



## Research Article

# Simplified Algorithms for Determining Cycle Shift between qPCR Fluorescence Curves

Michael E. Jones,<sup>1</sup> George C. Mayne,<sup>2</sup> Tingting Wang,<sup>2</sup>  
 David I. Watson,<sup>2</sup> and Damian J. Hussey<sup>2</sup>

<sup>1</sup>Department of Anaesthesia and Pain Management and Department of Anatomy, Flinders University, Bedford Park, SA 5042, Australia

<sup>2</sup>Department of Surgery, Flinders University, Bedford Park, SA 5042, Australia

Correspondence should be addressed to Michael E. Jones; michael.e.jones@flinders.edu.au

Received 16 September 2014; Revised 31 December 2014; Accepted 13 January 2015

Academic Editor: Andre Fujita

Copyright © 2015 Michael E. Jones et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

The polymerase chain reaction is a central component of current molecular biology. It is a cyclic process, in each early cycle of which the template DNA approximately doubles. An indicator which fluoresces when bound to DNA quantifies the DNA present at the end of each cycle, giving rise to a fluorescence curve which is characteristically sigmoid in shape. The fluorescence curve quantifies the amount of DNA initially present; the more the initial DNA, the earlier the rise in the fluorescence. Accordingly the amount of DNA initially present in two samples can be compared: the sample with the less DNA gives rise to a relatively delayed fluorescence curve and the ratio of the DNAs can be deduced from the separation of the curves. There is, however, a second determinant of this separation, the fold increase in DNA per cycle: ideally a twofold increase but frequently less. Current guidelines recommend that this be determined experimentally by carrying out PCR on a series of dilutions. If the value of the fold increase is known, then the algorithm for determining the separation can be reduced to a relatively simple computation, rather than employing a multidimensional nonlinear optimization such as the Marquardt-Levenberg as currently employed.

## 1. Introduction

The polymerase chain reaction (PCR) introduced by Mullis et al. [1] is a cyclic process, each cycle of which involves three stages: denaturation, annealing (to a primer, possibly also to some forms of fluorescent probe), and extension. Optimally each strand of template DNA gives rise to its complement in each cycle: the amount of DNA doubles each cycle. In reality each strand is copied with probability  $p \leq 1$ . If  $p$  were constant, then the DNA present after  $C$  cycles would be

$$N_c = N_0 (1 + p)^C, \quad (1)$$

where  $N_0$  is the amount initially present. If two samples, A and B which initially comprise  $N_a$  and  $N_b$  strands, have the same value of the probability  $p$  and if they exhibit the same fluorescence after  $C_a$  and  $C_b$  cycles, respectively, then

$$N_a (1 + p)^{C_a} = N_b (1 + p)^{C_b}, \quad (2)$$

implying

$$\frac{N_a}{N_b} = (1 + p)^{(C_b - C_a)} = (1 + p)^{\Delta C}, \quad (3)$$

where  $\Delta C = C_b - C_a$ .

In practice this simple “exponential” model for the growth in fluorescence is complicated by three factors. As resources are depleted and as template DNA strands bind to their complement rather than to primer, the value of  $(1 + p)$  in (1) diminishes: the fluorescence curve is sigmoid rather than exponential. There is a background fluorescence present initially and not indicative of DNA, and this background varies between replicates. There is also, between replicates, a variation in the scale, in that the plateau level that the fluorescence approaches also varies. These considerations give rise to a collection of models for the fluorescence curve seen in PCR. The open-source facility “qPCR” [2] offers ten nonlinear sigmoidal models (and nine others) that

can be fitted to the fluorescence data using the Marquardt-Levenberg [3–5] algorithm.

Notwithstanding the utility of the Marquardt-Levenberg algorithm and of open-source utilities such as qpcR which implement it, many scientists prefer to write and understand their own analytic software. To this end we develop, in the following sections, justification for a four-parameter model. When, as recommended by MIQE guidelines [6], the amplification efficiency is known, this reduces to a three-parameter model in which two of the parameters are linear; there is only one nonlinear parameter. The calculations are then greatly simplified and the multidimensional Marquardt-Levenberg algorithm becomes unnecessary; nonlinear optimization in a single dimension is trivial, as the problem reduces to maximizing a correlation. Finding a zero of a function is, however, even more elementary than finding a maximum, and we derive the function, the zero of which defines the value of the required parameter.

Regardless of whether the adopted approach is finding the maximum correlation or the zero of a function, the linear parameters are obtained algebraically by simple linear regression. The nonlinear search, whether for a maximum or a zero, is in a single dimension and its convergence can be guaranteed.

## 2. Theoretical Development

In the following we review briefly the exponential model for amplification and the justification of a more realistic sigmoid model. We note that finding the single nonlinear parameter is a “partially linear” problem and this gives rise to several approaches to its estimation: an approximate algebraic method (without iteration) and more exact methods involving iteration. A modicum of algebra then simplifies the problem to one of finding a zero rather than a maximum. We will then apply the methods to the estimation of the nonlinear parameter using real data in order to establish three things. First is that the maximum method and the zero method arrive at the same parameter value and that this is the same value returned by the more usual Marquardt-Levenberg approach. Second is to establish the extent to which iteration improves the algebraic approximate estimator: how much gained by the extra computation required? Finally we determine computational times: our algorithm is easier to code, but can it compete with the established Marquardt-Levenberg algorithm?

**2.1. The Exponential Model.** If, as envisaged during early cycles, the amount of DNA increases geometrically, the increase is proportional to the amount present, and the corresponding continuous function,  $y(c)$ , satisfies the differential equation

$$\frac{\partial y}{\partial c} = \beta y, \quad (4)$$

for some constant  $\beta$ . Then it is elementary that a solution is  $y(c) = y_0 e^{\beta c}$ , where  $y_0$  is the amount initially present and  $e^\beta$  is the fold increase per cycle.

**2.2. Resource-Limited Growth: A Sigmoid Model.** If limiting resources, or other factors, limit growth such that fluorescence asymptotically approaches some limit, then the differential equation,

$$\frac{\partial y}{\partial c} = \beta y (1 - y), \quad (5)$$

is perhaps the simplest to describe such a situation; for small  $y$ , growth per cycle corresponds to (4) above, but growth then diminishes as  $y$  approaches an asymptote (here, an asymptote of one). Equation (5) has the solution (Bernoulli’s method, see Appendix A):

$$y(c) = \frac{1}{1 + e^{\beta(\lambda - c)}}, \quad (6)$$

for some  $\lambda$ . Looking at (6),  $\lambda$  is the value of  $c$  for which  $y(c) = 0.5$ . For early cycles ( $\lambda \gg c$ ) the fold increase per cycle is  $e^\beta$  as before. If we add to this a baseline fluorescence,  $F_b$ , which may vary between replicates and a scale  $F_{\max}$  which may also vary between replicates (both of which are therefore essentially nuisance parameters), we have been describing the fluorescence curve which is a four-parameter sigmoid curve of the form

$$y(c) = F_b + \frac{F_{\max}}{1 + e^{\beta(\lambda - c)}}, \quad (7)$$

which is one of the more widely used models of PCR fluorescence, corresponding to model b4 of the qpcR package. As before,  $e^\beta$  is the fold increase per cycle when  $y(c)$  is small, and if this is known from a dilution series, then  $e^\beta$  or equivalently  $\beta$  is known and there are only three unknown parameters.

**2.3. The Partially Linear Problem.** Prior knowledge of  $\beta$  radically alters our approach, because there is then only one parameter entering the problem nonlinearly. This situation arises not only following the prior determination of  $\beta$  by dilution series, as in MIQE guidelines, but also during the analysis of a dilution series where  $\beta$  is determined by fixed-point convergence [7]. Note that we can write (7) in the form

$$y(c) = F_b + F_{\max} \times l(\lambda, c), \quad (8)$$

where  $l(\lambda, c)$  denotes the logistic curve

$$l(\lambda, c) = \frac{1}{1 + e^{\beta(\lambda - c)}}, \quad (9)$$

so that, for any given value of  $\lambda$ , the value of  $l(\lambda, c)$  at any cycle  $c$  is known, and fitting the model values  $\hat{y}(c)$  to the observed fluorescence values  $y(c)$  is only a problem of simple linear regression. The optimal value of  $\lambda$  is then the value for which the residual sum of squares is minimized.

Regardless of the algorithm used in the computation, all the iterative procedures used here converge on the least-squares estimates of  $F_b$ ,  $F_{\max}$ , and  $\lambda$ . That the least-squares estimate of  $\lambda$  also maximizes the Pearson product-moment correlation coefficient which follows from a well-known identity. Briefly, if the data are pairs  $(x_i, y_i)$  and we regress

the dependent variate  $y_i$  on the independent  $x_i$ , then the coefficient of determinism is denoted and defined by

$$R^2 = 1 - \frac{\text{Residual sum of squares}}{\text{Total sum of squares}}, \quad (10)$$

and the Pearson correlation coefficient is denoted and defined by

$$r_{xy} = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{s_x s_y}, \quad (11)$$

where  $\bar{x}$ ,  $\bar{y}$ ,  $s_x$ , and  $s_y$  are means and standard deviations, respectively. Then the identity states that the coefficient of determinism is, under these circumstances, equal to the square of the correlation coefficient. That is,

$$R^2 = 1 - \frac{\text{Residual sum of squares}}{\text{Total sum of squares}} = [r_{xy}]^2. \quad (12)$$

It is immediate that where the residual sum of squares depends also on a parameter  $\lambda$ , the value which minimizes the residual sum of squares must maximize the correlation coefficient. An elementary proof of the identity appears in our Github repository [8].

But having determined  $\lambda$ , what is its experimental significance? Looking at (9), note that when  $c = \lambda$ , we have  $l(\lambda, \lambda) = 0.5$ ; that is,  $\lambda$  is the fractional cycle at which the growth of fluorescence is half of its maximum. We can also show that it is the point of maximum slope of the fluorescence curve. Two samples being compared have the same template DNA and the same value of  $\beta$ ; they differ only in their values of  $\lambda$  (and in the nuisance parameters  $F_b$  and  $F_{\max}$  which do not influence  $\lambda$ ). For two samples A and B having values  $\lambda_a$  and  $\lambda_b$ , the shift of one curve relative to the other is  $\Delta C = \lambda_a - \lambda_b$ .

**2.4. The Approximate Estimator of  $\lambda$ .** Reducing the problem to finding the optimal value of a single nonlinear parameter does not avoid the necessity of finding an approximate starting value. Given that  $\lambda$  is the value at which slope of the curve  $l(\lambda, c)$  is greatest, we conjecture that the appropriate starting value for a curve such as that in Figure 1, from the publicly available *batschl* data set, would be at about  $\lambda = 29$ . Figure 2 shows the correlation between observed fluorescence and  $l(\lambda, c)$  for values of  $\lambda$  between 1 and 45. As expected, the maximum correlation is around the expected value of 29. More precisely, if we take the curve near its maximum to be quadratic, we can interpolate between the maximum cycle and the two adjacent ones to derive an interpolated fractional cycle at which the correlation would be greatest. This interpolated value is  $\lambda = 28.9207$ .

In the following analysis we will refer to this approach, seeking an approximate value of the maximum correlation by quadratic interpolation, as the *CorQuad* algorithm: it is algebraic rather than iterative and will give only an approximation to the least-squares estimate to which all other (iterative) algorithms will converge.

**2.5. The Optimal  $\lambda$  by Iterated Maximum.** Finding the maximum (or minimum) of a nonlinear function by iteration

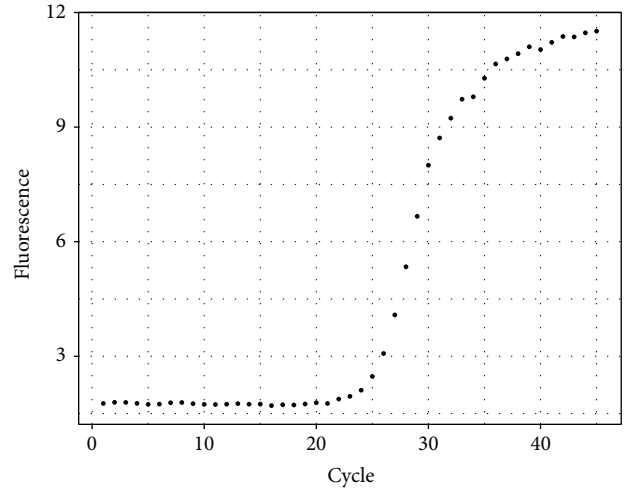


FIGURE 1: The typical sigmoid-shaped fluorescence curve obtained during qPCR. Baseline and scale vary between replicates.

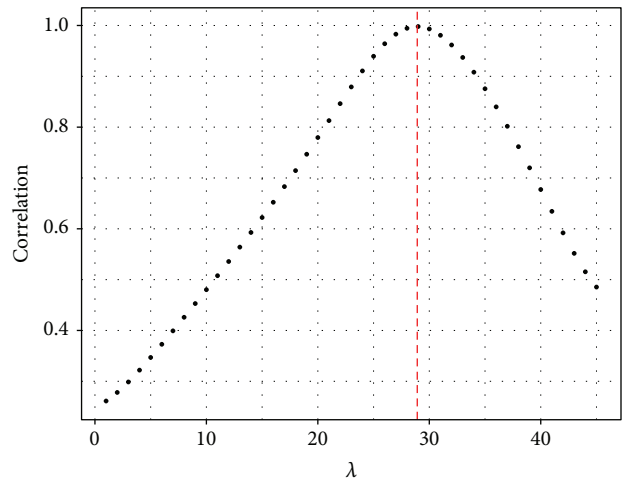


FIGURE 2: Correlation between observed fluorescence and a logistic curve. The value of  $\lambda$  maximizing the correlation is the same as the least-squares estimate obtained by other methods.

is routine once a good first approximation is known. In Figure 2, for instance, iteration arrives at a maximum correlation at  $\lambda = 28.9164$ . The nonlinear optimization here is in one dimension, for which we have used the R function *optimize()* (a mixture of golden section and quadratic interpolation) and we refer to this algorithm as *CorOpt*. Once the least-squares estimate of  $\lambda$  is known, the least-squares estimates of  $F_b$  and  $F_{\max}$  are found algebraically by simple linear regression.

**2.6. The Optimal  $\lambda$  as the Root of a Function.** Even simpler than finding the maximum of a function is finding the zero of a function. The optimal value of  $\lambda$  is that which, in maximizing the correlation, equivalently minimizes the residual sum of squares between the expected fluorescence and that found experimentally. Denoting by  $r_i$  the error or residual by which the  $i$ th observation differs from the model

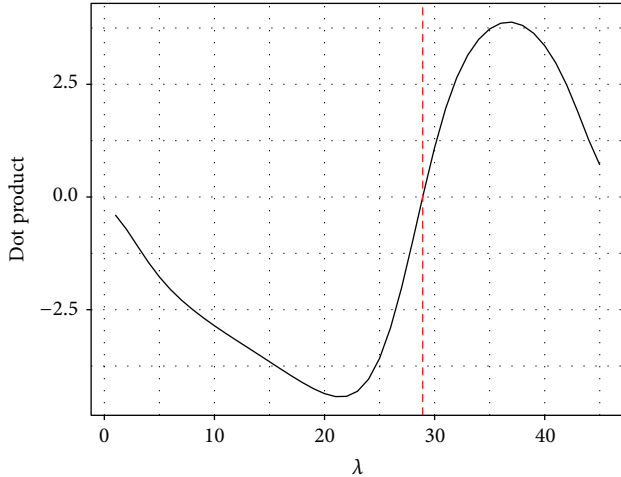


FIGURE 3: The function, the root of which defines optimal value of  $\lambda$ , is nearly linear in the vicinity of the optimum. Its shape, however, precludes the use of Newton's method to find the root unless the initial value is close to the root. The root here is found at  $\lambda = 28.9165$ .

value  $\hat{y}_i$ , the outcome of Appendix B is that if we define the function

$$g(\lambda, i) = g_i = e^{(\lambda-i)} [l(\lambda, i)]^2, \quad (13)$$

where  $l(\lambda, c)$  is the logistic as defined in (9), then the optimal value of  $\lambda$  satisfies

$$\sum_i g_i r_i = 0, \quad (14)$$

which is to say that the vector  $\mathbf{g}$  is orthogonal to the residuals  $\mathbf{r}$ .

Figure 3 is a plot of the dot product  $\sum_i g_i r_i$  for values of  $\lambda$  from 1 to 45, demonstrating its zero at the appropriate value of  $\lambda$ . Two algorithms using this approach, but with minor coding differences to illustrate variations in computational time, are hereafter referred to as *SlowZ* and *FastZ*.

### 3. Materials and Methods

Analysis was carried out using the R programming environment [9] under GNU/Linux Ubuntu 14.04 LTS together with the R graphics packages *ggplot2* [10] and analysis package *qpcR* [2]. The Marquardt-Levenberg algorithm used for comparison uses the function *nls.lm()* which is to be found in the *minpack.lm* package [11] and computational time calculations implemented the *Rprof()* function in the *R.utils* package [12]. Computational time depends heavily upon coding technique, particularly in high-level languages like R. We take the liberty of coding the zero-approach in two ways. One (*SlowZ*) uses the notoriously slow *mean()* and *lm()* functions in R for calculating mean and performing simple linear regression. The other (*FastZ*) codes these using the traditional  $\sum x_i/n$  for mean and the usual formula for simple linear regression, avoiding the computational overhead of the various checks carried out by the above R functions.

The other three approaches are Marquardt-Levenberg (*ML*) and the maximum correlation developed above, either with a quadratic estimate for the maximum (*CorQuad*) or with an iterative approach (*CorOpt*) to maximum using the *optimize()* function in R; both use the *cor()* function of R, known to be very slow.

We use as data thirteen files from the publicly available data sets distributed with the *qpcR* package (*batsch 1-5*, *boggy*, *guescinil*, *lievens1*, *reps*, *reps2*, *reps3*, *rutledge*, and *sistil*) together with ten from our own laboratory available through our online repository at Github [8].

In all there are 836 fluorescence curves. For each we have five ways of determining  $\lambda$ , given a known  $\beta$ .

We aim to address three issues.

- (i) We must confirm that the four iterative approaches all converge towards the same least-squares estimates of the parameters  $\lambda$ ,  $F_b$ , and  $F_{\max}$ .
- (ii) We must then determine the extent to which the approximate approach *CorQuad*, which involves no iteration, differs from the more precise approach of the iterative methods.
- (iii) We should compare computational times for the five algorithms.

We ran all five methods for each fluorescence curve in each of the data sets. For determining computational time using *Rprof()* we cycled through 2000 iterations of each data set. Times quoted are from an Intel core2 Duo 3.0 GHz machine.

### 4. Results and Discussion

As expected, the mean difference between the  $\lambda$  values estimated by the iteration approaches was very small; for a median difference of  $1.7 \times 10^{-7}$  cycles and for the 836 curves analysed the biggest difference was  $4.4 \times 10^{-5}$  cycles. As expected theoretically, the iterative approaches all converge on the same least-squares parameter estimates. The only differences in the algorithms are ease of understanding, ease of coding, and computational time or rapidity of convergence.

We were surprised to find very little difference between the algebraic approximation and the iterative methods. The median difference in the estimation of  $\lambda$  was  $1.4 \times 10^{-4}$  cycles, and the biggest difference in the 836 comparisons was only 0.036 cycles.

We would emphasize, however, that the data analysed here are data of the quality that researchers are happy to make publicly available. The day-to-day results from a working laboratory may well uncover data sets exhibiting more pathological behaviour. Of course, the algebraic approach could be refined by taking smaller steps than a whole cycle in calculating the correlation curve, but even with whole-cycle steps the method is more than acceptably accurate. It is trivially easy to implement on a spread-sheet.

As expected, finding a zero of a function in one dimension is computationally faster than finding the maximum of a function in three dimensions; our algorithm, with elementary coding, is faster than the Marquardt-Levenberg.

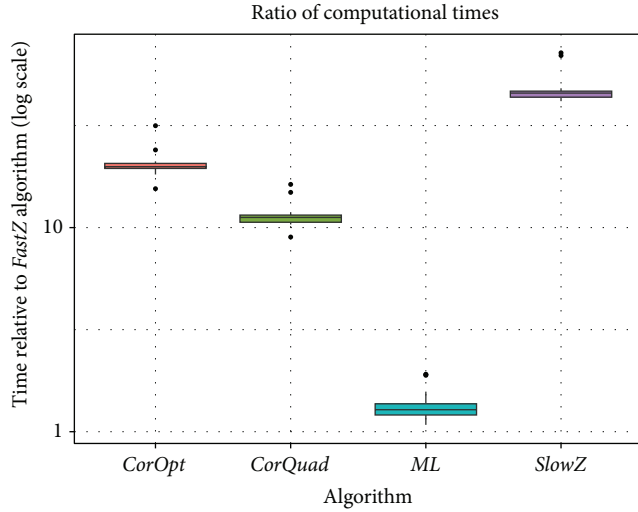


FIGURE 4: Computational times relative to that for the *FastZ* algorithm. Times were calculated individually for each of the 23 data sets.

The average times per analysed curve were 14.65 msec for *SlowZ*, 4.126 msec for *CorrOpt*, 2.448 msec for *CorrQuad*, 0.4280 msec for Marquardt-Levenberg, and 0.3344 msec for *FastZ*; the Marquardt-Levenberg took 28% more time than *FastZ*.

These times will vary, depending among other things on the shape of the experimental curves being analysed. We have therefore made comparisons individually for each of the 23 data sets; Figure 4 shows a boxplot of the computational times for the algorithms relative to that of the fastest, the *FastZ* algorithm. The interquartile ranges are 19.5 to 20.64 for *CorOpt*, 10.62 to 11.52 for *CorQuad*, 1.21 to 1.37 for *ML*, and 43.6 to 46.6 for *SlowZ*. Of course, one can generate any relative difference in computational time by modifying the tolerance at which iteration terminates. Here, Marquardt-Levenberg tolerance was such that its mean residual was greater than that for *FastZ*. That is to say, even after, on average, 28% more time, it was still further from the optimum than was *FastZ*.

## 5. Conclusions

Although the proof of the root-finding iterative method (Appendix B) is more demanding, it is much less so than is the logic underlying the Marquardt-Levenberg algorithm. The extra programming needed to calculate the vector  $\mathbf{g}$  is, for many high-level languages like R, a single line of code. The logic behind Brent's method [13] for root-finding is outlined in three pages of Press et al. [5]. We suggest that the algorithms developed here allow groups with a modicum of programming ability to analyse their qPCR data with less recourse to proprietary "black-box" software. Although our approach is faster than the Marquardt-Levenberg commonly used, we do not see speed as being its principal advantage. If speed is important, the solution should be a faster computer,

the use of compiled code, or programming in a faster language such as C++.

## Appendices

### A. Bernoulli's Method for (5)

The differential equation

$$\frac{dy}{dc} = \beta y(1 - y) \quad (\text{A.1})$$

is a Bernoulli equation. Making the substitutions

$$u = \frac{1}{y}, \quad \frac{du}{dy} = -\frac{1}{y^2} = -u^2 \quad (\text{A.2})$$

gives

$$\frac{dy}{dc} \cdot \frac{du}{dy} = -u^2 \beta \left(\frac{1}{u}\right) \left(1 - \frac{1}{u}\right), \quad (\text{A.3})$$

$$\frac{du}{dc} = -\beta(u - 1).$$

Separating variables and integrating, we have

$$\frac{1}{u-1} du = -\beta dc,$$

$$\ln(u-1) = -\beta c + k, \quad (\text{A.4})$$

$$u-1 = e^{-\beta c + k},$$

$$\frac{1}{y} = 1 + e^{-\beta c + k}.$$

Recasting the constant of integration  $k$  as  $\lambda\beta$  for some  $\lambda$  yields

$$y = \frac{1}{1 + e^{\beta(\lambda - c)}}. \quad (\text{A.5})$$

### B. Derivation of the *SlowZ* and *FastZ* Algorithms

We seek  $\lambda$  that minimizes the sum of squares of residuals  $\mathbf{r}'\mathbf{r} = \sum r_i^2$  and have asserted that the required  $\lambda$  satisfies  $\sum r_i g_i(\lambda) = 0$ , where

$$g_i(\lambda) = \frac{e^{\beta(\lambda - i)}}{(1 + e^{\beta(\lambda - i)})^2}. \quad (\text{B.1})$$

Let  $\mathbf{g}$  denote the vector  $(g_1(\lambda), g_2(\lambda), \dots, g_n(\lambda))'$ .

By construction the linear regression has resolved the observed vector of fluorescences,  $\mathbf{y}$ , into components

$$\mathbf{y} = \hat{\mathbf{y}} + \mathbf{r} = \mathbf{X}\mathbf{b} + \mathbf{r}, \quad (\text{B.2})$$

where  $\hat{\mathbf{y}} = \mathbf{X}\mathbf{b}$  lies in the solution space spanned by the columns of  $\mathbf{X}$  and  $\mathbf{r}$  is orthogonal to  $\mathbf{X}$ . The vector of parameters is  $\mathbf{b} = (F_b, F_{\max})'$ , the first column of  $\mathbf{X}$  is

$(1, 1, 1, \dots, 1)'$ , and the second is  $\mathbf{l} = (l_1(\lambda), l_2(\lambda), \dots, l_n(\lambda))'$ , where

$$l_i(\lambda) = \frac{1}{1 + e^{\beta(\lambda-i)}}. \quad (\text{B.3})$$

The vector of residuals  $\mathbf{r}$  can be expressed as  $\mathbf{r} = \mathbf{y} - \mathbf{X}\mathbf{b}$ .

Then

$$\begin{aligned} \frac{d}{d\lambda}(\mathbf{r}'\mathbf{r}) &= \left(\frac{d}{d\lambda}\mathbf{r}'\right)\mathbf{r} + \mathbf{r}'\left(\frac{d}{d\lambda}\mathbf{r}\right) \\ &= 2\mathbf{r}'\left(\frac{d}{d\lambda}\mathbf{r}\right). \end{aligned} \quad (\text{B.4})$$

So

$$\begin{aligned} \frac{d}{d\lambda}(\mathbf{r}'\mathbf{r}) = 0 &\implies \mathbf{r}'\frac{d}{d\lambda}\mathbf{r} = 0 \\ &\implies \mathbf{r}'\frac{d}{d\lambda}[\mathbf{y} - \mathbf{X}\mathbf{b}] = 0 \\ &\implies \mathbf{r}'\frac{d\mathbf{y}}{d\lambda} - \mathbf{r}'\frac{d\mathbf{X}}{d\lambda}\mathbf{b} - \mathbf{r}'\mathbf{X}\frac{d\mathbf{b}}{d\lambda} = 0. \end{aligned} \quad (\text{B.5})$$

But the first term above is zero because  $\mathbf{y}$  is the vector of observations, and in the third term,  $\mathbf{r}'\mathbf{X} = 0$  because by construction  $\mathbf{r}'$  is orthogonal to  $\mathbf{X}$ . Considering the middle term, the first column of  $\mathbf{X}$  is independent of  $\lambda$ , so

$$\mathbf{r}'\frac{d\mathbf{X}}{d\lambda}\mathbf{b} = F_{\max}\mathbf{r}'\frac{d\mathbf{l}}{d\lambda}. \quad (\text{B.6})$$

But

$$\frac{d\mathbf{l}}{d\lambda} = \beta\mathbf{g}. \quad (\text{B.7})$$

Hence

$$\frac{d}{d\lambda}(\mathbf{r}'\mathbf{r}) = 0 \implies \mathbf{r}'\mathbf{g} = 0. \quad (\text{B.8})$$

## Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

## Acknowledgments

This work was funded by Grant APP1022720 from the National Health and Medical Research Council of Australia. The authors gratefully acknowledge the contributions of two anonymous reviewers whose suggestions, we hope, clarified several issues.

## References

- [1] K. Mullis, F. Faloona, S. Scharf, R. Saiki, G. Horn, and H. Erlich, "Specific enzymatic amplification of DNA in vitro: the polymerase chain reaction," *Cold Spring Harbor Symposia on Quantitative Biology*, vol. 51, no. 1, pp. 263–273, 1986.
- [2] A. N. Spiess, "qpcR: Modelling and analysis of real-time PCR data," R package 1.3-8, 2014, <http://cran.rproject.org/web/packages/qpcR/qpcR.pdf>.
- [3] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.
- [4] K. Levenberg, "A method for the solution of certain nonlinear problems in least squares," *The Quarterly of Applied Mathematics*, vol. 2, pp. 164–168, 1944.
- [5] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1986.
- [6] S. A. Bustin, V. Benes, J. A. Garson et al., "The MIQE guidelines: minimum information for publication of quantitative real-time PCR experiments," *Clinical Chemistry*, vol. 55, no. 4, pp. 611–622, 2009.
- [7] M. E. Jones, G. C. Mayne, T. Wang, D. I. Watson, and D. J. Hussey, "A fixed-point algorithm for estimating amplification efficiency from apolymerase chain reaction dilution series," *BMC Bioinformatics*, vol. 15, article 372, 2014.
- [8] "Online repository of PCR data," <https://github.com/Michael-EmlynJones/PCRData.git>.
- [9] R Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria, 2014.
- [10] H. Wickham, *ggplot2: Elegant Graphics for Data Analysis*, Springer, New York, NY, USA, 2009.
- [11] T. V. Elzhov, K. M. Mullen, A. N. Spiess, and B. Bolker, "Package 'minpack.lm,'" 2013, <http://cran.rproject.org/web/packages/minpack.lm/minpack.lm.pdf>.
- [12] B. Bengtsson, Package 'R.utils', 2014, <http://cran.rproject.org/web/packages/R.utils/R.utils.pdf>.
- [13] R. P. Brent, *Algorithms for Minimization without Derivatives*, Prentice-Hall, Englewood Cliffs, NJ, USA, 1973.



**Hindawi**

Submit your manuscripts at  
<http://www.hindawi.com>

