It is easy to forget that AIC and MDL are just fancy statistical tools that were invented to aid the scientific process. They are not the arbiters of truth. Like any such tool, they are blind to the quality of the data and the plausibility of the models under consideration. They will be most useful when considered in the context of the other selection criteria outlined at the beginning of this chapter (e.g., interpretability, falsifiability).

Acknowledgments

Both authors were supported by NIH Grant R01 MH57472.

[15] Practical Robust Fit of Enzyme Inhibition Data

By PETR KUZMIČ, CRAIG HILL, and JAMES W. JANC

Introduction

The analysis of enzyme inhibition data in the context of preclinical drug screening presents unique challenges to the data analyst. The good news lies in the advances in laboratory robotics, miniaturization, and computing technology. However, the good news is also the bad news. Now that we *can* perform thousands of enzyme assays at a time, how do we sensibly manage and interpret the vast amount of generated information? New methods of biochemical data analysis are needed to match the improvements in research hardware.

Another challenge is presented by increasing constraints on material resources, given the need to assay an ever larger number (thousands or hundreds of thousands) of individual compounds in any given project. Of course, only a small number of hits advance from high-throughput screening, using a single-point assay, into the dose–response screening to determine the inhibition constant. Even so, the sheer number of enzyme inhibitors that need to be screened often leads to suboptimal experimental design.

For example, a strategic decision may have been made that all inhibitor dose–response curves will contain only 8 data points (running down the columns of a 96-well plate), and that the concentration–velocity data points will not be duplicated, so that each 96-well plate can accommodate up to 12 inhibitors. With only eight nonduplicated data points, and with as many as four adjustable nonlinear parameters (e.g., in the four-parameter logistic

0076-6879/04 $35.00

equation), the experimental data better be extremely accurate and the concentrations optimally chosen. Alas, too often neither is the case.

This chapter is concerned with one particular nuisance arising in secondary preclinical screening of enzyme inhibitors, namely, the presence of gross outliers. For our purposes, outliers are data points that are affected by gross errors caused by malfunctioning volumetric equipment, by a human error in data entry, or by countless other possible mishaps. It is shown that Huber's Minimax approach to robust statistical estimation is particularly preferable over the conventional least-squares analysis.

Theory

Iteratively Reweighted Least Squares

Assume that the dependent variable y is related to the independent variable x through the functional relationship $y = f(x, \mathbf{p})$, where $\mathbf{p}$ is the m vector of adjustable model parameters to be estimated from the available data pairs $\{\{x_1, y_1\}, \{x_2, y_2\}, \ldots, \{x_n, y_n\}\}$. The usual ordinary least-squares[1–3] (OLS) estimation problem can be formulated as is shown in Eq. (1).

$$\min_{\mathbf{p}} S = \sum_{i=1}^{n} (y_i - f(x_i, \mathbf{p}))^2 \tag{1}$$

Many efficient computational methods exist to accomplish this minimization. Unfortunately, the OLS estimate of the model parameters is sensitive to the presence of outliers,[4] which has led to the design of various alternatives. For example, according to Eq. (2), instead of minimizing the sum of squared deviations, one might minimize the sum of absolute deviations.

$$\min_{\mathbf{p}} S = \sum_{i=1}^{n} | y_i - f(x_i, \mathbf{p})| \tag{2}$$

Computationally, the least absolute deviation (LAD) fit is more difficult than OLS. One approach[4] is to resort to derivative-free methods, such as the Nelder–Mead simplex algorithm.[5] A more feasible approach, leading

[1] M. L. Johnson and S. G. Frasier, *Methods Enzymol.* **117,** 301 (1985).
[2] M. L. Johnson and L. M. Faunt, *Methods Enzymol.* **210,** 1 (1992).
[3] M. L. Johnson, *Anal. Biochem.* **206,** 215 (1992).
[4] M. L. Johnson, *Methods Enzymol.* **321,** 417 (2000).
[5] J. A. Nelder and R. Mead, *Computer J.* **7,** 308 (1965).

to the same results, is to use *iteratively reweighted least squares*. This is based on the fact that the LAD fit can be accomplished as a sequence of weighted LS fits.

$$\text{repeat} \quad \min_{\mathbf{p}} S = \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2 \tag{3}$$

In each step, the best-fit values $\hat{y}_i = f(x_i, \hat{\mathbf{p}})$ from the previous LS fit are used to recompute weights, such that $w_i = 1/|y_i - \hat{y}_i|$. Throughout this chapter the ''hat'' accent (^) represents ''best-fit'' quantities. After a sufficient number of reweighted LS minimizations, the model parameters **p** converge to the same values that would be obtained by LAD minimization using, e.g., the simplex method.[5]

A similar iteratively reweighted least-squares procedure forms the basis of the robust fit method discussed in this chapter.

Huber's Method

The LAD fit has been used occasionally for data analysis in biochemical kinetics.[6] It does solve the outlier problem, but it is probably not appropriate in most experimental situations arising in biochemistry. As had been pointed out,[4] the LAD fit does not provide *maximum likelihood* parameter estimates (ML, or M estimates) if the underlying statistical distribution of random errors is Gaussian, according to probability density function (4):

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\left(\frac{x-\mu}{\sigma}\right)^2\right] \tag{4}$$

The LAD fit does produce ML parameter estimates if (and only if) the underlying error distribution function is a double-sided exponential, but such distribution is seen only infrequently.[6] On the other hand, the very presence of outliers in real-world experimental data proves the fact that strictly Gaussian error distribution is also unrealistic. What is the solution to this quandary?

Huber[7] proposed that random experimental errors arising in the physical sciences could be described by using the *contaminated Gaussian distribution* [Eq. (5)], where $\Phi(x)$ is the cumulative normal distribution:

[6] I. B. C. Matheson, *Comput. Chem.* **14,** 49 (1990).
[7] P. J. Huber, ''Robust Statistics.'' John Wiley & Sons, New York, 1981.

$$F(x) = (1 - \varepsilon)\Phi\left(\frac{x - \mu}{\sigma}\right) + \varepsilon\Phi\left(\frac{x - \mu}{3\sigma}\right) \quad (5)$$

$$\Phi(x) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-y^2/2} dy \quad (6)$$

According to Huber,[7] ε is typically in the range between 0.01 and 0.1, which does not imply that between 1 and 10% of all experiments necessarily are affected by gross errors, although this may be true in particular circumstances. The assumption that $0.01 \leq \varepsilon \leq 0.1$ merely implies the existence of two distinct categories of measurements, the majority are ''good'' points with the standard deviation σ, and a few are ''bad'' points drawn from another Gaussian distribution, characterized by the standard deviation several times larger (e.g., 3σ).

Starting from similar distributional assumptions, and from the central role of statistical *influence functions*,[8,9] a rigorous theory of robust estimation had been built.[7,10–12] Regardless of the particular form of the influence function, many computational algorithms for robust regression analysis rely on iteratively reweighted least squares,[13–16] as does Huber's method used here.

Huber's influence function[7] is constructed as follows. All ''good'' data points (to be defined below) are assigned the same weight in the iteratively reweighted series of LS estimations, exactly as they are in OLS. In contrast, deviant or ''bad'' points are progressively deemphasized, by being assigned progressively smaller weights according to Eq. (7). Here, a ''good'' data point is one for which the *standardized residual* R_i, defined in Eq. (8), is smaller in absolute value than a certain multiple of the estimated standard deviation of fit. The cutoff criterion c serves as an empirical tuning constant.

[8] D. A. Belsley, E. Kuh, and R. E. Welsch, ''Regression Diagnostics: Identifying Influential Data and Sources of Collinearity.'' John Wiley & Sons, New York, 1980.
[9] R. D. Cook and S. Weisberg, ''Residuals and Influence in Regression.'' Chapman & Hall, New York, 1982.
[10] W. J. J. Rey, ''Introduction to Robust and Quasi-Robust Statistical Methods.'' Springer-Verlag, New York, 1983.
[11] F. R. Hampel, E. M. Roncheti, P. J. Rousseeuw, and W. A. Stahel, ''Robust Statistics.'' John Wiley & Sons, New York, 1986.
[12] P. J. Rousseeuw and A. M. Leroy, ''Robust Regression and Outlier Detection.'' Wiley Interscience, New York, 1987.
[13] P. W. Holland and R. E. Welsch, *Commun. Stat. Theory Methods* **A6,** 813 (1977).
[14] D. Coleman, P. Holland, N. Kaden, V. Klema, and S. C. Peters, *ACM Trans. Math. Software* **6,** 327 (1980).
[15] J. O. Street, R. J. Carroll, and D. Ruppert, *Am. Stat.* **42,** 152 (1988).
[16] R. Heiberger and R. A. Becker, *J. Comput. Graphics Stat.* **1,** 181 (1992).

$$w_i = \begin{cases} 1 & \text{if} \quad |R_i| \le c \\ c/|R_i| & \text{if} \quad |R_i| > c \end{cases} \tag{7}$$

$$R_i = \frac{y_i - \hat{y}_i}{\hat{\sigma}\sqrt{1 - h_i}} \tag{8}$$

It should be noted that different authors (including manual writers for advanced statistical software packages, such as S-PLUS, SAS, and MATLAB) variously refer to R_i either as standardized residuals or as *Studentized residuals*. This confusion is clearly explained by Rawlings.[17]

Importantly, Huber established that with $c = 1.345$, the M estimator so defined is 95% efficient. By ''efficiency'' we mean the ratio of variances from Huber's M estimate relative to normal regression models, assuming that the underlying error distribution is in fact normal.

The standard deviation of fit, $\hat{\sigma}$, is estimated from the median absolute deviation (MAD), computed as the median absolute deviation of the residuals from their median. In Eq. (9), MAD is divided by the factor relating the probable error (E_{50}) to the standard deviation (SD): $E_{50} = 0.6745 \times SD$. Note that MAD relates to the mean square error (MSE) as MAD $\approx$ (MSE)$^{1/2}$.

$$\hat{s} = \frac{\text{med}\{|(y_i - \hat{y}_i) - \text{med}\{y_i - \hat{y}_i\}|\}}{0.6745} \tag{9}$$

''Hat'' Matrix and Nonlinear Leverages

The quantity h_i appearing in the denominator of Eq. (8) is the *leverage* of the ith data point in the nonlinear least-squares regression. It is the diagonal element of the $n \times n$ ''hat'' matrix $\mathbf{H}$ defined in Eq. (10), where $\mathbf{J}$ is the familiar $n \times m$ Jacobian matrix of first derivatives.[18] Recall that m is the number of adjustable parameters in the fitting model:

$$\mathbf{H} = \mathbf{J}(\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1}\mathbf{J}^{\mathrm{T}} \tag{10}$$

$$\{J\}_{ij} = \frac{\partial f(x_i, \hat{\mathbf{p}})}{\partial p_j} \tag{11}$$

[17] J. O. Rawlings, ''Applied Regression Analysis—A Research Tool.'' Wadsworth, Belmont, CA, 1988.

[18] M. L. Johnson, *Methods Enzymol.* **321,** 425 (2000).

In linear regression, the experimental values **y** and the least-squares fit values $\hat{\mathbf{y}}$ are related through the simple matrix equation $\hat{\mathbf{y}} = \mathbf{H}\mathbf{y}$, thus the term ''hat'' matrix. The matrix **H** has interesting mathematical properties, many of which are practically useful for checking the computation of h_i. For example, **H** is a symmetric and idempotent projection matrix, that is, $\mathbf{HH} = \mathbf{H}$. It has m eigenvalues equal to 1 and $n - m$ eigenvalues equal to 0. The trace is $tr\{H\} = p$, and for all diagonal elements, we must have $0 \leq h_i \leq 1$.

Because we are interested only in the diagonal elements of the hat matrix, we can compute them directly by using Eq. (12), where $\mathbf{j}_i$ is the ith row vector in the Jacobian matrix **J**.

$$h_i = \mathbf{j}_i(\mathbf{J}^{\mathrm{T}}\mathbf{J})^{-1}\mathbf{j}_i^{\mathrm{T}} \tag{12}$$

The $m \times m$ matrix inverse $(\mathbf{J}^{\mathrm{T}}\,\mathbf{J})^{-1}$ is hardly ever computed as written. Instead, in our work we utilize the QR decomposition[19] $\mathbf{J} = \mathbf{QR}$, where **R** is an $m \times m$ upper triangular invertible matrix with positive entries in its diagonal. Many good implementations of the QR decomposition algorithm are available as canned software.[20]

Kinetic Model

The kinetics of tight-binding enzyme inhibition[21,22] is described here by Eq. (13), where V_{b} is the baseline or background reaction rate, V_0 is the reaction rate observed in the absence of inhibitor (''negative control''), [E] and [I] are, respectively, the concentrations of the enzyme and the inhibitor, and K_{i} is the apparent inhibition constant.[23]

$$v = V_{\mathrm{b}} + V_0\frac{[\mathrm{E}] - [\mathrm{I}] - K_{\mathrm{i}} + \sqrt{([\mathrm{E}] - [\mathrm{I}] - K_{\mathrm{i}})^2 + 4[\mathrm{E}]K_{\mathrm{i}}}}{2[\mathrm{E}]} \tag{13}$$

Numerical Example

The experimental data in the first three columns of Table I represent the micromolar concentration of an inhibitor (x_i) and the corresponding initial velocities of an enzyme reaction (y_i).

[19] D. C. Lay, ''Linear Algebra and Its Applications.'' Addison-Wesley, Reading, MA, 1994.
[20] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and Brian P. Flannery, ''Numerical Recipes in C.'' Cambridge University Press, Cambridge, 1992.
[21] J. F. Morrison, *Biochim. Biophys. Acta* **185,** 269 (1969).
[22] J. W. Williams and J. F. Morrison, *Methods Enzymol.* **63,** 437 (1979).
[23] S. Cha, *Biochem. Pharmacol.* **24,** 2177 (1975).

TABLE I
RESULTS OF FIT FOR REPRESENTATIVE ENZYME INHIBITOR[a]

	Data		Least-squares fit				Robust fit			
i	x_i	y_i	$\hat{y}_i$	r_i	R_i	h_i	$\hat{y}_i$	r_i	R_i	w_i
1	0	133.0	143.4	10.4	−0.51	0.51	139.8	−6.8	−0.27	1
2	0.0031	143.6	135.5	8.1	0.35	0.36	136.8	6.8	0.24	1
3	0.0122	132.8	115.9	16.9	0.68	0.27	128.6	4.2	0.14	1
4	0.0488	34.0	71.3	−37.3	**−1.95**	0.57	103.3	**−69.3**	**−2.98**	0.12
5	0.195	65.8	27.0	**38.8**	1.55	0.26	57.2	8.6	0.28	1
6	0.781	13.5	7.6	5.9	0.2	0.03	20.3	−6.8	−0.19	1
7	3.125	3.4	2.0	1.4	0.05	0	5.6	−2.2	−0.06	1
8	12.5	1.1	0.5	0.6	0.02	0	1.5	−0.4	−0.01	1
9	50	0	0.1	−0.1	0	0	0.4	−0.4	−0.01	1

[a] Values in boldface represent the maximum absolute value in the given column.

Ordinary Least-Squares Fit

According to Huber's method, the robust regression analysis always begins with the ordinary least-squares fit, summarized in columns 4 through 7 in Table I. In the OLS fit to Eq. (13), the enzyme concentration ($[E] = 10\ nM$) and the background rate ($V_b = 0$) were treated as fixed constants; the apparent inhibition constant K_i and the control rate V_0 were treated as adjustable model parameters. The best fit values are shown in the first row of Table II.

It is important to note the difference between ordinary residuals $r_i = y_i - \hat{y}_i$ and the standardized residuals R_i defined by Eq. (8). Examining the ordinary residuals r_i, one might be tempted to conclude that the fifth data point ($i = 5$) could be an outlier, because it has the largest absolute deviation. In contrast, the standardized residuals R_i suggest that the fourth data point could be an outlier, because it is associated with the largest (in absolute value) standardized residual.

This difference between r_i and R_i is caused by the nonlinear leverages h_i for each data point. Note that the leverage for the fourth data point (0.57) is more than twice the leverage for the fifth data point (0.26), which means that in the iteratively reweighted least squares the fourth data point will be initially given larger weight $[1/(1 - 0.57)^{1/2} = 1.52]$, compared with the fifth data point $[1/(1 - 0.26)^{1/2} = 1.16]$.

The leverages h_i for data points 7 through 9 are practically zero, which means that these data points contribute practically no useful information.

TABLE II
RESULTS OF FIT USING VARIOUS ANALYSIS METHODS[a]

Method	K_i (nM)	V_0	$n_{w<1}$	$\sum w_i$
Least squares	43.3 ± 25.1	143.4 ± 15.8	0	9
Huber ($c = 10$)	43.3 ± 25.1	143.4 ± 15.8	0	9
Huber ($c = 1.345$)	131.0 ± 47.0	139.8 ± 8.3	1	8.12
Huber ($c = 1$)	68.5 ± 17.2	151.4 ± 4.4	3	6.23
Huber ($c = 0.1$)	75.2 ± 3.2	149.8 ± 1.2	9	2.67
Huber ($c = 0.01$)	73.5 ± 3.1	150.7 ± 1.2	9	0.63
Huber ($c = 0.001$)	71.1 ± 0.2	151.9 ± 1.3	9	0.14
Absolute deviations (simplex)	77.0	148.7	—	—
Point deletion	146.1 ± 23.0	140.8 ± 3.7	0	8

[a] $n_{w<1}$ is the number of data points for which the final weight [Eq. (7)] was lower than unity; $\sum w_i$ is the sum of all final weights in the iteratively reweighted least squares.

This indicates a suboptimal experimental design. Huber[7] points out that large values of h_i should "serve as warning signals that the ith observation may have a decisive, yet hardly checkable, influence. Values $h_i \leq 0.2$ appear safe, values between 0.2 and 0.5 are risky, and if we can control the design at all, we had better avoid values above 0.5."

From this discussion it is clear the fourth data point is what Huber calls a *leverage point* (i.e., a data point associated with a dangerously high value of h_i), whereas data points 7 through 9 are useless. This is yet another unpleasant consequence of one-size-fits-all experimental designs traditionally seen in inhibitor screening. Most often, the same dilution ratio and the same maximum concentration are used for all inhibitors on the same 96-well plate, but if the inhibitors differ significantly in their inhibition constants, this uniform design generates a large number of data points with low information value.

Robust Fit

In the second stage of this analysis, the leverages h_i computed during the preliminary OLS fit were used in the iteratively reweighted OLS regression, employing the default value of Huber's tuning constant ($c = 1.345$). After several repeated OLS fits, the adjustable parameters converged to the values listed in the third row of Table II. The results of the fit are shown graphically in Fig. 1.

Note in Table I that the Huber reweighted regression ended with assigning unit weights (that is, giving the ordinary least-squares treatment)

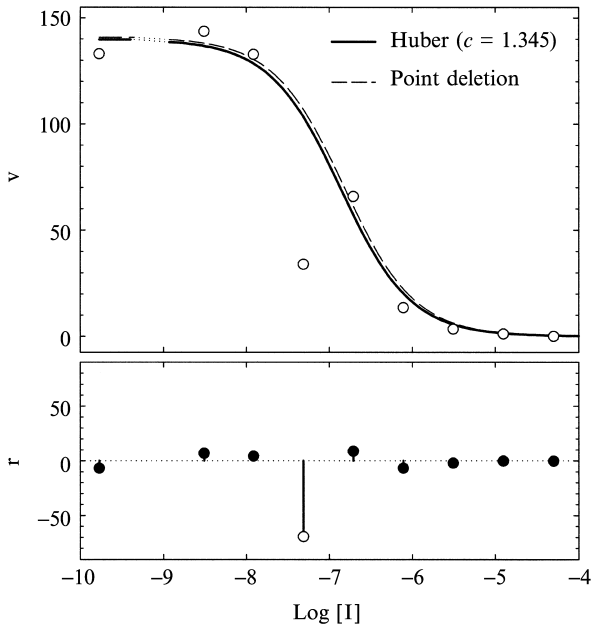

FIG. 1. *Top:* The open circles are data points (inhibitor concentrations versus initial velocities) for a particular enzyme inhibitor. The left most data point is the negative control, observed at zero inhibitor concentration. The thicker, solid curve represents the robust fit to rate Eq. (13) by using Huber's method with tuning constant $c = 1.345$. The thinner, dashed curve represents the results of the ordinary least-squares fit after the fourth data point ($\log[I] \approx -7.3$) was deleted. *Bottom:* The residuals for data points that were assigned the full unit weight ($w_i = 1$) in Huber's method are shown as solid circles. The residual shown as an open circle belongs to the (single) data point, which ended up with less than unit weight ($w_4 = 0.12$) in the iteratively reweighted least-squares fit.

to all data points except data point 4, which is assigned a small weight ($w_i = 0.12$). This result strongly suggests that the fourth data point is an outlier. Its standardized residual is almost equal to three ($R_i = 2.98$), which is yet another strong indication that the data point is affected by gross error.

Some authors[8] recommend that data points with $R_i > 2$ should simply be deleted. Others recommend a two-stage robust regression, starting with Huber's influence function (7) followed by Tukey's biweight scheme (14), where the 95% asymptotic efficiency of the standard normal distribution is achieved with $c = 4.6851$.

$$w_i = \begin{cases} 0 & \text{if} \quad |R_i| > c \\ [1 - (R_i/c^2)]^2 & \text{if} \quad |R_i| \leq c \end{cases} \tag{14}$$

Note that Tukey's outliers are given zero weights, and thus are effectively excluded from analysis. We have experimented with Tukey's biweight and found that, too often, it deleted too many data points from our small data sets. Instead, in the context of inhibitor screening, we formulated a heuristic policy for data exclusion as follows. If and only if the Huber method produces a single data point with the final weight $w_i < 1$, this single data point is deleted (by being assigned $w_i = 0$), and the analysis is repeated one last time as ordinary least squares. For our example inhibitor, where this condition does apply, the results are shown graphically as the thin dashed curve in Fig. 1.

The solid and dashed curves in Fig. 1 do appear pleasingly similar, suggesting that Huber's fit with $c = 1.345$ and OLS fit with the fourth data point deleted are consistent with each other. The best-fit values of adjustable parameters, shown for the OLS fit with deletion in the last row in Table II, are also comparable for the two methods, although they are not identical. The difference is caused by the outlier point being assigned a nonzero weight, $w_4 = 0.12$, in Huber's method. However, the difference between $K_\mathrm{i} = 131$ nM (Huber) and $K_\mathrm{i} = 146$ nM (OLS with deletion) is only about 10%, whereas the OLS method with full data set produced an inhibition constant ($K_\mathrm{i} = 43$ nM) that is off by more than a factor of three. Thus, the application of Huber's method alone produced two desirable effects. First, it reduced the systematic error in K_i due to a single outlier, from 330 to 10%. Second, it clearly diagnosed the presence of this outlier so it could be deleted.

Variations in Huber's Tuning Constant

Although the particular value $c = 1.345$ for Huber's tuning constant is rooted in statistical theory (it has been chosen because it leads to an M estimate that is 95% efficient), it is important to examine in practice how variations in c might affect the outcome of robust regression analyses in our particular experimental setting.

On the basis of theoretical considerations [see Eq. (7)], we can predict that as c becomes *very* large all data points will be assigned unit weights and the Huber regression turns into OLS. On the other hand, as c becomes small, we expect the Huber algorithm to resemble the LAD fit, because the weighting factors in the influence function (7) simply become reciprocal absolute residuals.

Figure 2 and the fourth row in Table II show the results of Huber regression with $c = 1.0$. Although this is only marginally lower than the recommended value $c = 1.345$, the results of fit are strikingly different. As is illustrated by the bottom panel in Fig. 2, the robust fit is dominated by six of the nine data points, which are assigned unit weights. Data points 1, 4, and 5 are deemphasized with weights $w_1 = 0.11$, $w_4 = 0.03$, and $w_5 = 0.10$. In other words, the Huber fit discovered too many outliers.

Further decreasing c to 0.1 created another problem. With $c = 0.1$ or lower, no points were assigned full weights in reweighted least squares. The best-fit values of model parameters remained approximately the same between $c = 0.1$ and $c = 0.001$, but the variances of the model parameters decreased drastically. This is expected from theory, because the Huber M estimate loses asymptotic efficiency as c moves away from its 95% efficient value of 1.345. Thus in any software system in which the Huber tuning

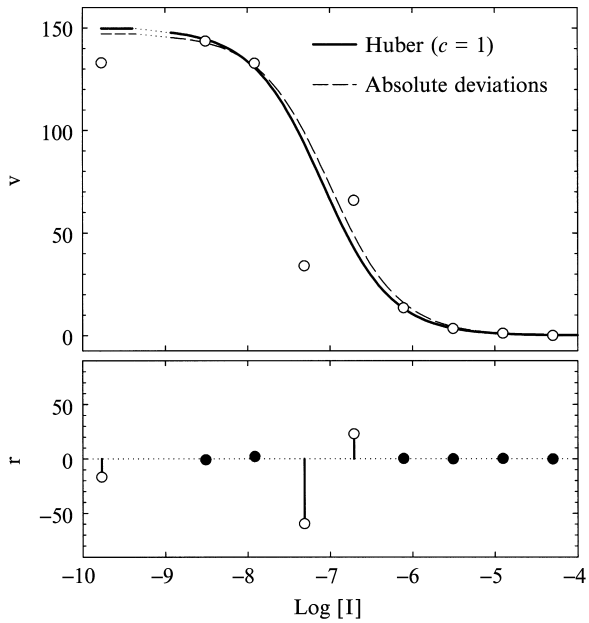

FIG. 2. *Top:* The thicker, solid curve represents the robust fit to rate Eq. (13) by using Huber's method with tuning constant $c = 1.0$. The thinner, dashed curve represents the results of the least-absolute deviation fit [Eq. (2)] using the Nelder–Mead simplex algorithm.[5,20] *Bottom:* For explanation of solid and open circles, see Fig. 1. Please note that the fit is completely dominated by six of the nine data points.

constant c is adjustable by the user (e.g., SAS, S-PLUS, MATLAB, or in the software built by us for inhibitor screening), one must proceed with caution. Lowering c might not only pick up too many ''outliers'' if the data set is small, it will also unrealistically shrink parameter variances.

As is expected from theoretical considerations, when we increased the tuning constant c above its 95% efficient value ($c = 1.345$), the algorithm simply turned into OLS. This is seen from Fig. 3 and the second row in Table II.

In some respects, these results are disconcerting. At least for this particular inhibitor, decreasing c only slightly (in fact, from 1.345 to 1.3; see Fig. 4) has led to the false identification of too many ''outliers.'' In contrast, increasing c eventually missed the single outlier altogether. An important question then concerns how wide a range c can have for the Huber method to remain useful for analyzing data sets as small as ours are (nine data points, two to four adjustable parameters).

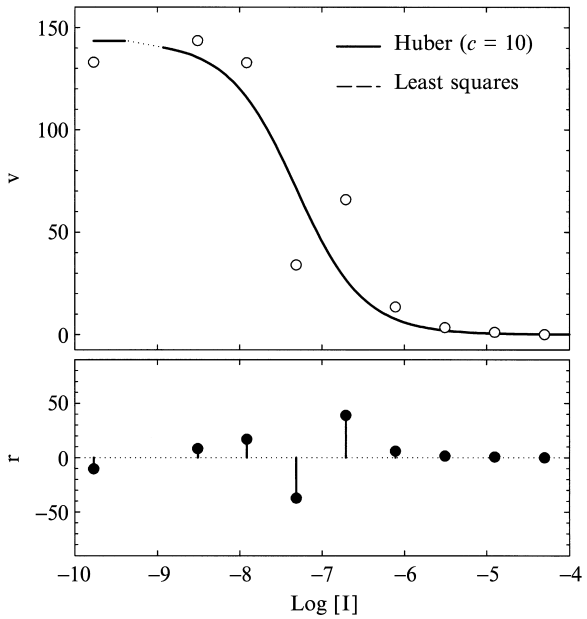

FIG. 3. *Top:* The thicker, solid curve represents the robust fit to rate Eq. (13) by using Huber's method with tuning constant $c = 10.0$. The thinner, dashed curve represents the results of the ordinary least-squares fit [Eq. (1)]. *Bottom:* For explanation of solid and open circles, see Fig. 1. Please note that two regression analyses yielded exactly identical results, as the two best-fit curves are indistinguishable.

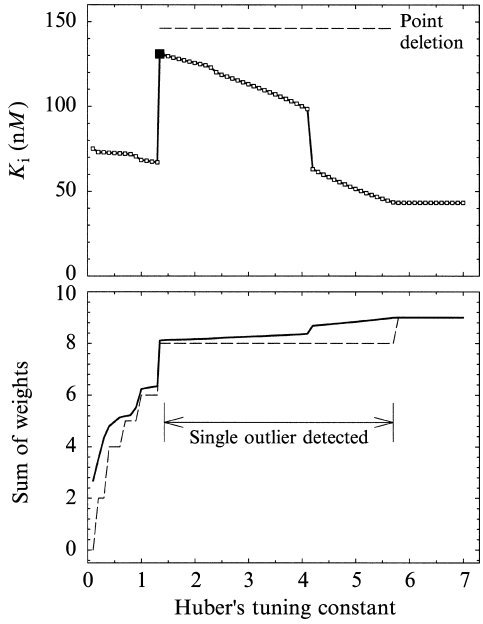

FIG. 4. *Top:* Variation in the best-fit value of the apparent inhibition constant K_i depending on the value of the Huber tuning constant c in Eq. (7). The large solid square is the value obtained at the default recommended value, $c = 1.345$. *Bottom:* The solid curve shows the sum of weights $\sum_{i=1}^{n} w_i$ obtained for the nine data points ($n = 9$) in the Huber regression, depending on the value of c. The dashed curve is the sum of all weights while counting only data points with full weights ($w_i = 1$).

To answer this question, we have varied c systematically between 0.1 and 10.0, stepping by $\Delta c = 0.1$ (100 different values), and performed the Huber regression at each point. The results are summarized in Fig. 4.

It is encouraging that the range of c values, within which the Huber algorithm successfully detected only a single data point with the final weight $w_i < 1$, is relatively wide ($c = 1.345$ through $c = 5.6$). Unfortunately, as the weight of this data point increases, the outlier progressively distorts the best-fit value of the inhibition constant.

Also note that the default value of the tuning constant ($c = 1.345$) precariously sits at the lower end of this interval. Thus, perhaps a minuscule change in the data could cause the algorithm to tip over that edge, and suggest falsely that the data set contains three ''outliers'' instead of one. These are serious challenges for the designer and administrator of a software

system designed for automatic, robust, high-throughput analysis of enzyme inhibition data.

Implementation Notes

In the course of our ongoing work on new methods for automated data analysis in preclinical screening,[24–26] we have tested Huber's robust regression method on tens of thousands of enzyme inhibitors. The efficiency of the method in handling occasional outliers was good. This is in contrast with the OLS fit, where data points with large deviations have unduly large influence, and with the LAD fit,[4,6] where data points with small deviations dominate the fitted curve.

The latter statement is consistent with the fact that, in LAD regression implemented as iteratively reweighted least squares, the weights are $1/|y_i - \hat{y}_i|$, implying that a data point with zero deviation has infinite weight. In working with relatively small data sets and with nonlinear fitting models, this feature of LAD is particularly dangerous. A sufficiently flexible nonlinear model with four parameters will always go through four data points, completely ignoring the remaining data, which become deemphasized in LAD fit.

The practical success of Huber's method applied even to relatively small data sets, such as those arising in preclinical screening, is due to the fact that the method behaves as OLS does if the data are ''good,'' but at the same time it gives the LAD treatment to suspected outliers, while maintaining 95% asymptotic efficiency. The following are a few of many possible implementation issues, which were encountered in translating the theory of Huber's regression into a practically useful software system.

Replicated Measurements

Huber's robust regression method is suitable for the analysis of either replicated or nonreplicated data. However, when the number of replicates is small, it is best not to automatically average these replicates (e.g., duplicates) and then analyze the averaged data. Consider the hypothetical example where the replicated initial velocities are [100, 99], [81, 79], [62, **30**], [40, 38], [19, 21], and so on. These are 10 data points, only one of which

[24] P. Kuzmič, S. Sideris, L. M. Cregar, K. C. Elrod, K. D. Rice, and J. W. Janc, *Anal. Biochem.* **281,** 62 (2000).
[25] P. Kuzmič, K. C. Elrod, L. M. Cregar, S. Sideris, R. Rai, and J. W. Janc, *Anal. Biochem.* **286,** 45 (2000).
[26] P. Kuzmič, C. Hill, M. P. Kirtley, and J. W. Janc, *Anal. Biochem.* **319,** 372 (2003).

is clearly an outlier. By averaging we obtain five data points [99.5], [80], [**46**], [39], [20], among which the outlier would be more difficult to detect if the fitting model is nonlinear.

In this hypothetical example, a better alternative to robust regression might be *weighted least squares* (WLS) fit, where the weighting factors are reciprocal SDs from each replicate. Our experience shows that WLS with a small number of replicates can be treacherous for the following reason. If the inhibitor dose–response curve contains only a small number of (replicated) data points, it is possible that one particular duplicate might be fortuitously accompanied by a small standard error, much smaller than the standard errors from other averages. In such a case, the particular data point would be assigned a disproportionately large weight, and consequently it might unduly influence the regression. In a production software system, the user or the administrator should have a choice to decide whether to use robust regression or WLS, but preferably not both at the same time.

Outliers versus Deviations from Fitting Model

Practical experience with Huber's regression shows that in many cases the method will assign weights smaller than one to more than one data point, and sometimes even to all data points in the given dose–response curve. Obviously not all data points can be ''outliers'' if our distributional assumption [Eq. (5)] is correct. Rather, too many ''outliers'' (data points with $w_i < 1$) simply suggest that the fitted model is incorrect.

In such cases, a sensible software system would disregard the robust fit and revert to OLS with a suitable warning. Alternately, if only one of the weights is much smaller than the others, it might make sense to delete the corresponding data point (by setting its $w_i = 0$) and run one final OLS fit. A further refinement of this policy would be to take into account the total number of data points with $w_i < 1$, or the sum of weights for the entire data set (see Fig. 4).

One might be willing to accept the results of Huber's robust regression analysis only if the majority of data points end up with $w_i = 1$, or otherwise conclude that the model is inadequate, issue a warning, and end with OLS as the last resort. However the software system is constructed, it is important to avoid situations illustrated in Fig. 2, where few data points completely dominate the regression while the remaining data points are effectively excluded through weighting. Again, in a production software system, the user or the administrator should be able to control these policies.

Conclusions

We have discussed several mathematical quantities that should be of interest to the biochemical data analyst, but, to our knowledge, are hardly ever mentioned in the mainstream biochemical literature. First, standardized residuals defined by Eq. (8) are significantly more informative than ordinary residuals ($y - \hat{y}_i$). Standardized residuals are more helpful than ordinary residuals not only for outlier detection, but also for model diagnostics.

Second, nonlinear leverages, which are diagonal elements of the ''hat'' matrix **H** [Eq. (10)], are useful for quickly assessing the presence of unduly influential data points (whether outliers or not) and the optimality of experiment design. If, after a least-squares fit, we find that a particular data point is associated with $h_i > 0.7$, it means that this is a leverage point. Small, random changes in this single data value might have a large effect on model parameters, which is undesirable.

On the other hand, if we find that too many data points are associated with zero leverages ($h_i = 0$), it means these data points were wasted, because they contribute no useful information at all about the model parameters. In such case, one should seriously consider improving the experimental design (in the case of inhibitor screening, the layout of concentrations) for the next round of experiments.

Both standardized residuals and leverages play a role in the Huber's method of robust regression analysis, implemented as iteratively reweighted least squares. We found that it is a good alternative to ordinary least squares when the goal is to exclude a single gross outlier from a relatively small data set. This approach can increase productivity in preclinical screening laboratories, faced with determining inhibition constants for thousands of enzyme inhibitors in a single project.

Kinetic data analysis does present unique challenges in an extensively automated and robotized enzymology laboratory, where success means a smooth flow of the massive data stream connecting microtiter plate readers with structure–activity databases. At the present time, no technology can completely replace a well-qualified enzymologist supervising the process. However, our practical experience shows that a software system judiciously implementing Huber's variant of robust regression does help in the specific task of objectively identifying grossly outlying data points.