\partial v_i} \right)^2 Var(v_i) \quad (7)$$

$$Var(\hat{V}_{\max}) = \sum_i \left(\frac{\partial \hat{V}_{\max}}{\partial v_i} \right)^2 Var(v_i) \quad (8)$$

The justification for this algorithm appears in the Appendix.

COMPUTATIONAL METHODS

Simulated data were constructed with a variety of error structures, and these data were analysed using the various estimators. The simulations were carried out on a Microbits computer using an Intel 80486 chip running under Microsoft DOS 5.0 programmed in Borland Turbo Pascal 6.0. Random error was generated by the PASCAL *Random* procedure and randomly reshuffled using the *Ran0* procedure distributed by Press *et al.* [13]. Data generated and analysed by the PASCAL code was reanalysed using the Minitab 7.2 code appearing in the Appendix.

COMPARISON OF ESTIMATORS

Three analyses are reported, using simulated data with different error structures. In each, error-free data were generated for an

Table 1. Estimator performance using data with homoscedastic error

The means and s.d. of estimates of K_m and V_{max} derived for 1000 simulations in which a normally distributed homoscedastic error of s.d. 5 was added to error-free data corresponding to a K_m of 1.0 and a V_{max} of 100. Abbreviations for estimators are: DLS_h , direct unweighted least squares; RLS_h , reweighted least squares with a single reweighting appropriate to homoscedastic data; $IRLS_h$, iteratively reweighted least squares, with a constant ten iterations, the reweighting being appropriate to homoscedastic error; DLS_p , direct least squares appropriate to Poisson-like error; RLS_p , reweighted least squares with single reweighting appropriate to Poisson-like error; $IRLS_p$, iteratively reweighted least squares (ten iterations) appropriate to Poisson-like error; DLS_{ccv} , direct least squares appropriate to a constant coefficient of variation; Lineweaver, Lineweaver–Burk plot of $1/v$ against $1/s$; Eadie, Eadie–Hofstee of v against v/s ; Hanes, Hanes plot of s/v against s .

Estimator	K_m		V_{max}	
	Mean	S.D.	Mean	S.D.
DLS_h	1.00	0.14	100.0	5.06
RLS_h	0.95	0.13	99.0	4.89
$IRLS_h$	0.96	0.13	99.3	4.91
DLS_p	0.97	0.15	99.4	5.47
RLS_p	0.94	0.15	98.7	5.31
$IRLS_p$	0.94	0.15	98.8	5.32
DLS_{ccv}	0.94	0.20	98.7	7.42
Lineweaver	0.34	44.60	69.1	2015
Eadie	0.78	0.16	88.7	6.87
Hanes	1.10	0.24	101.7	6.75

Table 2. Estimator performance using data with Poisson-like error

The means and s.d. of estimates of K_m and V_{max} in 1000 simulations as for Table 1, but with normally distributed random error with s.d. equal to the square root of the expected velocity. Abbreviations for estimators are as for Table 1.

Estimator	K_m		V_{max}	
	Mean	S.D.	Mean	S.D.
DLS_h	1.01	0.21	100.4	8.50
RLS_h	0.96	0.19	100.1	8.16
$IRLS_h$	0.97	0.19	100.4	8.22
DLS_p	0.99	0.18	100.2	7.85
RLS_p	0.96	0.17	100.0	7.69
$IRLS_p$	0.96	0.17	100.1	7.71
DLS_{ccv}	0.97	0.19	100.4	8.56
Lineweaver	1.12	4.49	103.9	200.49
Eadie	0.79	0.15	89.3	7.40
Hanes	1.06	0.22	100.3	9.13

experiment with $K_m = 1.0$ and $V_{max} = 100$. Substrate concentrations of 0.2, 0.5, 1.0, 2.0 and 5.0 were used, there being three replicates at each substrate concentration. The error structures are referred to as homoscedastic (*h*), Poisson-like (*P*), and constant coefficient of variation (*ccv*). For each error structure, 1000 simulations were run in each of which the three classical reciprocal transforms, three direct least-squares estimators and four proposed reweighted least-squares estimators were applied to the data. The performance of the estimators, as characterized by the bias and standard deviation of estimates of K_m and V_{max} , was assessed. For homoscedastic error, the error was normally distributed with standard deviation of 5. This corresponds to 30% of the expected velocity at the lowest substrate con-

Table 3. Estimator performance with a constant coefficient of variation

The means and s.d. of estimates of K_m and V_{max} in 1000 simulations as for Table 1, but with normally distributed random error with s.d. equal to 20% of the expected velocity. Abbreviations for estimators are as for Table 1.

Estimator	K_m		V_{max}	
	Mean	S.D.	Mean	S.D.
DLS_h	1.05	0.36	101.5	15.5
RLS_h	1.00	0.31	103.2	14.6
$IRLS_h$	1.02	0.32	103.8	14.8
DLS_p	1.01	0.25	102.2	13.1
RLS_p	1.00	0.25	103.4	13.0
$IRLS_p$	1.01	0.25	103.6	13.0
DLS_{ccv}	1.02	0.22	104.1	12.5
Lineweaver	1.08	0.48	100.3	25.3
Eadie	0.75	0.17	87.2	10.5
Hanes	1.04	0.35	98.0	16.3

centration and 6% at the highest. The performance of the estimators on data having homoscedastic error is summarized in Table 1.

For a Poisson-like structure, the error was normally distributed with a standard deviation equal to the square root of the expected velocity. This was 25% of the lowest expected velocity and 11% of the highest. With this error, the estimators performed as in Table 2.

For a constant coefficient of variation, the random error was normally distributed with a standard deviation 20% of the expected velocity, and the estimator performance is summarized in Table 3.

Several trends are clear in these simulations. For all error structures, the reciprocal transforms performed less satisfactorily, often markedly so, than the least-squares estimators. In particular, the previously reported inferiority of the Lineweaver–Burk method [11] is confirmed. This is particularly so for homoscedastic error, where the method usually overestimates the parameters, but the occasional estimate is large and negative (i.e. $\hat{K}_m = -1388$ and $\hat{V}_{max} = -62723$). Such outlying estimates dominate the mean and variance in the simulations for this plot. For the reweighted least-squares estimators, the difference between a single reweighting and ten iterations was negligible. All least-squares estimators performed acceptably on all error structures. As expected on theoretical grounds, the appropriately weighted direct least-squares estimators performed marginally more satisfactorily, but the significant finding was that this margin was small. The margin was most obvious when the direct least-squares estimator appropriate to a constant coefficient of variation was applied to data with homoscedastic error. The problem then was essentially not one of bias, but of an increased standard deviation; a 6% bias in estimating K_m is unlikely to be of practical significance, but a standard deviation of 0.201 in the K_m estimate (variance 0.0406) as opposed to 0.141 (variance 0.0198) for the appropriately weighted direct least-square estimator represents a relative efficiency of 49%. This efficiency increased to 93% with a single reweighting appropriate to Poisson-like error. In advocating weighting appropriate to Poisson-like error, the point is not that this is always the error structure to be found (although there is support for such a structure [8]), but that such an estimator appears to function adequately across a spectrum of likely error structures.

Because the model is non-linear, estimates of parameter variances based on linearizing assumptions will be only approximate, even when the assumption of Poisson errors is correct.

By analogy with linear models we might expect the actual parameter value to lie within 2.160 estimated standard errors of the estimated parameter value 95 % of the time, and within 3.012 estimated standard errors 99 % of the time (these values are derived from the Student t distribution with 13 degrees of freedom; we have 15 readings in the simulated experiments, and two estimated parameters). With Poisson distributed error, these confidence intervals were found to be 94.5 % and 98.8 % for $\hat{V}_{\max}$, and 92 % and 97.5 % for $\hat{K}_m$. With inappropriate error structures, the confidence intervals calculated assuming a Poisson error are less reliable, but still remarkably good. With homoscedastic error, the confidence intervals were found to be 96.4 % and 99.2 % for $\hat{V}_{\max}$, and 88.5 % and 96 % for $\hat{K}_m$. With constant coefficient of variation they were 90 % and 98 % for $\hat{V}_{\max}$, and 94.5 % and 98 % for $\hat{K}_m$.

DISCUSSION

In the analysis of initial-velocity data from enzyme experiments there is a trade-off between convenience and reliability. Reciprocal transforms methods are convenient, but unreliable. Direct least-squares fits are inconvenient, but are reliable when common pitfalls are avoided. The analysis confirms the theoretical expectation that the most reliable estimates are derived from an appropriately weighted least-squares algorithm. To advocate the routine use of such an algorithm is, however, a counsel of perfection; the error structure, and hence the appropriate weighting, is usually unknown.

Least-squares estimators matched to Poisson-like error, in which the standard deviation is proportional to the square root of the expected velocity, perform very well across a spectrum of likely error structures. The reweighted algorithms have the same standard deviation as the direct least-squares algorithms, and iterative reweighting has nothing useful to offer over a single reweighting.

The proposed algorithm makes an initial estimate appropriate to a constant coefficient of variation, and it then applies a single reweighting appropriate to a Poisson-like error structure. It offers a simple algebraic solution to the efficient estimation of enzyme parameters, and its statistical properties will be close to those of the least-squares estimator derived by the more classical Marquardt-Levenberg approach. By whatever algorithm these estimators are derived, there are three further series of calculations which the experimentalist may wish to carry out. They are the examination of residual errors, the estimation of standard errors for $\hat{K}_m$ and $\hat{V}_{\max}$, and an examination of other values of K_m and $V_{\max}$, in order to ensure that the algorithm has indeed delivered estimates acceptably close to optimal.

The calculation of residual errors using a spreadsheet is elementary, and its implementation is offered in the Appendix. The estimation of standard errors is less straightforward, because they will usually be used as a rough guide to confidence intervals for the parameter estimates. One approach is to use Monte Carlo simulation [13], and the algorithm outlined is ideal for this because it is rapid and requires no initial estimates. For such an application the algorithm would be implemented in a general programming language such as PASCAL or C; spreadsheets tend to be cumbersome and too slow for large numbers of

iterations. The alternative approach advocated in a seminal paper by Wilkinson [14], is to assume that the variance $Var(K_m)$ of K_m and the variance $Var(v_i)$ of the i^{th} initial-velocity estimate are related by:

$$Var(K_m) = \sum_i \left(\frac{\partial \hat{K}_m}{\partial v_i} \right)^2 Var(v_i)$$

and analogously for $Var(V_{\max})$. This assumes both that the partial derivatives $\partial \hat{K}_m / \partial v_i$ can conveniently be calculated and that the variances $Var(v_i)$ are known. One approach to the partial derivatives is to calculate them numerically, but this is cumbersome on a spreadsheet program. A good approximation, however, is to calculate partial derivatives that would obtain if the objective function of eqn. (2) has been exactly, rather than approximately, minimized. On that basis, the partial derivatives can be obtained analytically in a form convenient for spreadsheet calculation and the confidence intervals derived from this. This is the basis of the algorithm given and is illustrated in the Appendix. As Atkins & Nimmo [15] have emphasized, however, such estimates are only approximations.

Finally, the experimentalist would usually like some measure of the extent to which the delivered estimates are optimal for his particular experimental data. The most direct approach to this is to 'try' values of K_m and $V_{\max}$, other than those offered, and to compare the fit. This amounts to a re-examination of residual errors, and to assist this the code offered in the Appendix prints out Ω , the weighted sum of squares that it is intended to minimize. A more convenient approach, however, relies on the calculations already carried out to estimate parameter variances. The function

$$G(k) = XW - Y[kW + U]$$

will be zero when k is the required optimum $\hat{K}_m$ which minimizes Ω of eqn. (2) [7]. The Appendix illustrates the optional use of this to 'polish' the estimate $\hat{K}_m$ and to ensure that $\hat{K}_m$ and $\hat{V}_{\max}$ have indeed minimized Ω .

This work was supported by grants from the National Health and Medical Research Council of Australia.

REFERENCES

1. Roberts, D. V. (1977) *Enzyme Kinetics*, Cambridge Chemistry Texts, Cambridge University Press, Cambridge.
2. Bliss, C. I. & James, A. T. (1966) *Biometrics* **22**, 573–602
3. Duggleby, R. G. (1979) *J. Theor. Biol.* **81**, 671–684
4. Bürgisser, E. (1984) *Trends Pharmacol. Sci.* **5**, 142–144
5. Cornish-Bowden, A. & Eisenthal, R. (1974) *Biochem. J.* **139**, 721–730
6. Sadler, D. R. (1975) *Numerical Methods for Nonlinear Regression*, University of Queensland Press, St. Lucia
7. Jones, M. E. & Taransky, K. (1991) *Comput. Biol. Med.* **21**(6), 459–464
8. Storer, A. C., Darlison, M. G. & Cornish-Bowden, A. (1975) *Biochem. J.* **151**, 361–367
9. Jones, M. E. (1991) *Trends Anal. Chem.* **10**(7), 200–202
10. Askelöf, P., Korsfeldt, M. & Mannervik, B. (1976) *Eur. J. Biochem.* **69**, 61–67
11. Dowd, J. E. & Riggs, D. S. (1965) *J. Biol. Chem.* **240**(2), 863–869
12. Ryan, B. F., Joiner, B. L. & Ryan, T. A. (1985) *Minitab Handbook*, 2nd edn., Duxbury Press, Boston
13. Press, W. H., Flannery, B. P., Teukolsky, S. A. & Vetterling, W. T. (1986) *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge
14. Wilkinson, G. N. (1961) *Biochem. J.* **80**, 324–332
15. Atkins, G. L. & Nimmo, I. A. (1980) *Anal. Biochem.* **104**, 1–9

APPENDIX

Spreadsheet calculation of parameters

We assume that column one (C1; Table 1A) contains substrate concentrations, column 2 contains corresponding velocities, and

column 3, which will be used to derive weightings, is initially set to unity. The example is written for a MINITAB [2] spreadsheet,

Table 1A. Values returned in calculation of NAD⁺ data

Column entries are shown as they will occur during the calculation of the data of Wilkinson [1]. The spreadsheet is initially set up with s_i and v_i in columns 1 and 2 respectively, and 1.0 in column 3. Column entries are as they appear after the execution of 'estimate.mtb' followed by 'variance.mtb'.

C1 s_i	C2 v_i	C3 e_i	C5 $Var(v_i)$	C12 $\partial \hat{K}_m / \partial v_i$	C13 $\partial \hat{V}_{max} / \partial v_i$
0.138	0.148	0.133358	0.0001390	-3.61311	-1.68960
0.220	0.171	0.190118	0.0001982	-2.13890	-0.90812
0.291	0.234	0.230381	0.0002402	-1.78307	-0.65604
0.560	0.324	0.336428	0.0003507	-0.11153	0.30065
0.766	0.390	0.388447	0.0004050	0.69126	0.78527
1.460	0.493	0.485438	0.0005061	2.22226	1.68424

for which the instructions are particularly transparent; the j^{th} column is denoted Cj and j^{th} scalar variable K j . An alternative coding which utilizes the capacity of Minitab to perform multiple linear regression, is given subsequently.

The 'core' of the algorithm resides in the thirteen lines of code:

```

Let C8 = C3*C2
Sum C8 and store in K8
Let C7 = C8/C1
Sum C7 and store in K7
Let C6 = C8*C2
Sum C6 and store in K6
Let C5 = C6/C1
Sum C5 and store in K5
Let C4 = C5/C1
Sum C4 and store in K4
Let K1 = (K7*K6 - K5*K8)/(K4*K8 - K5*K7)
Let K2 = (K4*K6 - K5*K5)/(K4*K8 - K5*K7)
Let C3 = K2*C1/(K1 + C1)

```

in which the first ten lines calculate the variables A-E of step 1, and place their values in K4-K8 respectively. The next two lines put an estimate of K_m into K1 and of V_{max} into K2, according to step 2 of the algorithm. The final line calculates e_i , which is used as w_i in step 3. With those 13 lines of code stored as the instruction sequence 'core.mtb', the algorithm is executed with the instructions, which we call 'estimate.mtb':

```

execute 'core' 2 times
print K1, K2

```

The first execution of 'core.mtb' calculates values appropriate to a constant coefficient of variation and then re-adjusts the weights. The second calculates $\hat{K}_m$ and $\hat{V}_{max}$ appropriate to these weightings. K1 is then printed out as the estimate of K_m and K2 as the estimate of V_{max} .

Regression approach to parameters

If, like Minitab, a spreadsheet offers multiple linear regression, then a more succinct coding of 'core.mtb' is:

```

Let C6 = sqrt(C3)
Let C5 = C2*C6
Let C4 = C5/C1
Regress C6 on the 2 variables C4 and C5;
nonconstant;
coefficients in C6.

```

```

Let K1 = C6(1)/C6(2)
Let K2 = 1.0/C6(2)
Let C3 = K2*C1/(K1 + C1)

```

In this coding of 'core.mtb' the first three lines create the variables $\sqrt{w_i}$, $\sqrt{(w_i)v_i/s_i}$ and $\sqrt{(w_i)v_i}$ of eqn. (3) of the main paper. The next command carries out the regression, the sub-commands ensuring that there is no constant term and directing the estimated values of K_m/V_{max} and $1.0/V_{max}$ to re-use column 6. The last three lines calculate $\hat{K}_m$ and $\hat{V}_{max}$ from this, and recalculate the weightings as in the previous coding of 'core'.

Examination of residual errors

Examination of residual errors highlights deviations from the model and gives some idea of the error structure. The assumed variance of v_i is of the form $Var(v_i) = \sigma^2 e_i$ where σ^2 is to be calculated and e_i is the expected velocity. Residuals are calculated by code which we designate 'residual.mtb' and which assumes that C1-C3 contain s_i , v_i and e_i , while K1 and K2 contain the estimates $\hat{K}_m$ and $\hat{V}_{max}$ produced by 'estimate.mtb'. The required code is:

```

Let C4 = (C2 - C3)
Let C5 = C4*C4/C3
Sum C5 and put in K5
Let K5 = K5/(Count(C1) - 2)
Let C5 = K5*C3
Print C1-C5

```

The first line calculates the residual error in C4. Column 5 then calculates weighted squared residual error which is summed into K5 (Ω of eqn. 2 of the main paper), and printed, and then divided by $(n-2)$ to give the estimate $\hat{\sigma}^2$, which is left in K5. Column 5 is then re-used in the penultimate line to give an estimate, $e_i \hat{\sigma}^2$, of the variance of v_i .

Calculation of parameter variances

This is carried out by the following code which we refer to as 'variance.mtb':

```

Execute 'residual'
Let C6 = K1 + C1
Let C7 = C2*C2
Sum C7 and put in K7
Let C8 = C7/C1
Sum C8 and put in K8
Let C9 = C1/C6
Sum C9 and put in K9
Let C10 = C9/C6
Sum C10 and put into K10
Let C11 = C10/C6
Sum C11 and put into K11
Let K6 = (K11*(K1*K8 + K7) - K10*K8)
Let C12 = -C2*((K9/C1 - K10*(1.0 + K1/C1))/K6)
Let C13 = (C2/C1 + K8*K11*C12/K10)/(K10*K2)
Let C14 = C12*C12*C5
Sum C14 and put in K14
Let C15 = C13*C13*C5
Sum C15 and put in K15

```

in which the first line ensures that estimates of $Var(v_i)$ are placed in C5, and the second calculates $(\hat{K}_m + s_i)$ in C6. The next ten lines put the quantities U , W , X , Y and Z of step 6 into constants K7-K11 respectively. K6 is used to store the denominator that appears in the partial derivatives of eqn. (5) of the main paper. The partial derivatives $\partial \hat{K}_m / \partial v_i$ and $\partial \hat{V}_{max} / \partial v_i$ of eqns. (5) and (6)

of the main paper are then placed in columns 12 and 13 respectively. The individual components of $Var(\hat{K}_m)$ are then placed in C14 and summed to give $Var(\hat{K}_m)$ in K14. The same follows with the individual components of $Var(\hat{V}_{max.})$ in C15 being summed into K15.

Mathematical derivation

Given the objective function Ω of eqn. (4) of the main paper, the normal equations $\partial\Omega/\partial P = 0$ and $\partial\Omega/\partial Q = 0$, where $P = K_m/V_{max.}$ and $Q = 1/V_{max.}$ may be recast:

$$P \sum \frac{w_i v_i^2}{s_i^2} + Q \sum \frac{w_i v_i^2}{s_i} = \sum \frac{w_i v_i}{s_i}$$

$$P \sum \frac{w_i v_i^2}{s_i} + Q \sum w_i v_i^2 = \sum w_i v_i$$

Expressing this in matrix form, using A–E as defined in step 1 of the algorithm we have:

$$\begin{pmatrix} A & B \\ B & C \end{pmatrix} \begin{pmatrix} P \\ Q \end{pmatrix} = \begin{pmatrix} D \\ E \end{pmatrix}$$

Then step 2 amounts to the use of the determinant rule to solve for P and Q , from which $\hat{V}_{max.} = 1/Q$ and $\hat{K}_m = P/Q$. These calculations are, of course, equivalent to those underlying multiple linear regression.

The equations underlying the calculation of parameter variances are derived from the objective function Ω as in eqn. (2) of the main paper. The normal equations $\partial\Omega/\partial\hat{K}_m = 0$ and $\partial\Omega/\partial\hat{V}_{max.} = 0$ can be recast [3] in the form:

$$G(\hat{K}_m) = XW - Y(\hat{K}_m W + U) = 0 \quad (A1)$$

where U , W , X , Y and Z are as previously defined, and when $G(\hat{K}_m) = 0$ then:

$$\hat{V}_{max.} = \sqrt{(W/Y)} \quad (A2)$$

Then eqn. (A1) implies that:

$$\partial\hat{K}_m/\partial v_i = -\frac{\partial G}{\partial v_i} / \frac{\partial G}{\partial \hat{K}_m}$$

from which eqn. (5) of the main paper is immediate, and eqn. (A2) implies:

$$\frac{\partial\hat{V}_{max.}}{\partial v_i} = \frac{1}{2\hat{V}_{max.}} \frac{\partial}{\partial v_i} \left(\frac{W}{Y} \right)$$

from which eqn. (6) of the main paper follows.

These equations lead to a method of improving the parameter estimates using Newton's method: $\hat{K}_m$ will already be very close to the root of G , and 'variance.mtb' has already calculated $\partial G/\partial\hat{K}_m$, which is twice the value stored in K6. The final approach of $\hat{K}_m$ to the optimal value can be followed by examining $G(\hat{K}_m)$, which will approach zero as the estimate $\hat{K}_m$ is successively 'polished' by the following code 'polish.mtb':

```
Let K16 = K8*K9-K10*(K1*K8+K7)
Print K16
Let K1 = K1-0.5*K16/K6
Execute 'variance'
Let K2 = Sqrt(K8/K10)
Print K1, K2
Let C3 = K2*C1/(K1+C1)
```

in which the first two lines calculate and print out the current value of $G(\hat{K}_m)$, the third applies Newton's method of improving the root and the next three lines use 'variance.mtb' to recalculate the values U , W , X , Y and Z appropriate to the new value of $\hat{K}_m$, to calculate an updated $\hat{V}_{max.}$, and to print these estimates. The last line ensures that C3 contains the values of e_i appropriate to these estimates. In practice, as the simulations have shown, such polishing of the estimates will usually be unnecessary.

Worked example

In a seminal paper Wilkinson [1] analysed initial-velocity data from the enzymic formation of NAD^+ . These data and their analysis are shown in Table 1A. During analysis by 'estimate.mtb' the intermediate values of $\hat{K}_m$ and $\hat{V}_{max.}$ are 0.519762 and 0.649190 with final estimates 0.555442 and 0.670117. The value of weighted sum of squares of residuals, Ω , printed as K5 by 'residual.mtb' is 0.0041701, which is the sum of C5 before division by $n-2$. Estimates e_i and $Var(v_i)$ are as tabulated. The partial derivatives calculated by 'variance.mtb' are as tabulated and give variances $Var(\hat{K}_m) = 0.0061825$ and $Var(\hat{V}_{max.}) = 0.0023807$, which are K14 and K15 respectively. If 'polish.mtb' is applied three times to the then current estimate of K_m , successive estimates are 0.561093, 0.561183, and 0.561183 again, during which the function $G(K_m)$ which starts as -0.00346494 becomes -0.000053167 and then zero, while Ω decreases from 0.0041701 to 0.0041620. Successive values for $\hat{V}_{max.}$ are 0.672630, 0.672689 and 0.672689 again.

REFERENCES

1. Wilkinson, G. N. (1961) *Biochem. J.* **80**, 324–332
2. Ryan, B. F., Joiner, B. L. & Ryan, T. A. (1985) *Minitab Handbook*, 2nd edn., Duxbury Press, Boston
3. Jones, M. E. & Taransky, K. (1991) *Comput. Biol. Med.* **21**(6), 459–464