U.S. Environmental Protection Agency

Revised OP (Organophosphate) Cumulative Risk Assessment

June 10, 2002

III. Appendices

B. Hazard / Relative Potency Factor

III. Appendices

B. Hazard/RPF

1. Technical Aspects of Dose-Response Analysis

Background

EPA released a Preliminary Dose-Response Assessment for OPs on July 31, 2001 (USEPA 2001b) followed by a revised dose-response assessment on December 3, 2001. Both of these analyzes were reviewed by the FIFRA SAP in September 2001 and February 2002, respectively (FIFRA SAP 2001b, 2002). The current approach was supported by the SAP (FIFRA 2002). At the February 5-8, 2002 meeting of the SAP, EPA discussed some programming errors found after the December 3, 2001 release of the Preliminary Cumulative Risk Assessment. These errors have been corrected; the contents of III.B.4 (R programming code) reflect the corrections.

Dose-Response Modeling

The goal of the statistical methods was to estimate the dose that would be expected to result in a 10% reduction in brain AChE activity, the BMD_{10}. The data for this study were in the form of dose-response studies which measured the effect of different dose rates of OP pesticides on cholinesterase activities in brain, red blood cells, and plasma. The mean and standard deviation of cholinesterase activity, and number of animals examined were available for several dosages in each data set. Females and males were analyzed separately in each study. For each chemical there were several groups of studies labeled by separate MRIDs. Within each major study, one or more studies were conducted, each with measurements taken for several durations of exposure.

It is useful to describe the approach to modeling the dose-response data in three parts:

- the shape of the dose-response curve to be used;
- how multiple data sets were modeled at the same time;
- the statistical methods used to estimate values for the model parameters.

In this analysis, the dose-response function had to accommodate three important features of the data. First, since the data came from multiple studies, perhaps carried out in different laboratories and at different times, and even sometimes reporting cholinesterase activity in different units, activity at a given dosage was expressed as a fraction of control activity. Second, it was observed that, as dose increased, cholinesterase activity in quite a few data sets approached a lower non-zero asymptote. This asymptote varied among chemicals and possibly sexes. Finally, for many of the chemicals it was apparent that there is a "shoulder" on the dose response curve, such that the dose-response curve was shallower at lower doses than at higher.

These features of the dose-response were incorporated in the dose-response model in two phases. First, a model was developed relating dose to cholinesterase activity which allowed for a horizontal asympote, and expressed activity at a given dose level as a fraction of background, or control, activity. In this document, this first model is called the "basic" model. Next, a submodel relating internal dose to administered dose, was combined with the first model to make a new model with that could have a shoulder in the low-dose region. The next subsections discuss these two models in more detail.

Basic Dose-Response Model

The basic model is described by the equation:

$$y = A\left[P_B + (1 - P_B)\,\mathrm{e}^{\frac{\log\left(\frac{1-P_B-BMR}{1-P_B}\right)}{BMD}\times Dose}\right] \qquad \textbf{Eqn. 1}$$

Here, A is the level of cholinesterase activity in the absence of exposure to organophosphate, P_B is the fraction of cholinesterase activity remaining at a very high dose of organophosphate, *BMR* is the level of inhibition at which to estimate the benchmark dose (in this study, *BMR* is always 0.10), *BMD* is the benchmark dose, and *Dose* is the dose of organophosphate pesticide, generally in units of mg/kg/day. This model is essentially the same as was described in FIFRA SAP (2001b, 2002), only reparametrized so that *BMD* appears as an explicit parameter, thus simplifying the calculations. Note that the model is undefined if $P_B + BMR \geq 1$.

Expanded Dose-Response Model

A submodel relating internal dose to administered dose was combined with the basic model to make the expanded model which allows for a shoulder in the low-dose region.

1. Biologically Inspired Model: Accounting for Potential First-Pass Metabolism

At this time, the appropriate kinetic data needed for the development of a physiologically based pharmacokinetc model (PBPK model) for all OPs are not available. EPA has developed a *biologically inspired* model based on metabolic pathways for first-pass metabolism which are *theorized* to influence the shape of the dose-response curve.

When many chemicals are administered orally, much of the absorbed chemical is carried to the liver by the portal circulation, where they may be metabolized. In the presence of saturable metabolism the dose-response curve would be expected to have a shallower slope at lower doses, and the slope would gradually increase as metabolism became saturated and more of the active chemical enters the general circulation. Although a detailed treatment of this process for each chemical is beyond the scope of this project, this basic idea was used to derive a two-parameter function of dose that relates administered dose to internal dose. The resulting function was combined with the basic exponential model giving a model that has a

low dose shoulder while retaining the dose-response shape of the basic model for larger doses.

Consider the simple two-compartment pharmacokinetic model illustrated in Figure III.B.1-1.

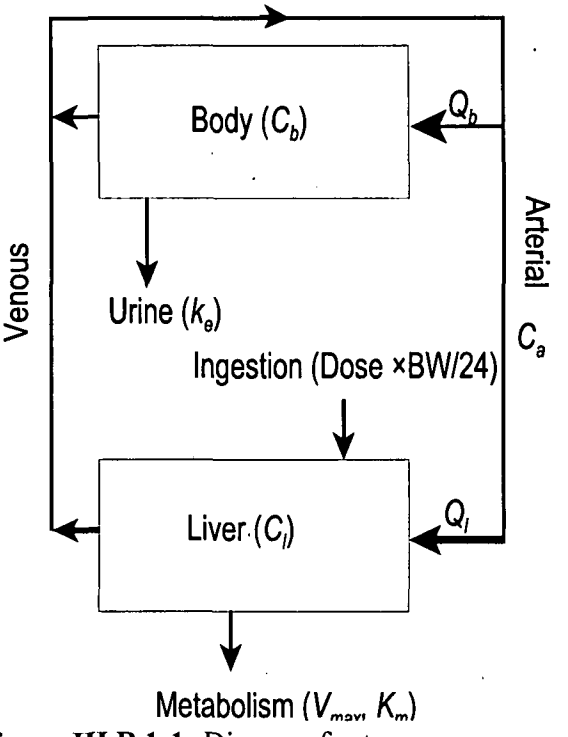


Figure III.B.1-1: Diagram for two-compartment PBPK model for the extension to the basic model

In this simple model, all the ingested chemical is taken directly to the liver, where it is metabolized. The residual unmetabolized chemical is then distributed to the rest of the body through the circulation. Intake of chemical is continuous. In this case, two differential equations and one algebraic equation describe the concentration in the liver and the rest of the body:

$$V_b \frac{dC_b}{dt} = Q_b \times (C_a - C_b) - k_e C_b$$

$$V_l \frac{dC_l}{dt} = Q_l \times (C_a - C_l) + \frac{Dose \times BW}{24} - \frac{V_{\max} C_l}{K_m + C_l}$$

$$C_a = \frac{Q_b C_b + Q_l C_l}{Q_b + Q_l}$$

Here, C_x is the concentration in compartment x, where x is *a* for arterial blood, *b* for the body other than liver, and *l* for liver. The volume of and blood flow to

compartment *x* are V_x and Q_x, where *x* is either *b* or *l*. V_{max} and K_m describe saturable metabolism of the chemical in the liver. The constant k_e is a first-order clearance term. Dose is expressed in milligrams per kilogram per day (hence the constant "24" to convert to hours), and body weight is expressed in kilograms. Thus, volumes in this parametrization are expressed in liters and concentrations in milligrams per liter.

At steady state, the derivatives are both 0: clearance just balances the dose rate. It can be shown (by solving the system of equations with derivatives set to zero) that the concentration in the body (C_b) at steady state is:

$$C_b = 0.5 * \frac{BW}{24 \times k_e} \left\{ \left(Dose - \frac{24 Q_l Q_b K_m k_e}{BW\,(Q_l k_e + Q_b k_e + Q_l Q_b)} - \frac{24 V_{\max}}{BW} \right) + \sqrt{\left(Dose - \frac{24 Q_l Q_b K_m k_e}{BW\,(Q_l k_e + Q_b k_e + Q_l Q_b)} - \frac{24 V_{\max}}{BW} \right)^2 + 4\,Dose \frac{24 Q_l Q_b K_m k_e}{BW\,(Q_l k_e + Q_b k_e + Q_l Q_b)}} \right\} \quad \textbf{Eqn (2)}$$

Here, the odd constants 0.5 and 4 arise because the solution involves finding the roots of a quadratic polynomial, and 24 arises because dose rates are usually expressed in terms of "per day", while other coefficients in the model are "per hour".

Equation (2) suggests using the function:

$$idose = 0.5 * \left\{ (Dose - S - D) + \sqrt{(Dose - S - D)^2 + 4 \times Dose \times S} \right\} \quad \textbf{Eqn (3)}$$

to describe the relationship between administered dose (*Dose*) and a scaled internal dose, where

$$S = \frac{24 Q_l Q_b K_m k_e}{BW\,(Q_l k_e + Q_b k_e + Q_l Q_b)},$$

and

$D = \frac{24 V_{\max}}{BW}$. In this parameterization of the model, V_{max}, k_e, and total blood flow (= Q_b + Q_l) should be proportional to body weight, so both *S* and *D* are independent of body weight. This is a function of two parameters (*S* and *D*), and approaches the function *idose* = *Dose* − *D* for larger doses; the slope with respect to dose when *Dose* is close to 0 is *S*/(*S* + *D*). *D* quantifies the displacement of the relationship between *Dose* and *idose* from the identity relationship, and *S* controls the shape of the relationship at low doses. In the limit as $D \to 0$ or $S \to \infty$, Equation (7) converges to *idose* = *Dose*.

In fact, it is reasonable to use Equation (3) to approximate the relationship between internal dose and administered dose in the chronic dosing setting, even in the absence of a detailed pharmacokinetic justification. The general properties of the equation capture the expected effects of first-pass metabolic clearance of an active compound: a shallow shoulder of the curve at lower doses, with a slope that

increases to a limit as the dose increases. As long as *S* and *D* are non-negative, varying these two parameters should result in a good approximation to virtually any low-dose deviation due to metabolic clearance, at least at the resolution available in bioassay dose-response data.

2. Equation for the Expanded Model.

The expanded model is just the basic model (Eqn. 1), in which *Dose* is replaced by an expression relating administered dose to internal dose. Note that, in this use of the model, the parameter *BMD* is the *internal dose* that corresponds to a *BMR* level of inhibition. Calculating the benchmark dose that corresponds to that internal dose requires setting Eqn. 3 equal to *BMD*, and solving for *Dose*.

Incorporating Differences among Datasets in the Modeling and Modeling Variability

The data for each chemical were modeled independently of all other chemicals. However, the data for any one chemical were to some extent from heterogeneous studies, grouped hierarchically. At the highest level of the hierarchy, the data could come from multiple major studies, indicated by different MRID numbers (A MRID no. is an identification code for a particular study; MRID is used in this discussion to describe the major studies). At that level, it could be expected that analytic methods could differ most distinctly, and different major studies might use different units to express cholinesterase activity. Within a major study were individual dose-response studies, often the result of multiple intermediate observations in a sub-chronic or chronic study. Although these were part of the same study, since the data collection was separated by relatively wide time intervals, there is still a reasonable expectation that details of method might vary among such data sets. Finally, within each individual dose-response study were data for both males and females. In order to combine all the data for a given chemical with a single model, all this variability needed to be incorporated in the model. This was done with a combination of allowing fixed effects to take different values in different dose-response data sets and sexes, treating a parameter as if it varied randomly across data sets, and treating some parameters as fixed for any given chemical. The following describes how each parameter was treated in the modeling.

- The parameters for the submodel relating administered dose to internal dose (*S* and *D* in Eqn 3) were given a single value for a given chemical, though they could differ between chemicals.
- The parameter governing the horizontal asymptote, P_B was allowed to differ between sexes, but otherwise to be the same value for all datasets for a given chemical.
- The background parameter, *A*, was estimated as a fixed value for each individual data set for each sex.
- The parameter *BMD* was treated as a random effect. Specifically,

$$\ln(BMD) = \mu_{lBMD} + E_{MRID} + E_{Data\ Set},$$

where μ_{lBMD} is the log of the geometric mean of the distribution of BMD among data sets, E_{MRID} and $E_{Data\ Set}$ are normally distributed random variables with mean 0 and different standard deviations that reflect variation of ln(*BMD*) among MRIDs and

among datasets within MRID, respectively. The parameter μ_{IBMD} was allowed to vary between males and females, but for each sex was constant over all data sets for a chemical. Some chemicals were represented by only one MRID, and some were represented by MRIDs with only a single data set in them. The above formula was reduced in the logical way for such chemicals. In particular, when only a single data set was available for a chemical, all the random effect terms would drop away, leaving only the log geometric mean for each sex.

- The variation among individual observations from animals of the same sex within a data set was assumed to be normal, with mean determined by the above model, and variance proportional to the mean cholinesterase activity level. An earlier version of this analysis (FIFRA SAP, 2001b) had treated the variance to be proportional to the square of the mean, and was based on analyzing the relationship between mean and variance across studies and chemicals. The current model is due to reexamining the relationship, focusing on the relationship within studies. The constant of proportionality was allowed to differ among MRIDs for a chemical, to allow for differences in units.

Estimating Parameters

It proved to be impossible to jointly estimate all the parameters for either the basic or the expanded model simultaneously. Therefore, parameters in these models were estimated using a combination of either *nlme*, a method for nonlinear mixed effects models (when there were multiple MRIDs and/or data sets for a chemical; almost all the chemicals) or *gnls*, generalized least squares, and profile likelihood. The functions *nlme* and *gnls* are from the package *nlme* for the statistical package *R* (Ihaka and Gentleman, 1996).

The *R* package *nlme* estimates parameters for nonlinear mixed effects models using the approach described in Lindstrom and Bates (1990). Davidian and Giltinan (1995, pp 164 – 174) give a good description of this model, where they refer to it as being based on "conditional first-order linearization". This approach involves approximating the nonlinear function using a Taylor expansion before carrying out maximum likelihood estimation. The implementation in *nlme* allows the fixed and random effects to be expressed as linear models of other independent variables. In this analysis, for example, *IBMD* was allowed to differ between sexes by modeling *IBMD ~ sex - 1*, where *sex* is a categorical variable in the data set that takes the values *"F"* or *"M"*. The term "*- 1* " indicates that an intercept term should not be fit for this model, so there would be an estimate of *IBMD* for each sex.

The function *gnls* in the *R* package *nlme* has a similar user interface as does the function *nlme*, but is appropriate when there are no random effects terms other than the error variance. Generalized least squares as a method is well described in Chapter 2 of Davidian and Giltinan (1995).

Parameters for the basic model were estimated first, and served as the basis for estimating parameters for the expanded model.

To fit the basic model, the values of P_{BF} and P_{BM} were set to each value on a grid of appropriate values, and the remaining parameters (background parameter for each individual data set, mean of ln(BMD) for males and females, standard deviations of ln(BMD) among MRIDs and among datasets within MRID, and parameters for the error variance) were estimated by the method appropriate to the dataset (that is, either *gnls* or *nlme*; see the discussion of these two methods, below). When all the models for that particular grid of P_B values were fit, a new grid was constructed by using the values of P_B on either side of the grid point with the largest loglikelihood as the new extremes, and repeating the process. When no BMD estimate on grid points surrounding the point with the largest loglikelihood differed from the BMD at the maximum by more than 5%, the process of iterative refining the grid stopped.

A similar method was used to estimate *S* and *D* in the expanded model. First, the values of P_{BF} and P_{BM} were fixed to their best estimates for the basic model, and were not further modified. In the expanded model, the grid being explored and refined was of values for *S* and *D*, but otherwise the process was the same as for the basic model. In the expanded model, the criterion for convergence was no difference between the maximum on the grid and neighboring points of greater than 10%.

References

Davidian, M. and Giltinan, D. M. 1995. *Nonlinear Models for Repeated Measurement Data*. Chapman and Hall. New York.

FIFRA SAP. (2001b). "Preliminary Cumulative Hazard and Dose-Response Assessment for Organophosphorus Pesticides: Determination of Relative Potency and Points of Departure for Cholinesterase Inhibition ." Report from the FIFRA Scientific Advisory Panel Meeting of September 5-6, 2001 (Report dated September 11, 2001). FIFRA Scientific Advisory Panel, Office of Science Coordination and Policy, Office of Prevention, Pesticides and Toxic Substances, U.S. Environmental Protection Agency. Washington, DC. SAP Report 2001-0X. Internet: http://www.epa.gov/scipoly/sap/2001/index.html

FIFRA SAP. (2002) "Methods Used to Conduct a Preliminary Cumulative Risk Assessment for Organophosphate Pesticides". Report from the FIFRA Scientific Advisory Panel Meeting of February 5-7, 2002. (Report dated March 19, 2002) : FIFRA Scientific Advisory Panel, Office of Science Coordination and Policy, Office of Prevention, Pesticides and Toxic Substances, U.S. Environmental Protection Agency. Washington, DC. SAP Report 20020-01. Internet: http://www.epa.gov/scipoly/sap/index.htm#june

Ihaka, R. and Gentleman, R. 1996. R: A language for data analysis and graphics. Journal of Computational and Graphical Statistics 5: 299–314.

Pinheiro, J. and Bates, D. M. 2000. *Mixed Effects Models in S and S-Plus*. Springer. Berlin.

USEPA (2001b). "Preliminary Cumulative Hazard and Dose-Response Assessment for Organophosphorus Pesticides: Determination of Relative Potency and Points of Departure for Cholinesterase Inhibition"; (released July 31, 2001) Office of Pesticide Programs, US Environmental Protection Agency, Washington DC. Internet: http://www.epa.gov/pesticides/cumulative/EPA_approach_methods.htm .

USEPA (2001c). "Preliminary Organophosphorus Pesticide Cumulative Risk Assessment". (issued in December, 2001), Office of Pesticide Programs, US Environmental Protection Agency, Washington DC. Internet: http://www.epa.gov/pesticides/cumulative/pra-op/index.htm

III. Appendices

B. Hazard/RPF

2. Dose-Response Curves

Key to Tables in Appendix III.B.2

Toxicology Profile Tables:

For each chemical, the studies reported in the toxicology profile tables correspond to the studies listed in the figures. Specifically, oral studies containing whole brain rat cholinesterase data used to determine potency are reported in the tables. In addition, dermal and inhalation toxicity studies are listed only for the chemicals with residential/ nonoccupational exposures and for the index chemical (methamidophos).

Key to Figures in Appendix III.B.2

a. **Dose-response Curve (Basic):**

Dose-response curve(s) from the basic model (low dose linear model). For chemicals with more than one study, the studies are plotted separately. Male data are red and female data are blue.

b. **Residuals from Basic Model:**

Plot of residuals from the basic model. Dotted red line represents 10% brain cholinesterase inhibition.

c. **Profile Likelihood for P_B:**

Profile likelihood plot for P_B (i.e., horizontal asymptote). The x-axis gives the ranges of P_B tried for female rat cholinesterase data (P_{BF}). The y-axis gives the ranges of P_B tried for male rat cholinesterase data (P_{BM}). As color moves from red to orange to yellow to very bright yellow, the likelihood values increase to a peak. The peak is marked by an X. Open circles are points that are not significantly different (P-value > 0.05) from the peak.

d.	**Profile Likelihood for *D* and *S*:**	Profile likelihood plot for *D* (i.e., horizontal displacement along the x-axis of the dose-response curve) and *S* (i.e., shape). As color moves from red to orange to yellow to very bright yellow, the likelihood values increase to a peak The peak is marked with an X. Closed circles are points that are not significantly different (P-value > 0.05) from the peak. Plot is listed only for those OPs where the expanded model fit the cholinesterase data significantly better than the basic.
e.	**Dose-response Curve (Expanded):**	Dose-response curve(s) from the expanded model (low dose flat model). For chemicals with more than one study, the studies are plotted separately. Male data are red and female data are blue. Plot(s) is/are listed only for those OPs where the expanded model fit the cholinesterase data significantly better than the basic.
f.	**Residuals from Model w/Low Dose Curvature:**	Plot of residuals from the expanded model. Dotted red line represents 10% brain cholinesterase inhibition. Plot is listed only for those OPs where the expanded model fit the cholinesterase data significantly better than the basic.

Table III.B.2-1. Acephate: Toxicology Profile Table

Acephate						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
40504819	82-1 (870.3100)	Subchronic Oral Toxicity–Rat (Special ChE inhibition study)	006680 012544 14258	0/0, 0.15/0.12, 0.36/0.28, 0.76/0.58, 11.48/8.90 mg/kg/day (females/males)	Nonguideline	Rat/ Sprague Dawley
00084017	83-5 (870.4300)	Combined Chronic Oral Toxicity/ Carcinogenicity–Rat	004951 012544	0/0, 0.3/0.2,3.1/ 2.4,47.2/38.2 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
45134301	82-2 (870-3200)	21-Day Dermal Toxicity–Rat	14210 41528	0, 20, 30, 40, 50 mg/kg/day	Nonguideline	Rat/ Sprague Dawley
44541101	82-2 (870.3200)	21-Day Dermal Toxicity–Rat	13396	0, 12, 60, 300 mg/kg/day	Guideline	Rat/ Sprague Dawley
45134302	82-4 (870.3465)	Subchronic Inhalation Toxicity–Rat	14223 41528	0, 0.001064, 0.003123, 0.005550 mg/L	Nonguideline	Rat/ Sprague Dawley
40504818	82-4 (870.3465)	4-Week Inhalation Toxicity–Rat	12544	0 (air), 1.05, 10.8, 93.6 mg/m^3	Guideline	Rat/ Fischer
40645903	82-4 (870.3465)	4-Week Inhalation Toxicity–Rat	12544	0 (air), 0.187, 0.507 mg/m^3	Guideline	Rat/ Fischer

Figure III.B.2-1. Acephate: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot For P_B

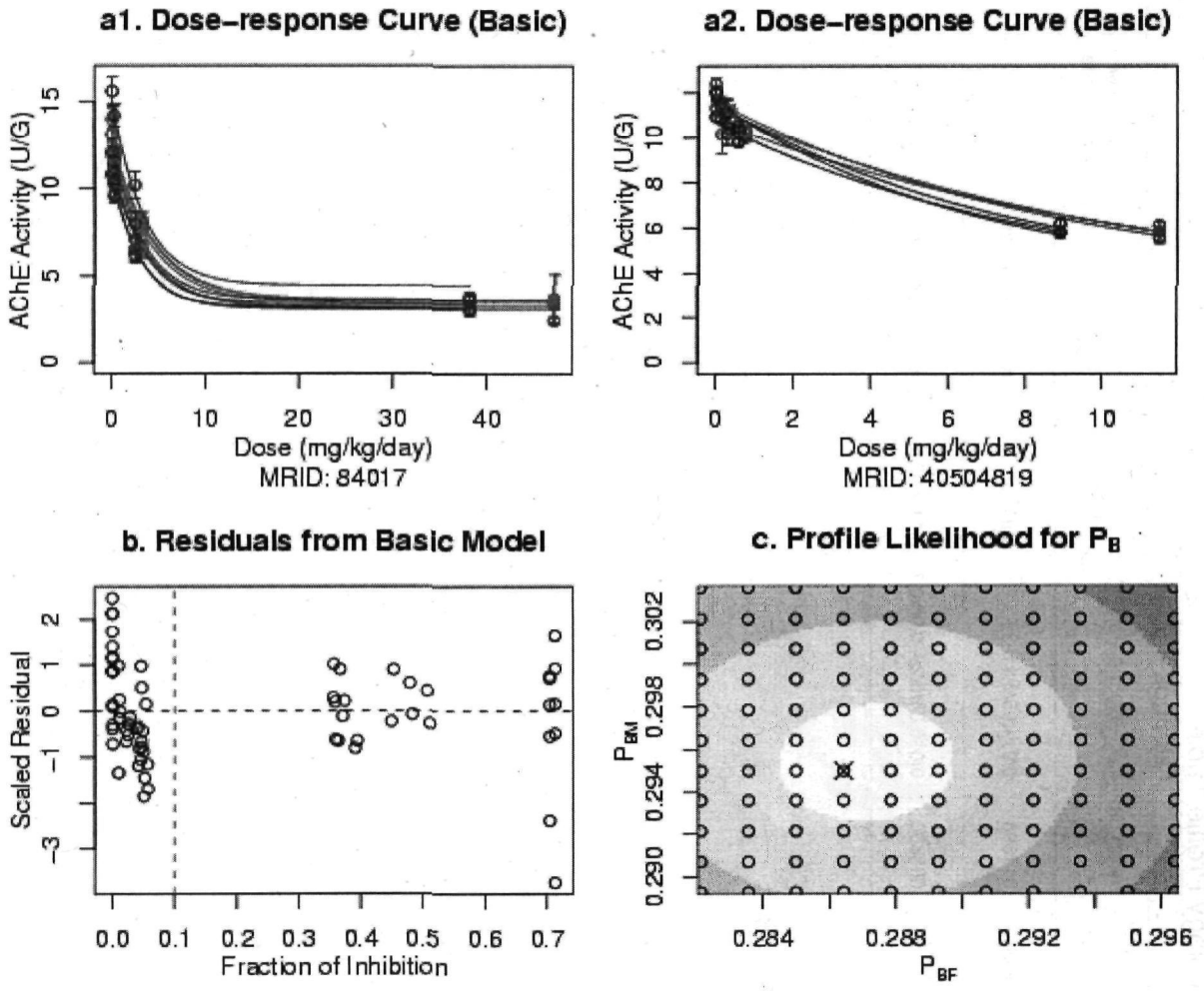


Table III.B.2-2. Azinphos-methyl: Toxicology Profile Table

Azinphos-methyl						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
43826601	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	011898	0/0, 1.05/0.91, 3.23/2.81, 6.99/7.87 mg/kg/day (females/males)	Guideline	Rat/ Fischer
41119901	83-5 (870.4300)	Combined Chronic Oral Toxicity/ Carcinogenicity–Rat	008300	0/0, 0.31/0.25, 0.96/0.75, 3.11/2.33 mg/kg/day (females/males)	Guideline	Rat/ Wistar

Figure III.B.2-2. Azinphos-methyl: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots For P_B, D, and S

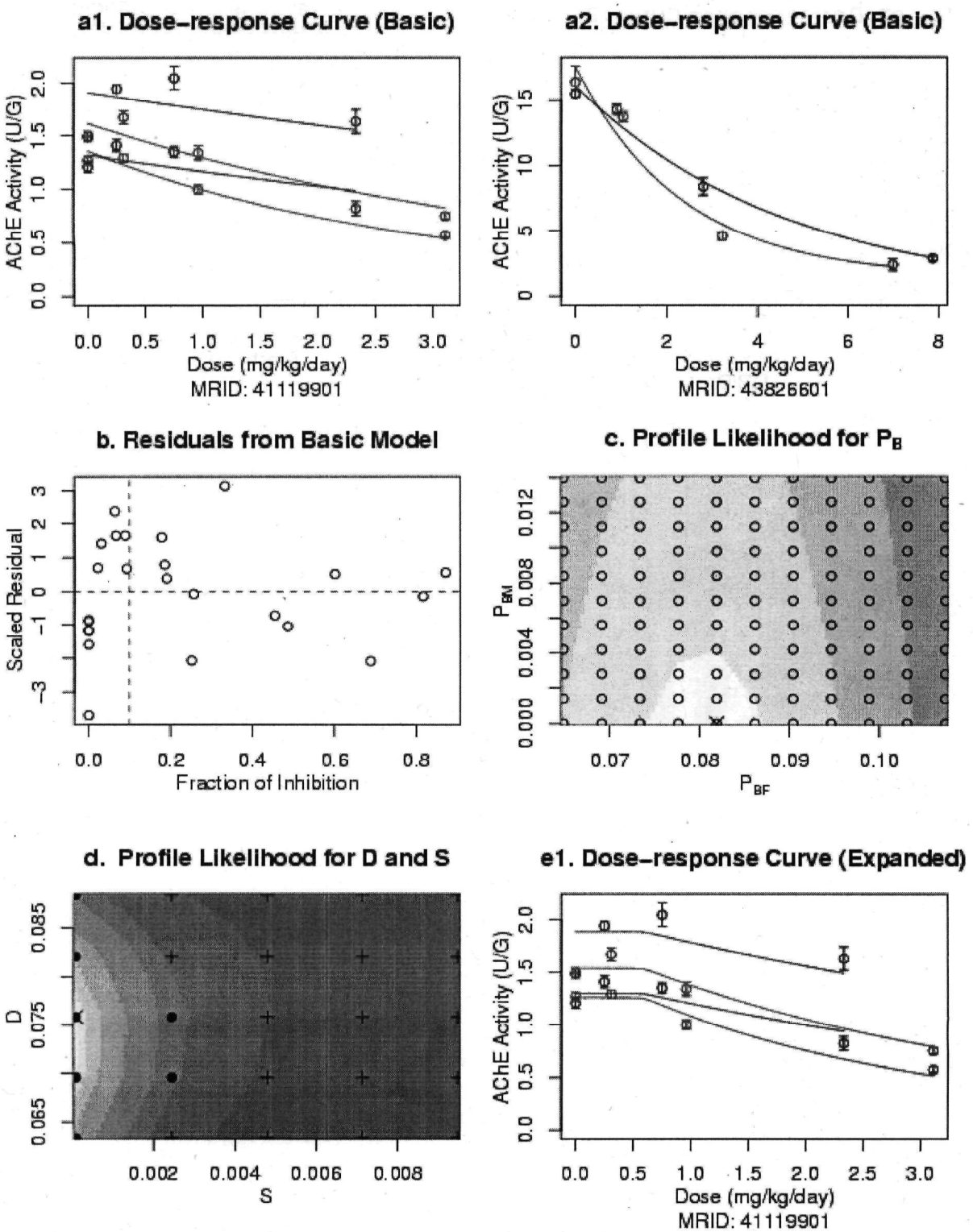
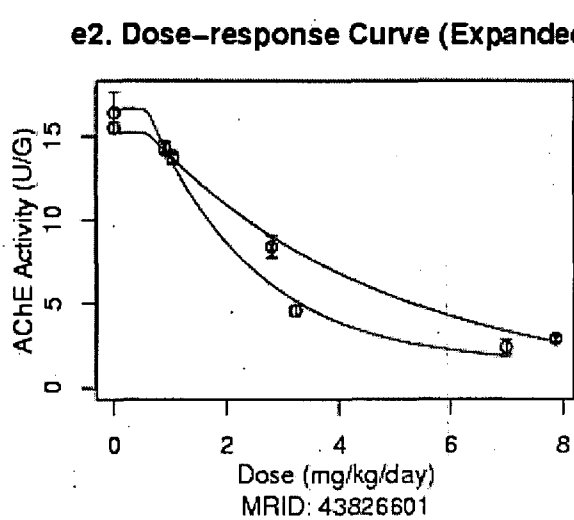


Figure III.B.2-2. Azinphos-methyl con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots For P_B, D, and S



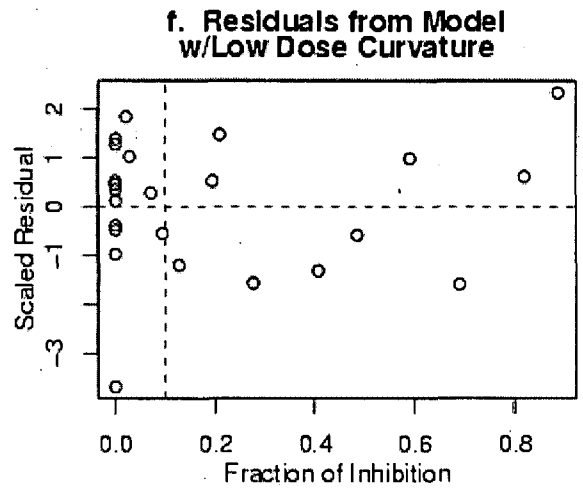


Table III.B.2-3. Bensulide: Toxicology Profile Table

Bensulide						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
43919601	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	12289	0/0, 5/5, 15/15, 45/46, or 100/110 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
44161101	83-5 (870.4300)	Combined Chronic Oral Toxicity/ *Carcinogenicity*–Rat	12289	0/0, 1/1, 15.30/15.10, 61.30/60.10 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
44801101 44809401	82-2 (870.3200)	21-Day Dermal Toxicity–Rat	013532	0, 30, 50, 500 mg/kg/day	Nonguideline	Rat/CD

Figure III.B.2-3. Bensulide: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

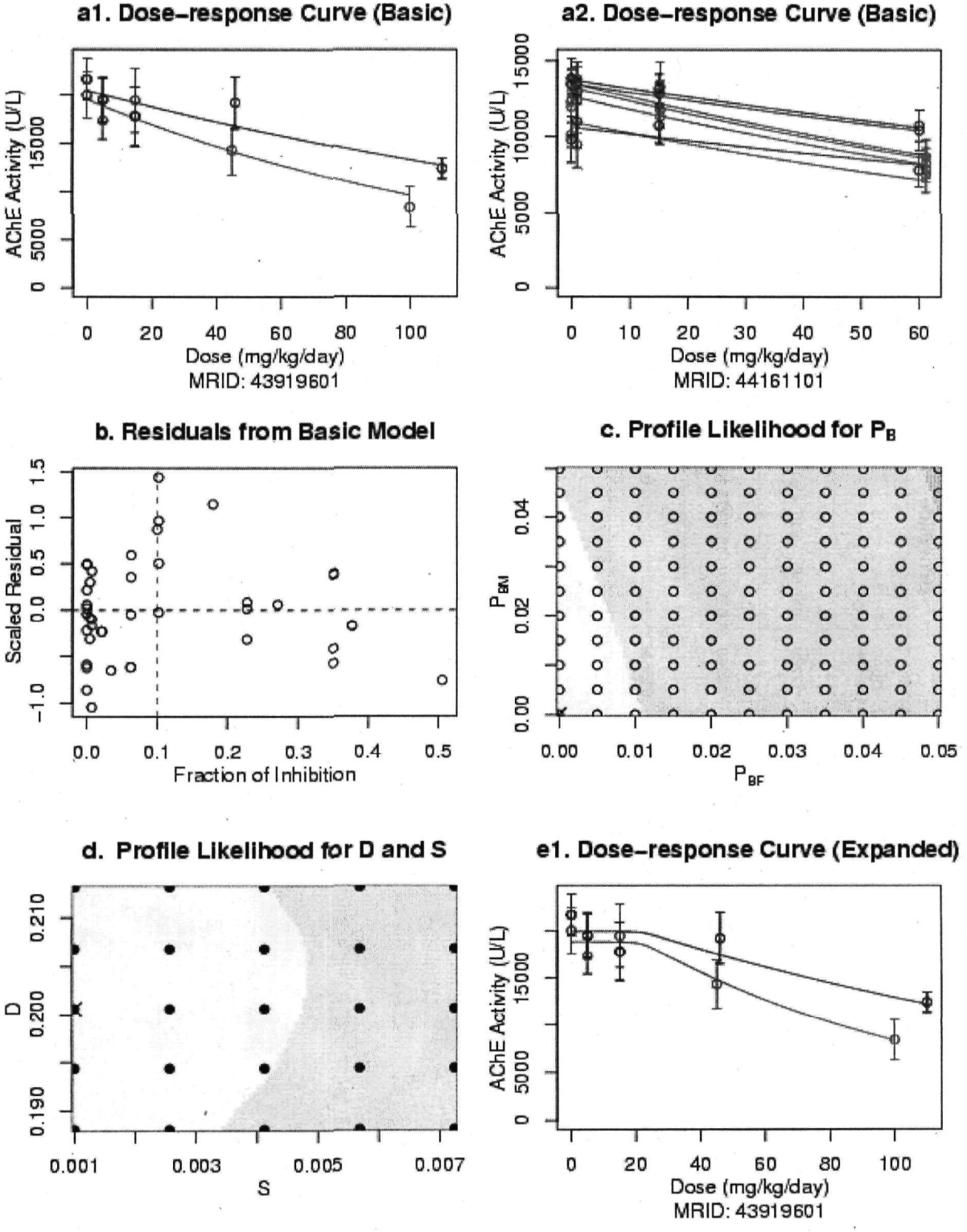


Figure III.B.2-3. Bensulide Con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

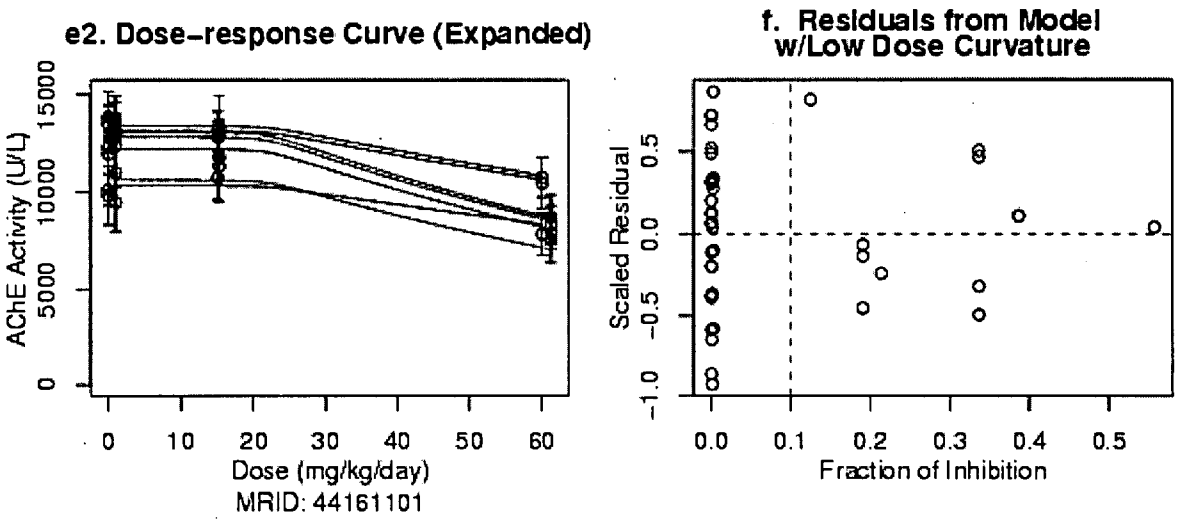


Table III.B.2-4. Chlorethoxyfos: Toxicology Profile Table

Chlorethoxyfos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
41290632	82-1 (870.3100)	Six Week Oral Toxicity - Rat	008330	0/0, 0.014/0.009, 0.132/0.091, 0.66/0.477, 1.3/0.958 mg/kg/day (females/males)	Supplemental	Rat/Crl:CD®BR
42559215	82-1 (870.3100)	Subchronic Oral Toxicity - Rat	NA	0, 0.008, 0.080, 0.635, 1.23, 1.63 mg/kg/day (females only)	Guideline	Rat/Crl:CD®BR
41736837	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity Study - Rat	NA	0/0, 0.005/0.004, 0.042/ 0.031, 0.208/ 0.154, 0.416/ 0.311 mg/kg/day (females/males)	Guideline	Rat/Crl:CD®BR

NA=Not available

Figure III.B.2-4. Chlorethoxyfos: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

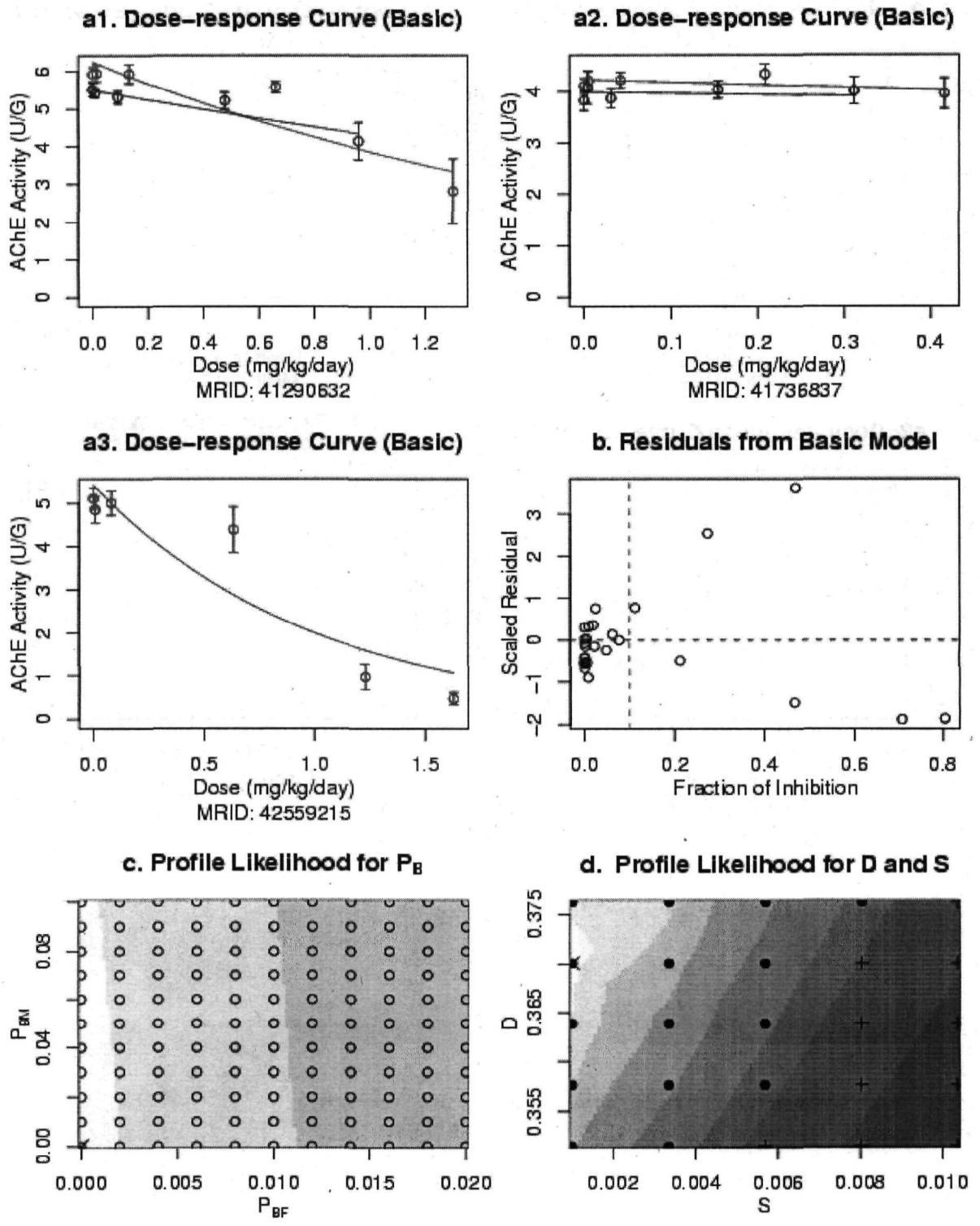


Figure III.B.2-4. Chlorethoxyfos con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

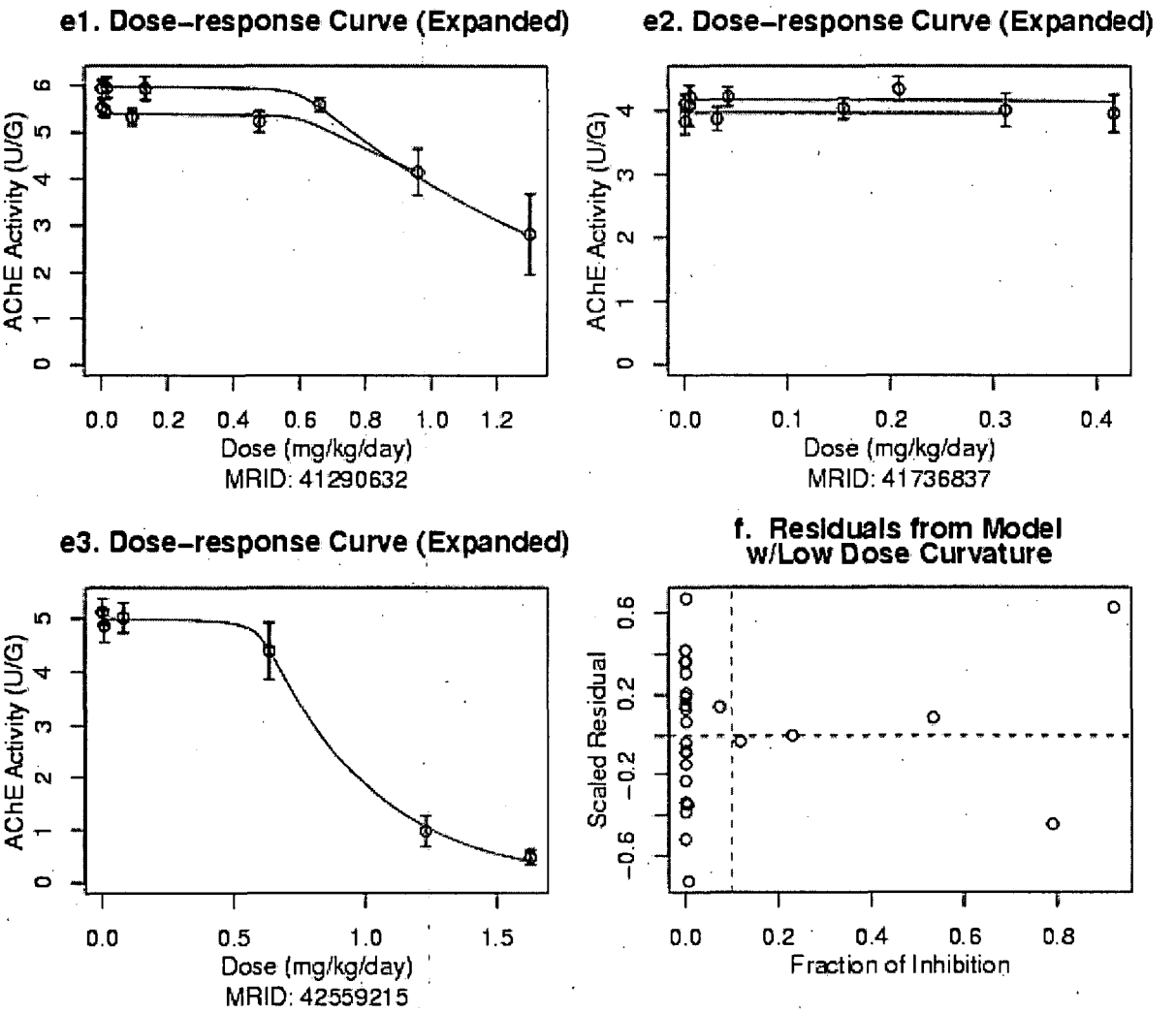


Table III.B.2-5. Chlorpyrifos: Toxicology Profile Table

Chlorpyrifos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
40952801	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	007102	0, 0.10, 1.00, 5.00, 15.00 mg/kg/day	Guideline	Rat/ Fischer
42172802	83-5 (870.4300)	Combined Chronic Oral Toxicity/ Carcinogenicity– Rat	009733 010605 013240	0/0, 0.01/0.01, 0.37/0.33, 7.61/6.77 mg/kg/day (females/males)	Guideline	Rat/ Fischer
40952802	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity– Rat	007107 013240	0, 0.05, 0.10, 1, 10 mg/kg/day	Guideline	Rat/ Fischer

Figure III.B.2-5 Chlorpyrifos: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots For P_B, D, and S

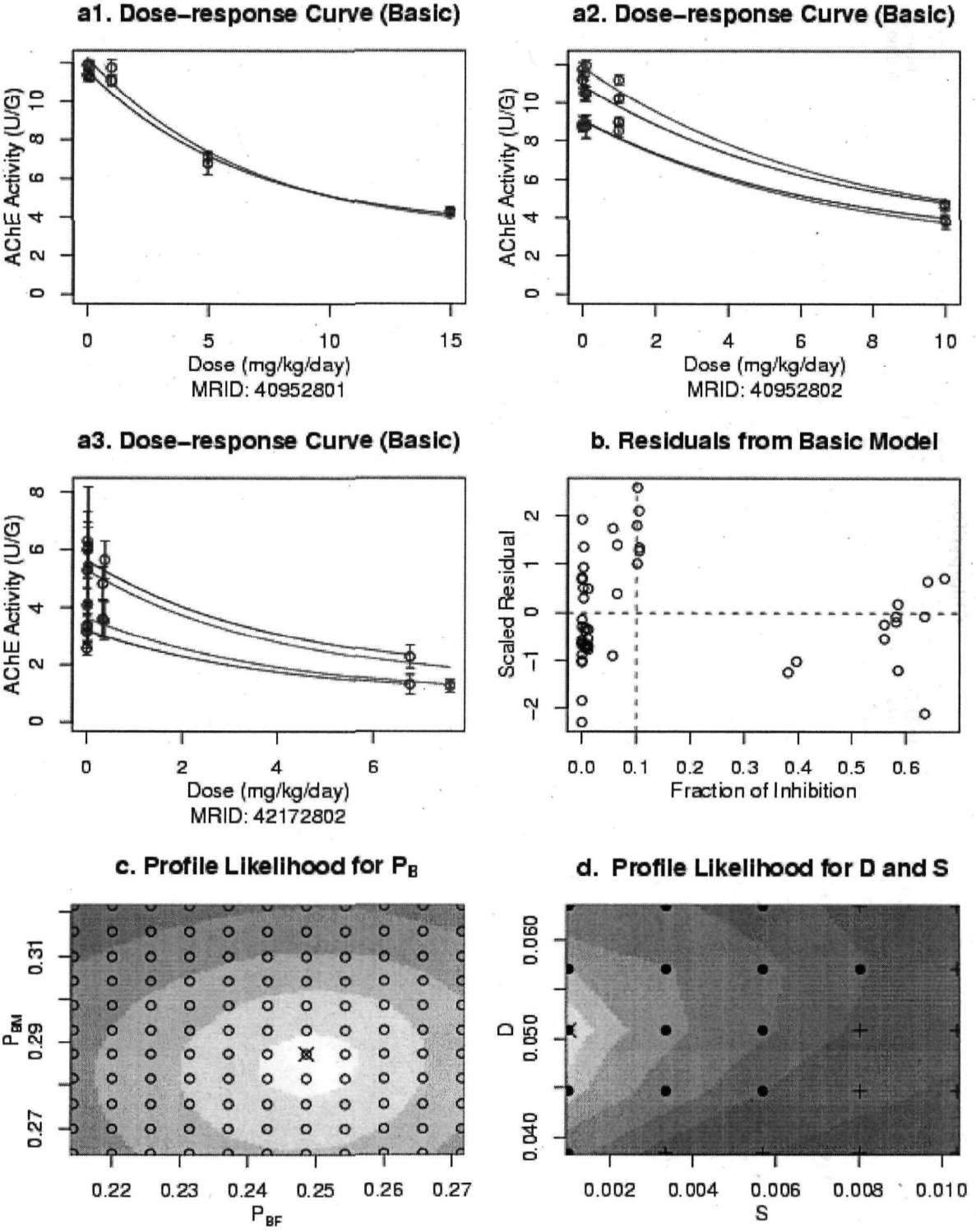


Figure III.B.2-5 Chlorpyrifos con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots For P_B, D, and S

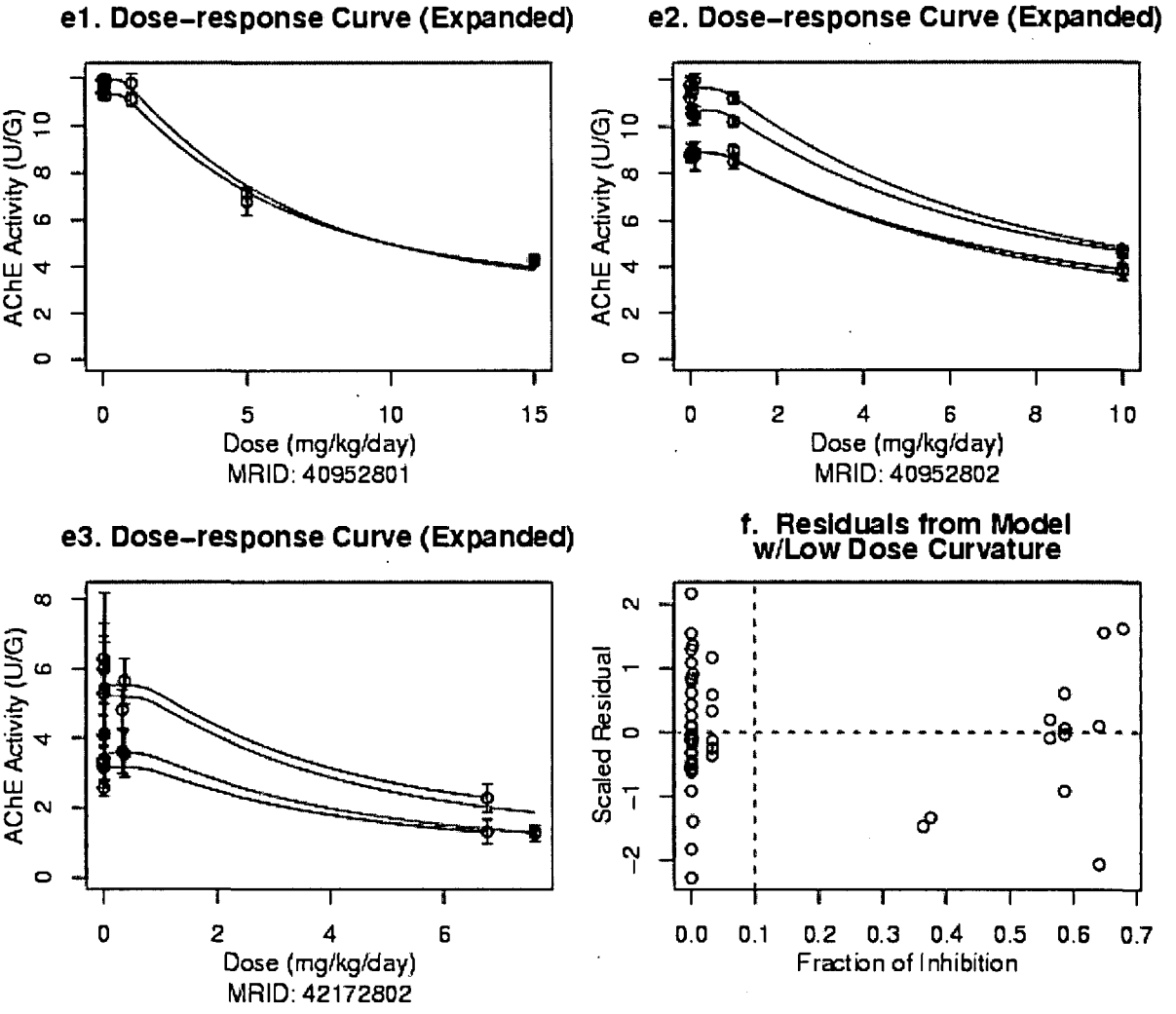


Table III.B.2-6. Chlorpyrifos-methyl: Toxicology Profile Table

Chlorpyrifos-methyl						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
42269001	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	009560	0, 0.05, 0.1, 1, 50 mg/kg/day	Guideline	Rat/Fischer
44906902	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	014122	0, 0.1, 1, 10, 250 mg/kg/day	Guideline	Rat/Fischer

Figure III.B.2-6. Chlorpyrifos-methyl: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot For P_B

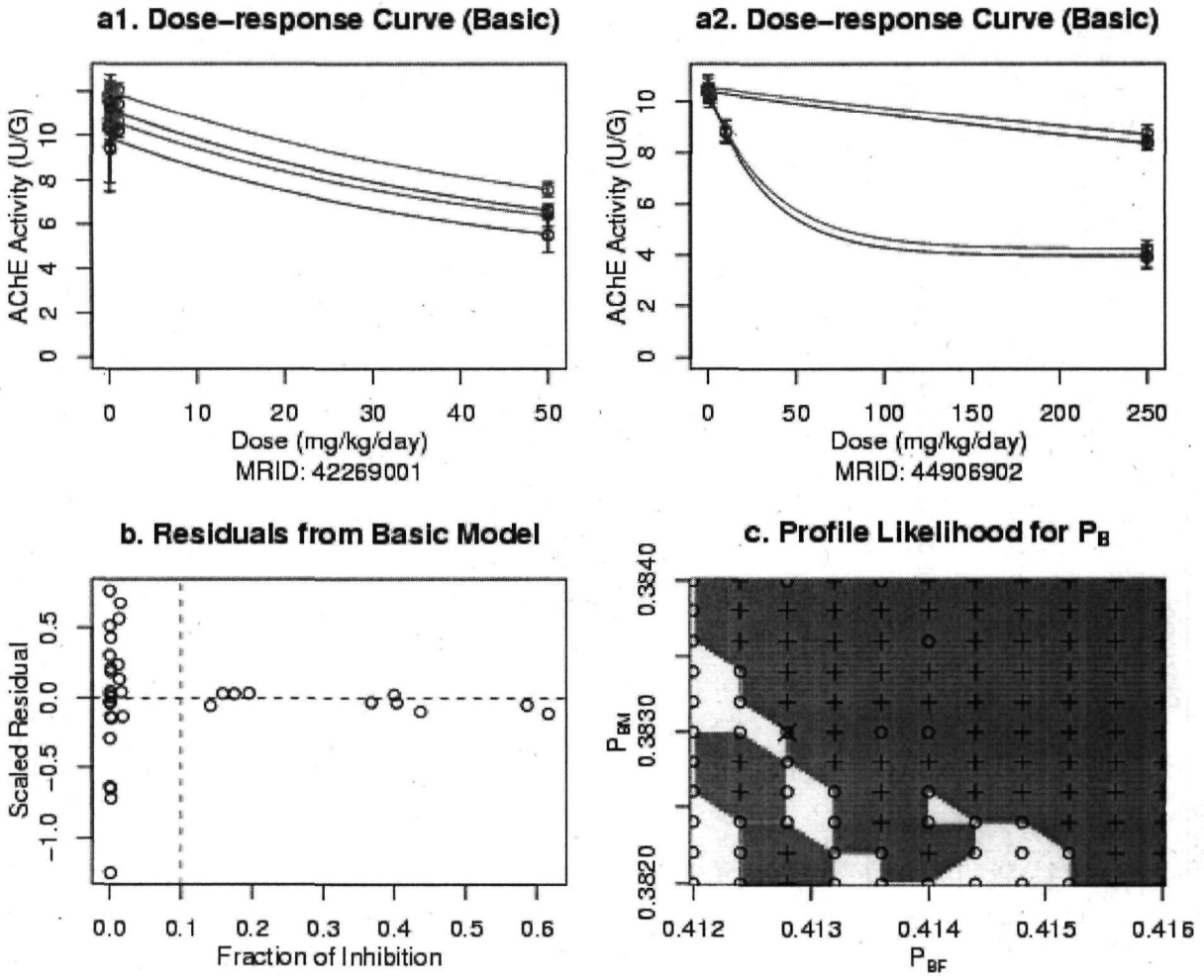


Table III.B.2-7. Diazinon: Toxicology Profile Table

Diazinon						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
40815003	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	007041 007553 012219	0/0, 0.04/0.03, 0.40/0.30, 19/15, 212/168 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
41942002	83-1 (870.4100)	Chronic Oral Toxicity–Rat	010331 012219	0, 0.005/0.004, 0.07/0.06, 6/5, or 12/10 mg/kg/day (males/females)	Guideline	Rat/ Sprague Dawley

Figure III.B.2-7. Diazinon: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

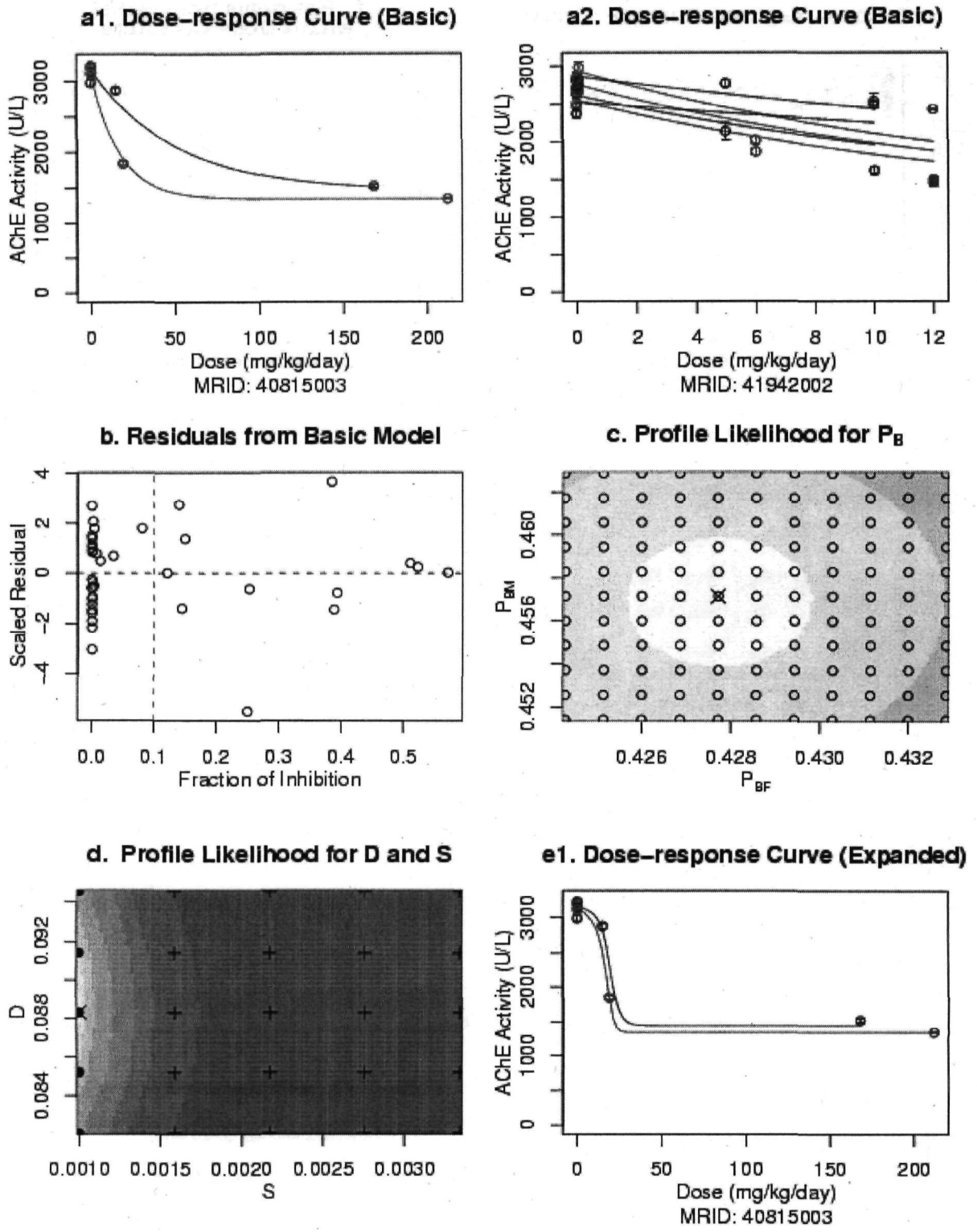


Figure III.B.2-7. Diazinon con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

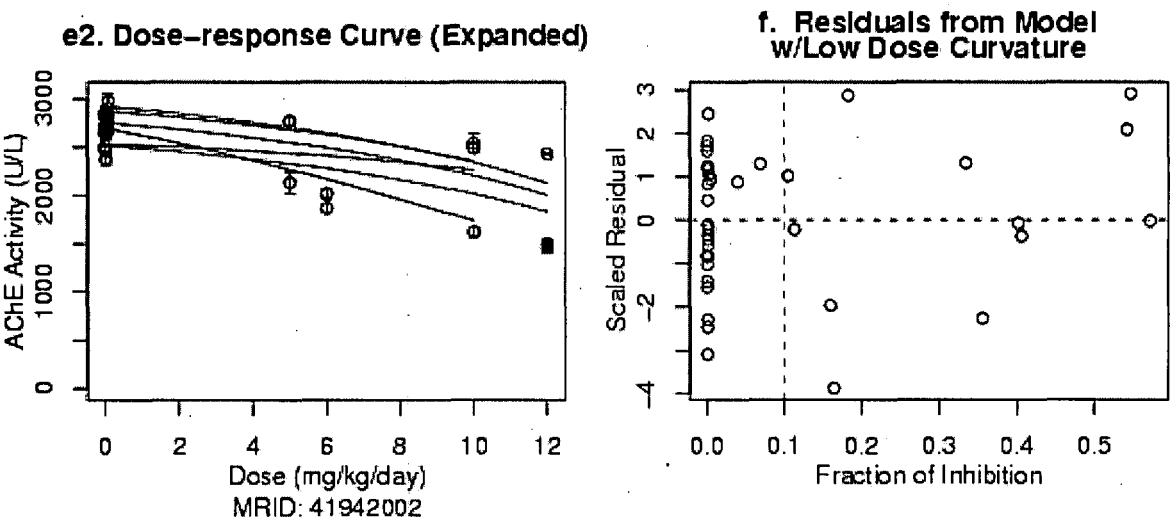


Table III.B.2-8. Dichlorvos: Toxicology Profile Table

Dichlorvos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
41004701	82-1 (870.3100)	Subchronic Oral (Gavage) Toxicity–Rat	007448	0, 0.1, 1.5, 15 mg/kg/day (gavage)	Guideline	Rat/ Sprague Dawley
00057695 00632569	83-5 (870.4300)	Combined Chronic Inhalation Toxicity/Carcinogenicity–Rat	001466 006860	0, 0.05, 0.5, 5 mg/m^3	Supplemental	Rat/ Carworth Farm E (CFE)

Figure III.B.2-8. Dichlorvos: Dose-response Curve Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

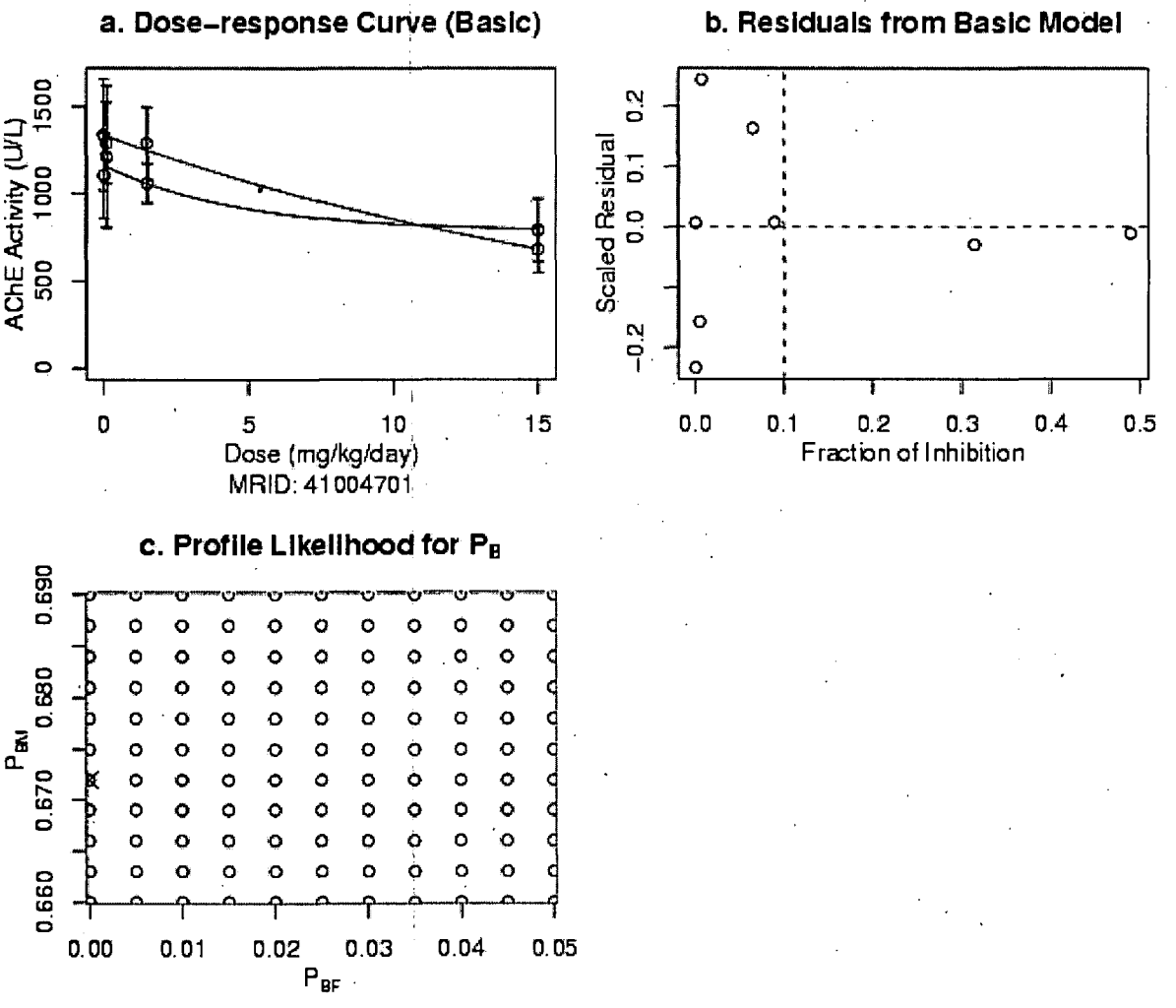


Table III.B.2-9. Dicrotophos: Toxicology Profile Table

Dicrotophos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
44527802	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	012994	0/, 0.03/0.02, 0.32/0.25,1.74/1.42 mg/kg/day (females/males)	Guideline	Rat/Sprague Dawley
43980201	82-7 (870.6200)	Subchronic Neurotoxicity –Rat	013048	0/0, 0.04/0.04, 0.45/0.39, 2.38/2.03 mg/kg/day (females/males)	Guideline	Rat/Sprague Dawley

Figure III.B.2-9. Dicrotophos: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

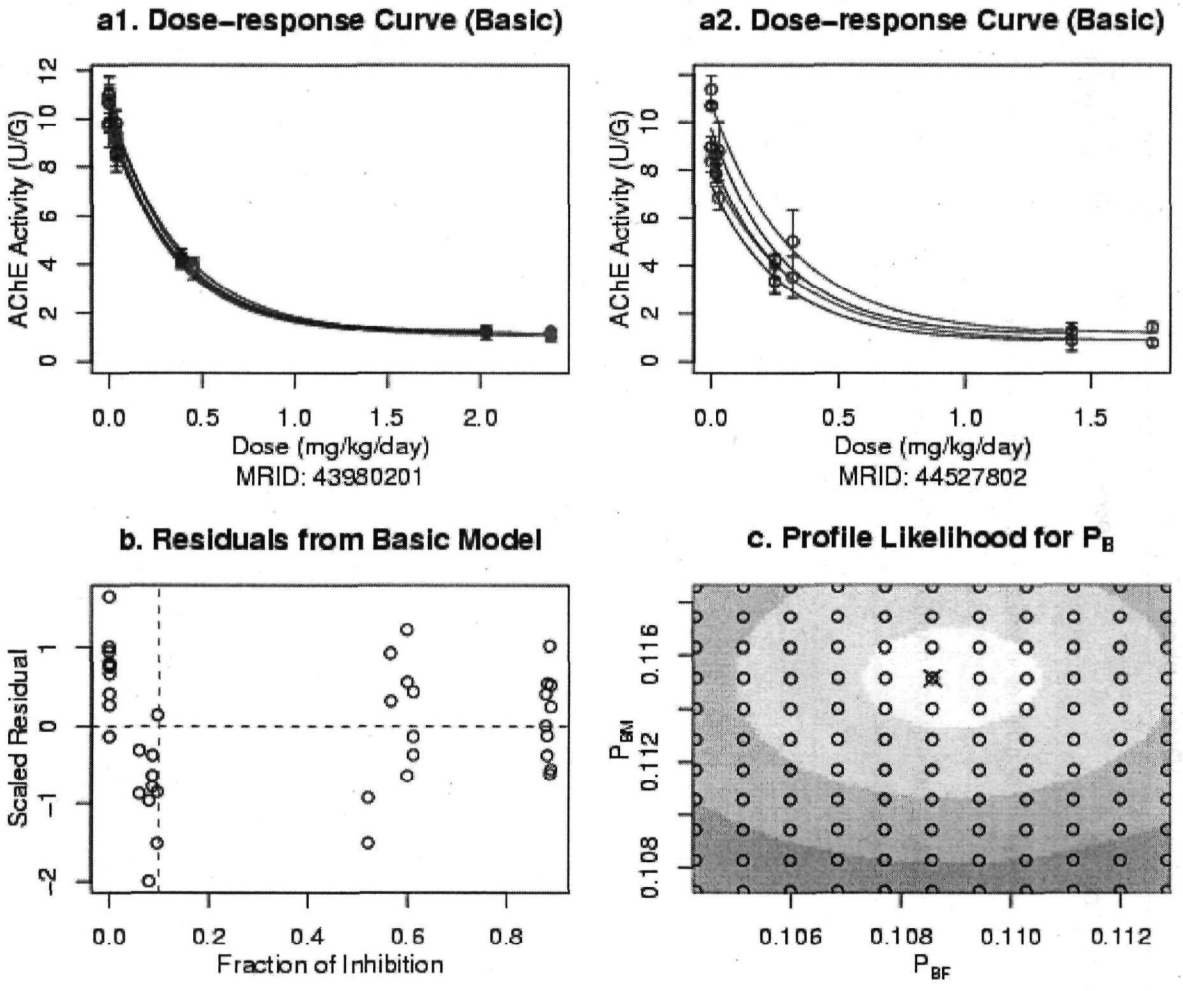


Table III.B.2-10. Dimethoate: Toxicology Profile Table

Dimethoate						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
164177	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	006398 008457	0/0, 0.06/0.04, 0.30/0.23, 1.48/1.16, 6.29/4.82 mg/kg/day (females/males)	Guideline	Rat/ Wistar

Figure III.B.2-10. Dimethoate: Dose-response Curve Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

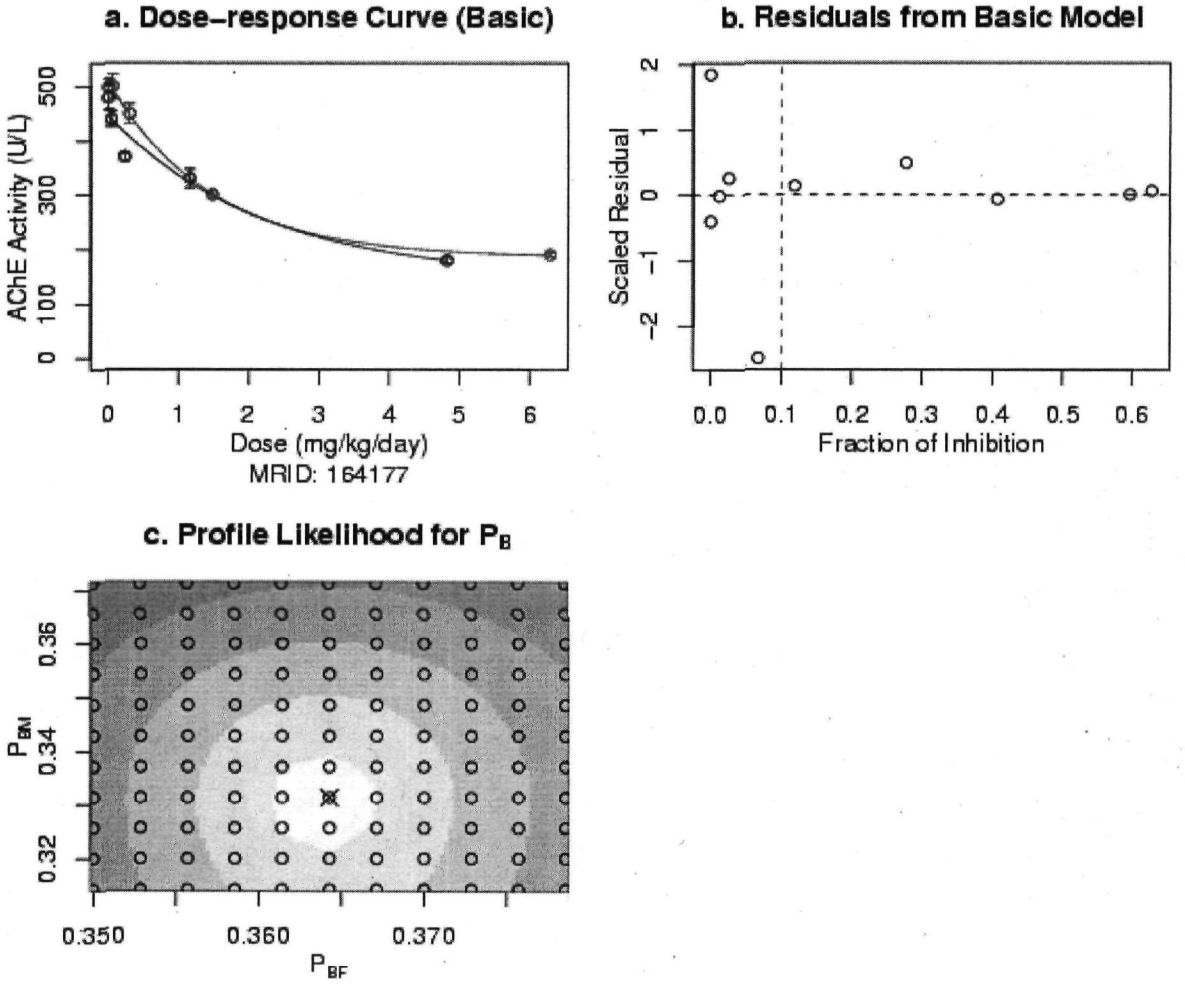


Table III.B.2-11. Disulfoton: Toxicology Profile Table

Disulfoton						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
42977401	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	011456	0/0, 0.07/0.06, 0.31/0.27, 1.30/1.08 mg/kg/day (females/males)	Guideline	Rat/ Fischer
43058401	Non-guideline study	Special 6-month Cholinesterase–Rat	011249	0/0, 0.02/0.02, 0.03/0.03, 0.07/0.06 mg/kg/day (females/males)	Nonguideline	Rat/ Fischer
146873/ 41850002	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	005029	0/0, 0.08/0.06, 0.26/0.22, 1.25/0.92 mg/kg/day (females/males)	Guideline	Rat/ Fischer
44758404	82-1 (870.3100)	28-Day Dietary Study - Rat	NA	Prep 1: 0/0, 0.18/0.17, 1.11/1.04 mg/kg/day Prep 2: 0/0, 0.16/0.14, 1.29/1.16 mg/kg/day	NA	Rat/Fischer
00162338	82-2 (870.3200)	21-Day Dermal Toxicity–Rabbit	005556	0, 0.4, 1.6, 6.5 mg/kg/day	Guideline	Rabbit/ New Zealand
45239601	82-2 (870.3200)	21-Day Dermal Toxicity–Rabbit	014448	0, 0.8, 1, 3 mg/kg/day	Guideline	Rabbit/ New Zealand
41224301	82-4 (870.3465)	Subchronic Inhalation Toxicity–Rat	011242	Air and PEG-400:50% ethanol vehicle controls, 0.016/0.018, 0.16/0.16, 1.4/1.4 mg/m^3 (females/males)	Guideline	Rat/ Fischer

Figure III.B.2-11. Disulfoton: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

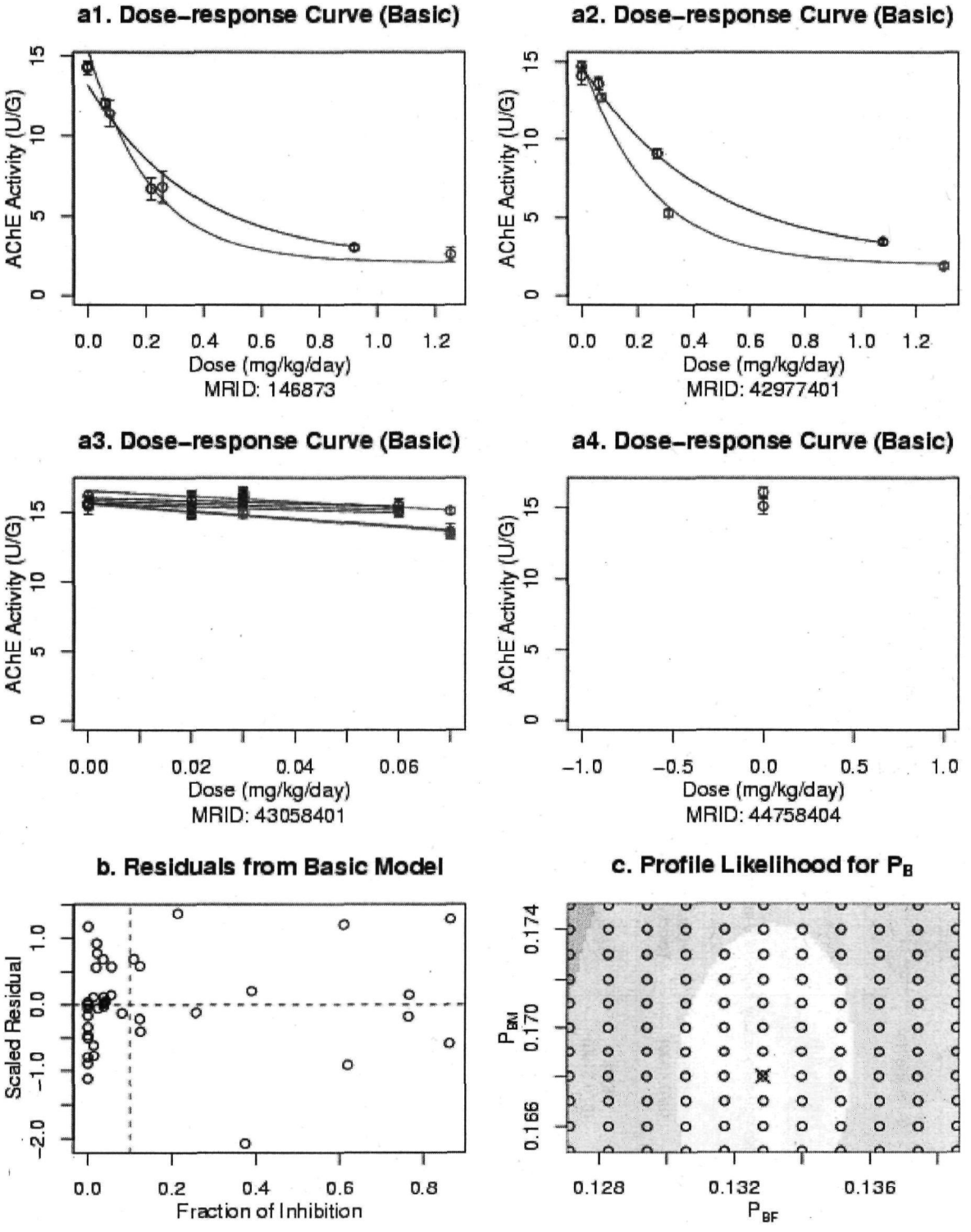


Figure III.1.1-11. Disulfoton con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

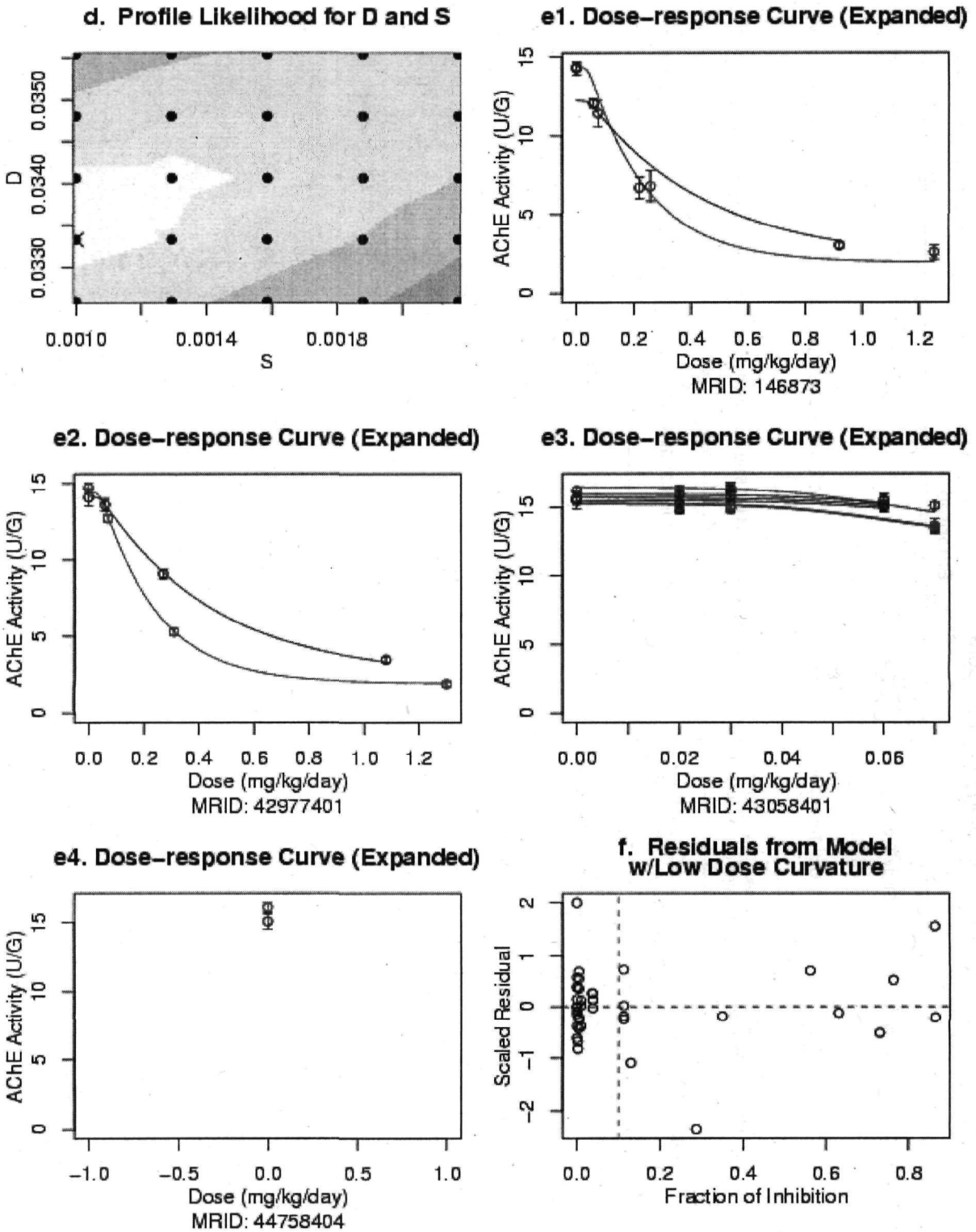


Table III.B.2-12. Ethoprop: Toxicology Profile Table

Ethoprop						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
75239	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	001789 001795 002775	0, 0.015, 0.05, 5 mg/kg/day	Supplementary	Rat/Charles River
40291801	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	012589	0/0, 0.052/0.041, 0.51/0.4, 5.12/4.19 mg/kg/day (females/males)	Supplementary	Rat/Fischer
138636	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	006006 005741 012589	0/0, 13.1/10;28/ 21,59.3/44.8 mg/kg/day (females/males)	Supplementary	Rat/Fischer
42530201	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	012589 010775	0/0, 0.06/0.04, 3.27/2.62, 23.98/18.55 mg/kg/day (females/males)	Guideline	Rat/Crl:CD

Figure III.B.2-12. Ethoprop: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

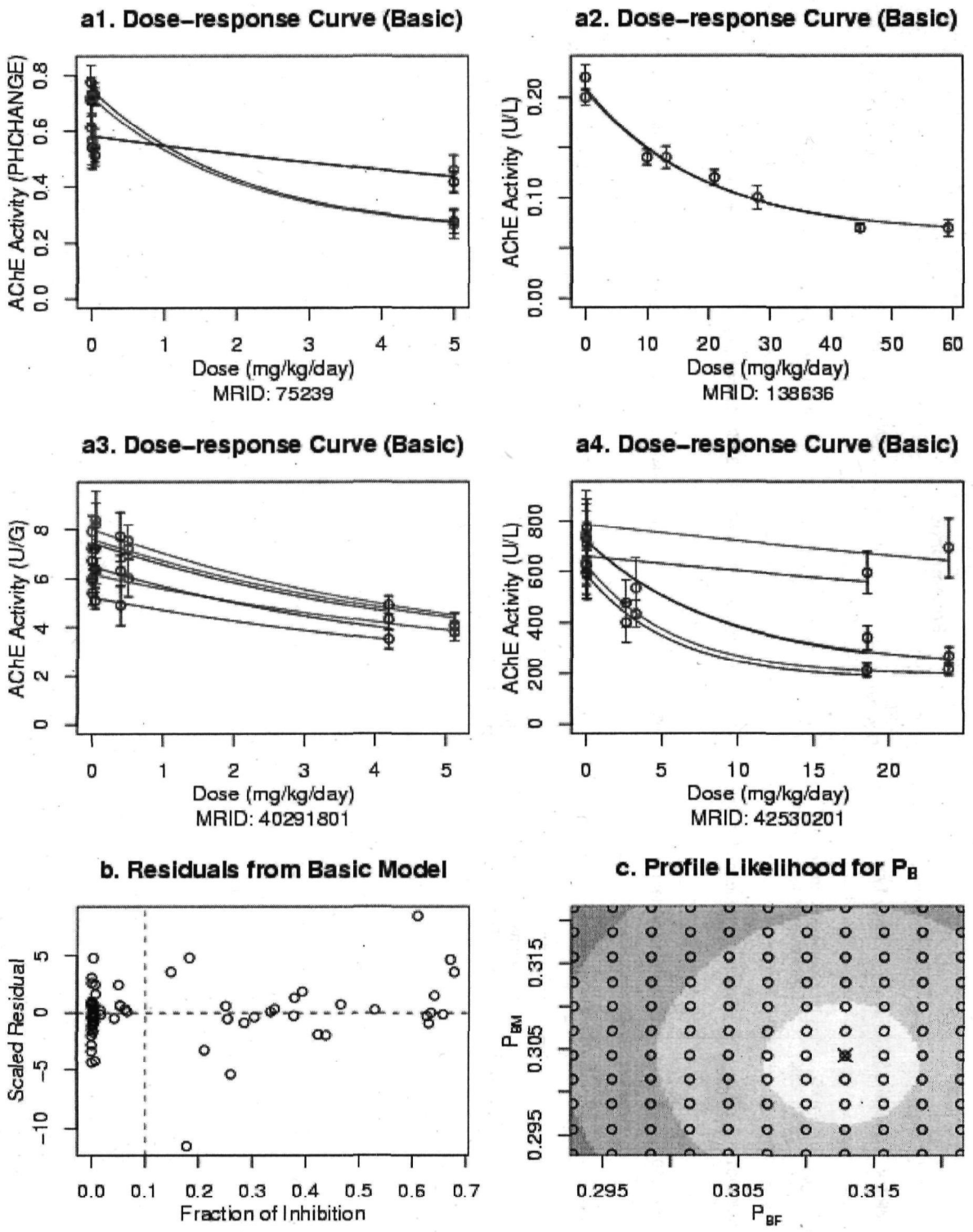


Table III.B.2-13. Fenamiphos: Toxicology Profile Table

Fenamiphos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
00161361	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	003331 003606 005722	0/0, 0.12/0.10, 0.60/0.46, 3.36/2.45 mg/kg/day (females/males)	Guideline	Rat/ Fischer
44051401	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	012019	0/0, 0.08/0.06, 0.80/0.61, 3.98/3.13 mg/kg/day (females/males)	Guideline	Rat/ Wistar
00161360	82-1 (870.3100)	90-Day Cholinesterase Study–Rat	003606	0, 0.018, 0.03, or 0.05 mg/kg/day	Minimum	Rat/Fischer
00154497	82-2 (870.3200)	21-Day Dermal Toxicity–Rabbit	004531 005722	0, 0.5, 2.5, 10 mg/kg/day	Guideline	Rabbit/ New Zealand White
40774809	82-4 (870.3465)	21-Day Inhalation Toxicity–Rat (nose only)	004531 010301 011035	0, 0.03, 0.25, 3.5 µg/L	Guideline	Rat/ Wistar

Figure III.B.2-13. Fenamiphos: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

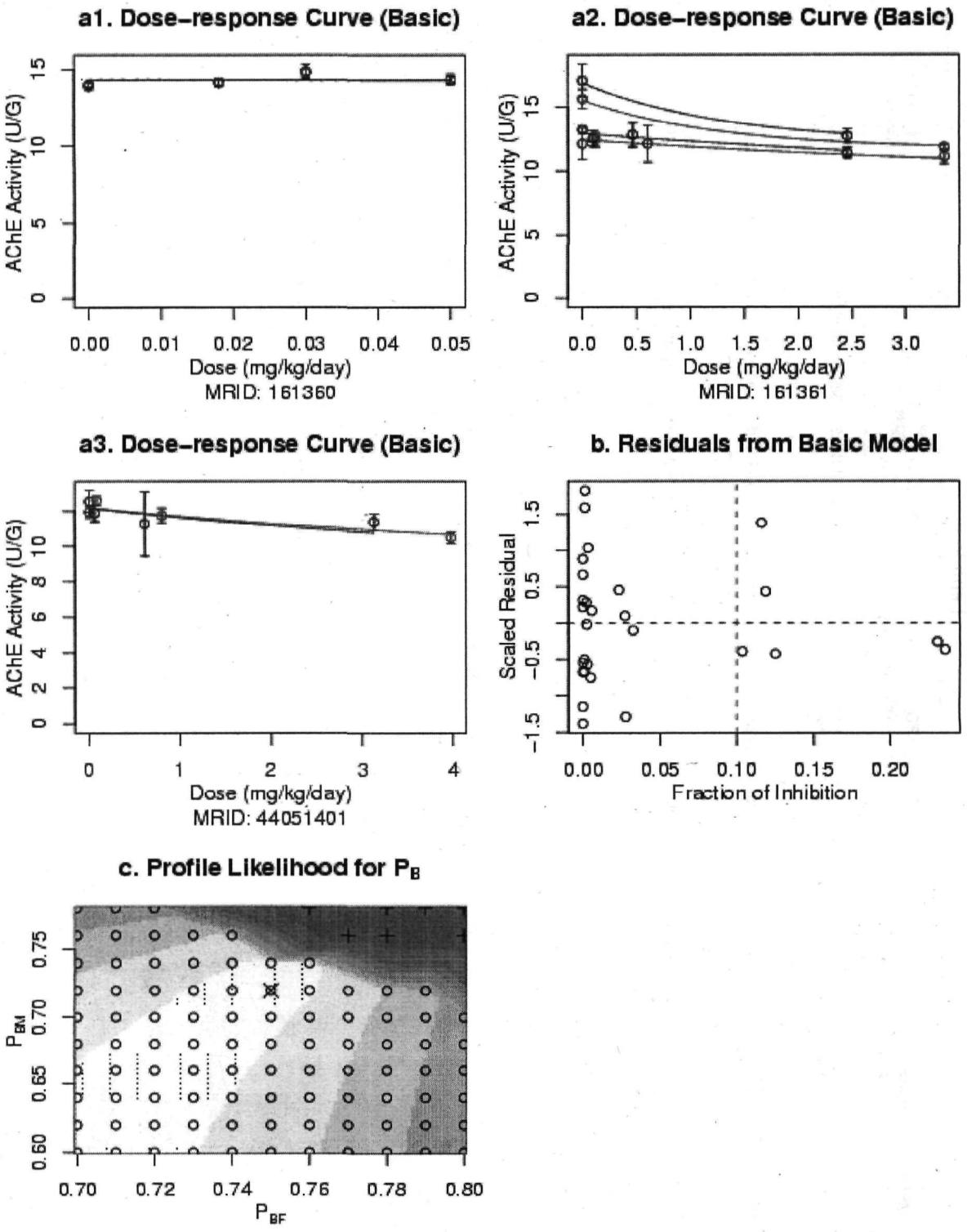


Table III.B.2-14. Fenthion: Toxicology Profile Table

Fenthion						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
41743101	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	011804 009870	0/0, 0.3/0.2, 1.3/0.8, 7.3/5.2 mg/kg/day	Guideline	Rat/ Fischer
44339401	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	012511	0, 0.17/0.13, 2.19/1.63,12.62/8.5 mg/kg/day (females/males)	Guideline	Rat/ Wistar
40329501	82-2 (870.3200)	21-Day Dermal Toxicity– Rabbit	011765	0, 5, 50, 100, 200, 400 mg/kg/day	Guideline	Rabbit/ New Zealand

Figure III.B.2-14. Fenthion: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

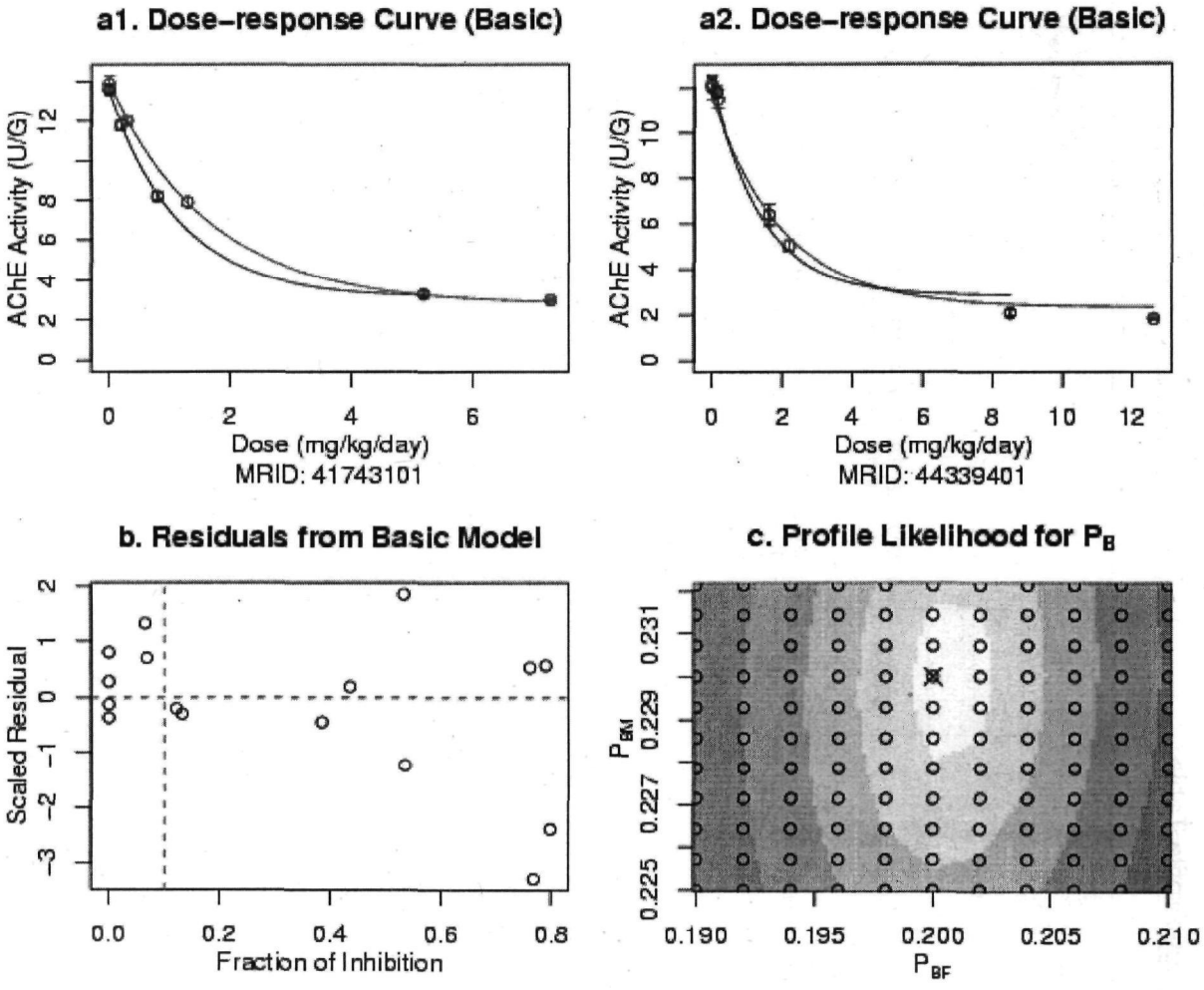


Table III.B.2-15. Fosthiazate: Toxicology Profile Table

Fosthiazate*						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
44269905	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	In review	0/0, 0.05/0.05, 0.1/0.1, 0.5/0.48, 1/0.97, 10.67/9.69, 43.52/40.87 mg/kg/day (females/males)	In review	Rat/ Charles River CD (remote SD origin)
41347632	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	008039	0/0, 0.09/0.08, 0.89/0.77, 4.74/4.12, 41.03/36.37 mg/kg/day (females/males)	Guideline	Rat/ CD
43559703	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	008039	0/0, 0.055/0.042, 0.54/0.41, 2.63/2.08, 12.53/8.94 mg/kg/day (females/males)	Guideline	Rat/ Charles River CD

*Not yet registered

Figure III.B.2-15. Fosthiazate: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

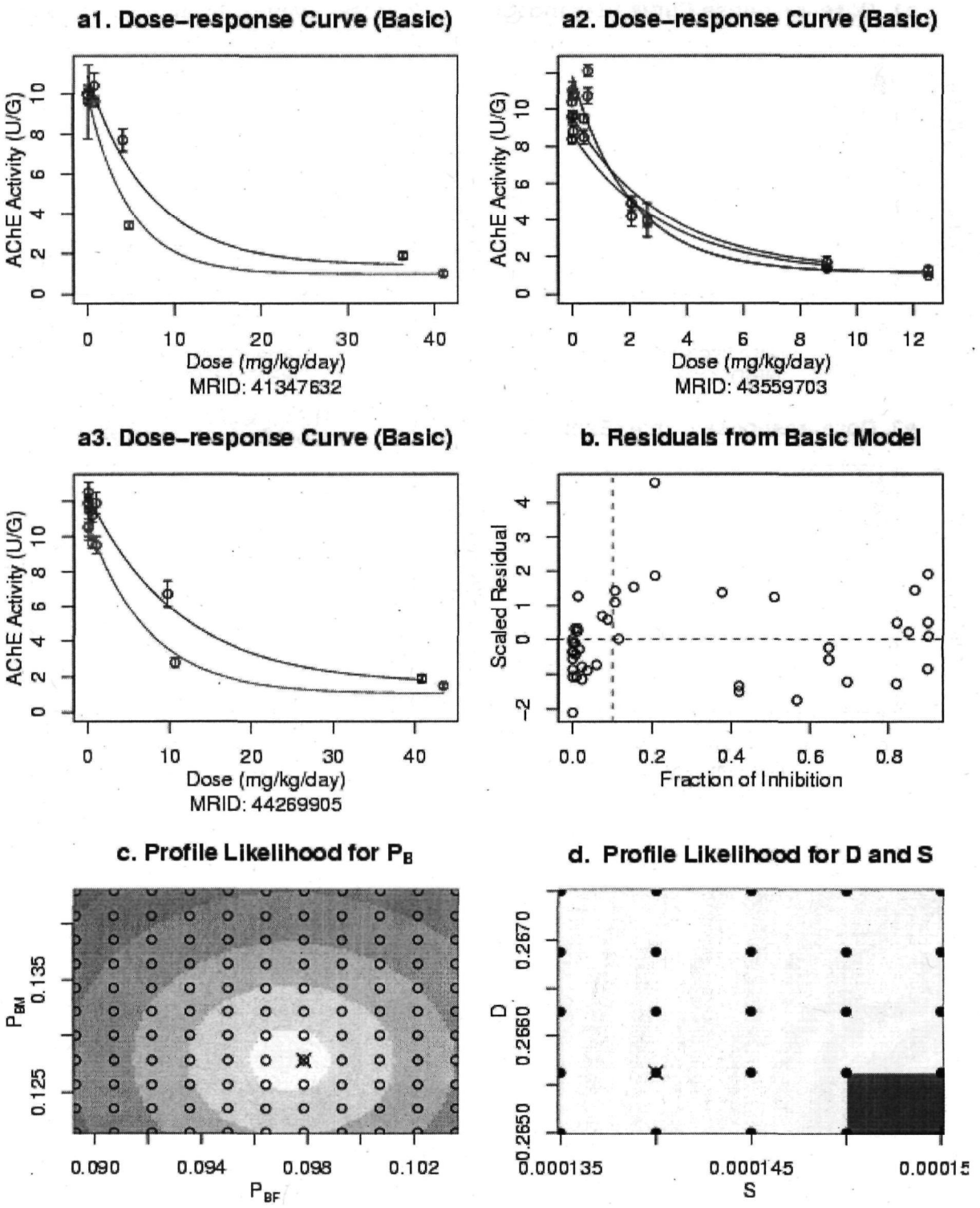


Figure III.B.2-15. Fosthiazate con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

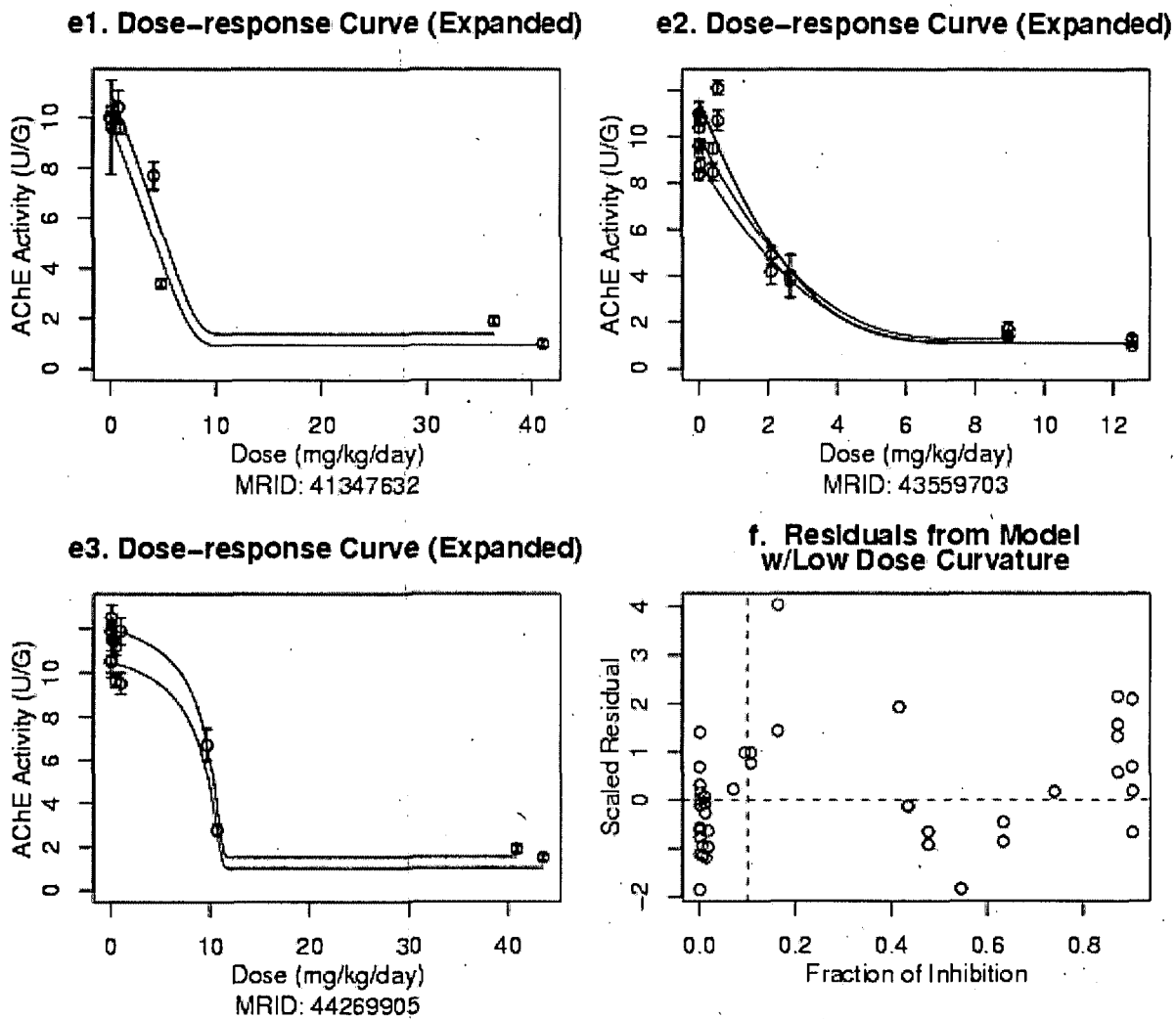


Table III.B.2-16. Malathion: Toxicology Profile Table

Malathion						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
43942901	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	013822 014120 014121	0/0, 5/4 , 35/29, 415/359, 868/739 mg/kg/day (females/males)	Guideline	Rat/ Fischer
41054201	82-2 (870.3200)	21-Day Dermal Toxicity–Rabbit	008714 009385 012433	0, 50, 300, 1000 mg/kg/day	Guideline	Rabbit/ New Zealand Albino
43266601	82-4 (870.3465)	13-Week Inhalation Toxicity–Rat	012433 011516	0 (air), 0.1, 0.45, 2.01 mg/L	Nonguideline	Rat/ Sprague Dawley

Figure III.B.2-16. Malathion: Dose-Response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

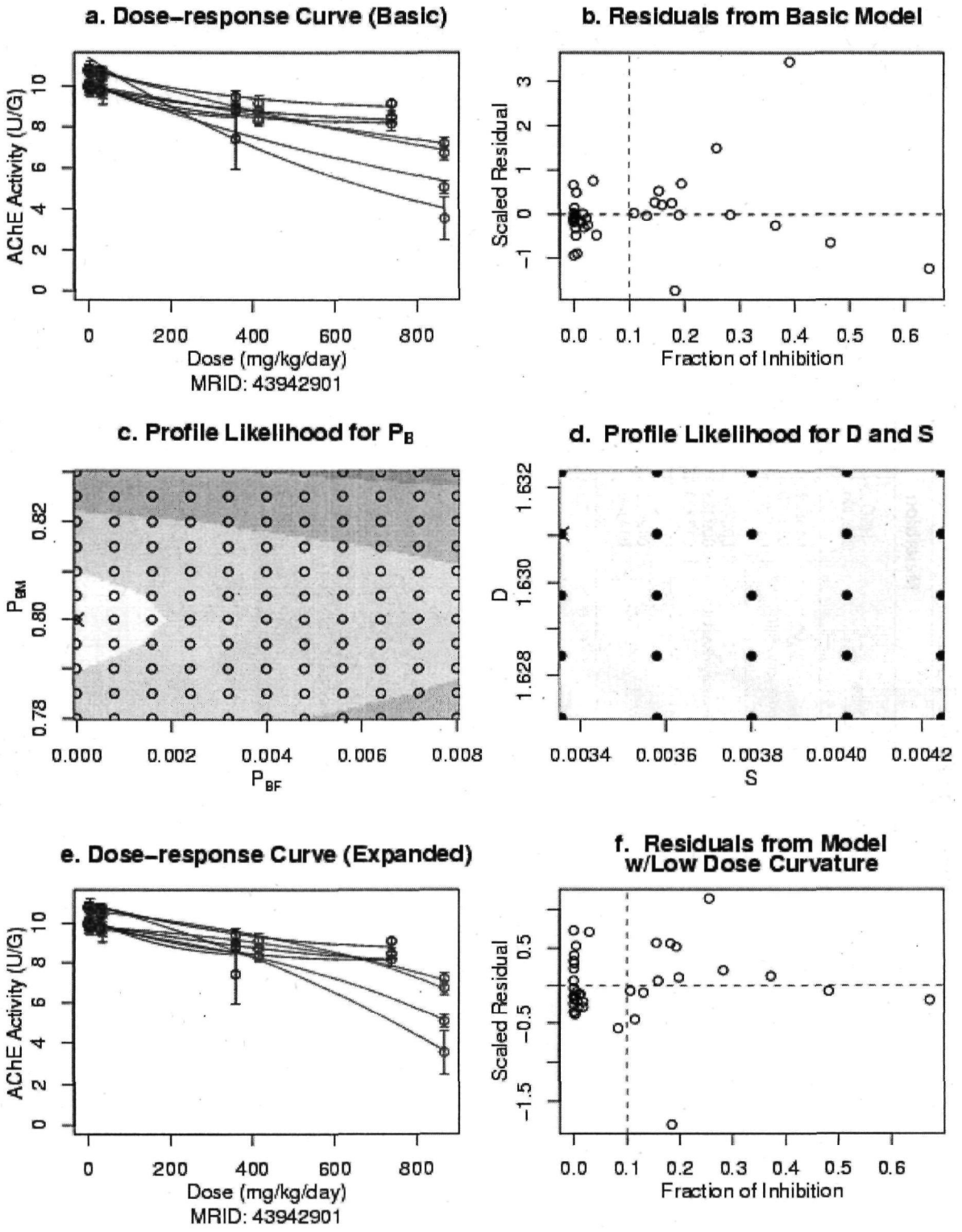


Table III.B.2-17. Methamidophos: Toxicology Profile Table

Methamidophos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
41867201	82-1 (870.3100)	Subchronic Oral Toxicity–Rat (Special ChE study)	008846 012826	0/0, 0.06/0.03, 0.06/0.07, 0.17/0.13, 0.28/0.24 mg/kg/day (females/males)	Guideline	Rat/ Fischer
43197901	82-7 (870-6200)	Subchronic Neurotoxicity–Rat	011530 012826	0/0, 0.07/0.07, 0.90/0.79, 4.94/4.26 mg/kg/day (females/males)	Guideline	Rat/ Fischer
00148452	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	005313 007124 012514	0/0, 0.116/0.095, 0.351/0.288, 1.056/0.848, 3.49/2.847 mg/kg/day (females/males)	Guideline	Rat/ Fischer
44525301	82-2 (870.3200)	21-Day Dermal Toxicity–Rat	13394	0, 0.75, 11.2, 36.5 mg/kg/day	Guideline	Rat/ Sprague Dawley
00147935	82-2 (870.3200)	21-Day Dermal Toxicity–Rabbit	11779	0, 0.5, 5 mg/kg/day	Nonguideline	Rabbit/ NZW
41402401	82-3 (870.3465)	Subchronic Inhalation Toxicity–Rat	011550 012826	Air and vehicle [PEG E400:ethanol] controls, 0.0011, 0.0054, 0.0231 mg/L	Guideline	Rat/ Wistar

Figure III.B.2-17A. Methamidophos: Dose-response Curves Using the Basic Model for the Oral Route, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

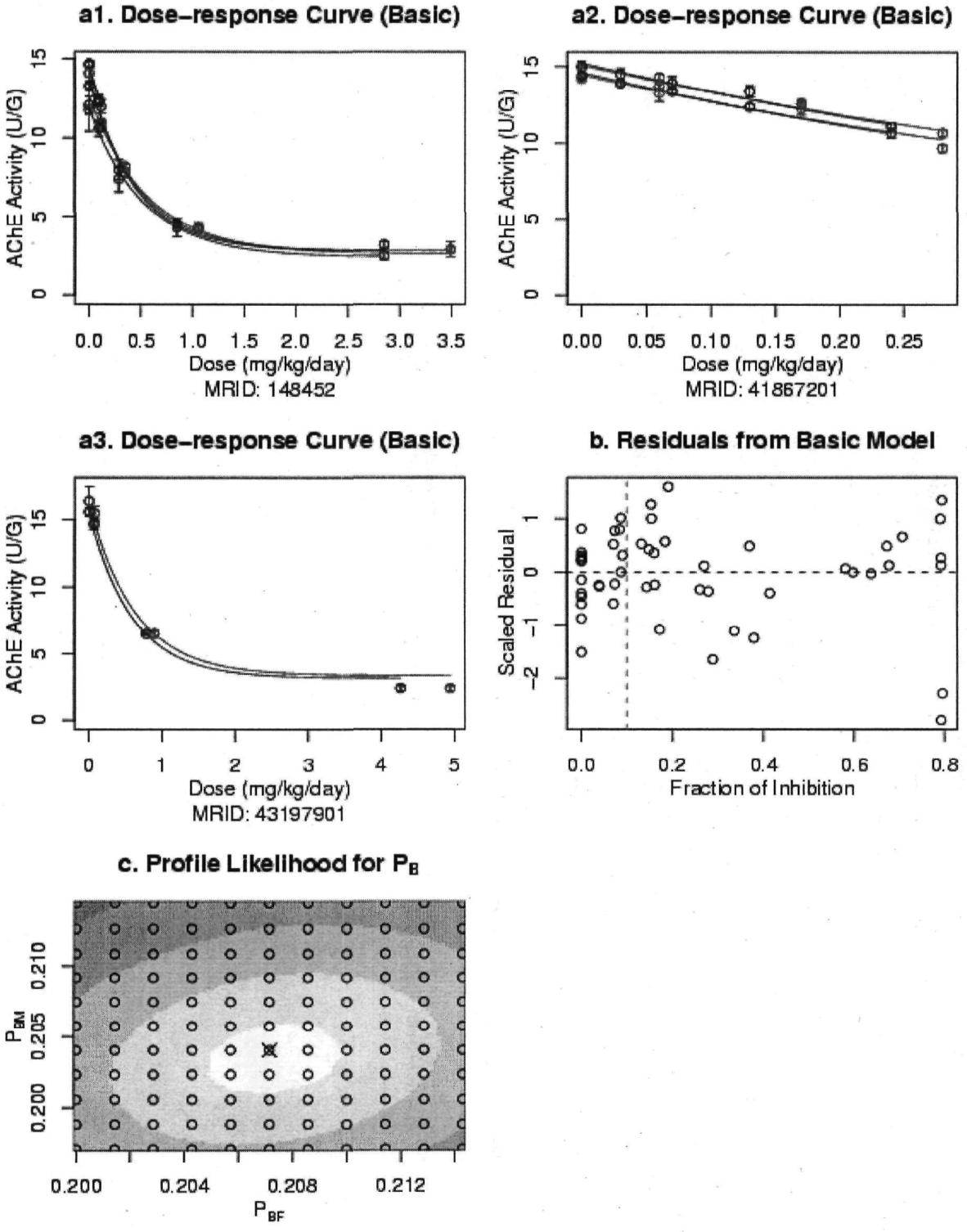
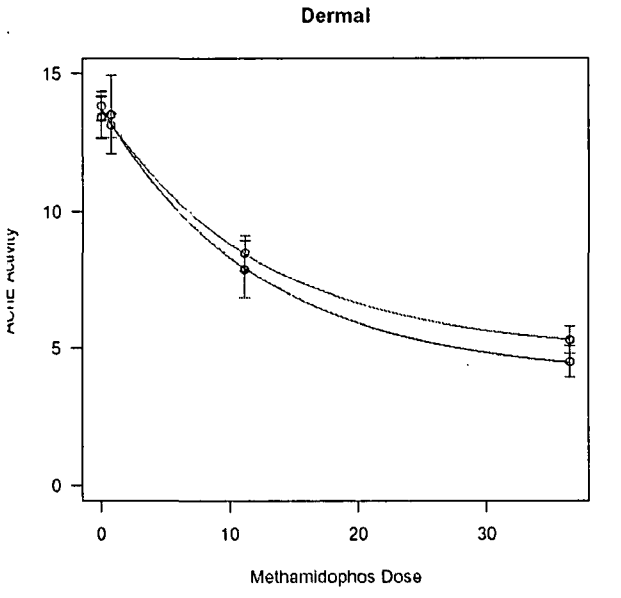


Figure III.B.2-17B. Methamidophos: Dose-response Curves Using the Basic Model for the Dermal and Inhalation Routes.



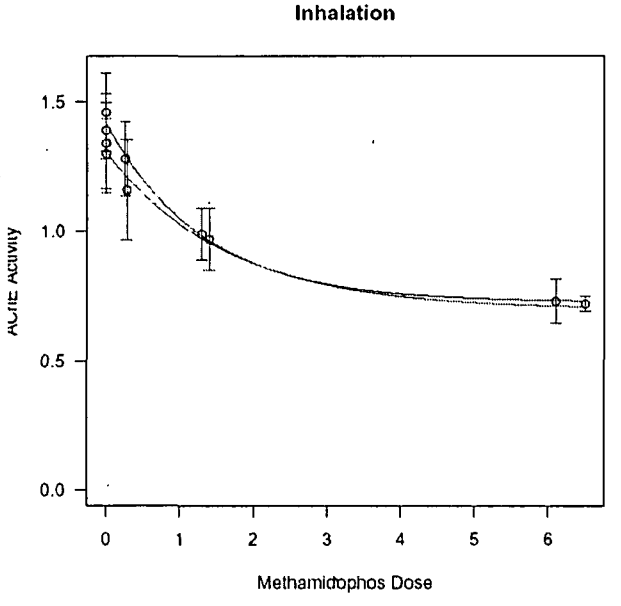


Table III.B.2-18. Methidathion: Toxicology Profile Table

Methidathion						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
00160260	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	005743 006587	0/0, 0.22/0.16, 2.2/1.72, 6.93/4.91 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley

Figure III.B.2-18. Methidathion: Dose-response Curve Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

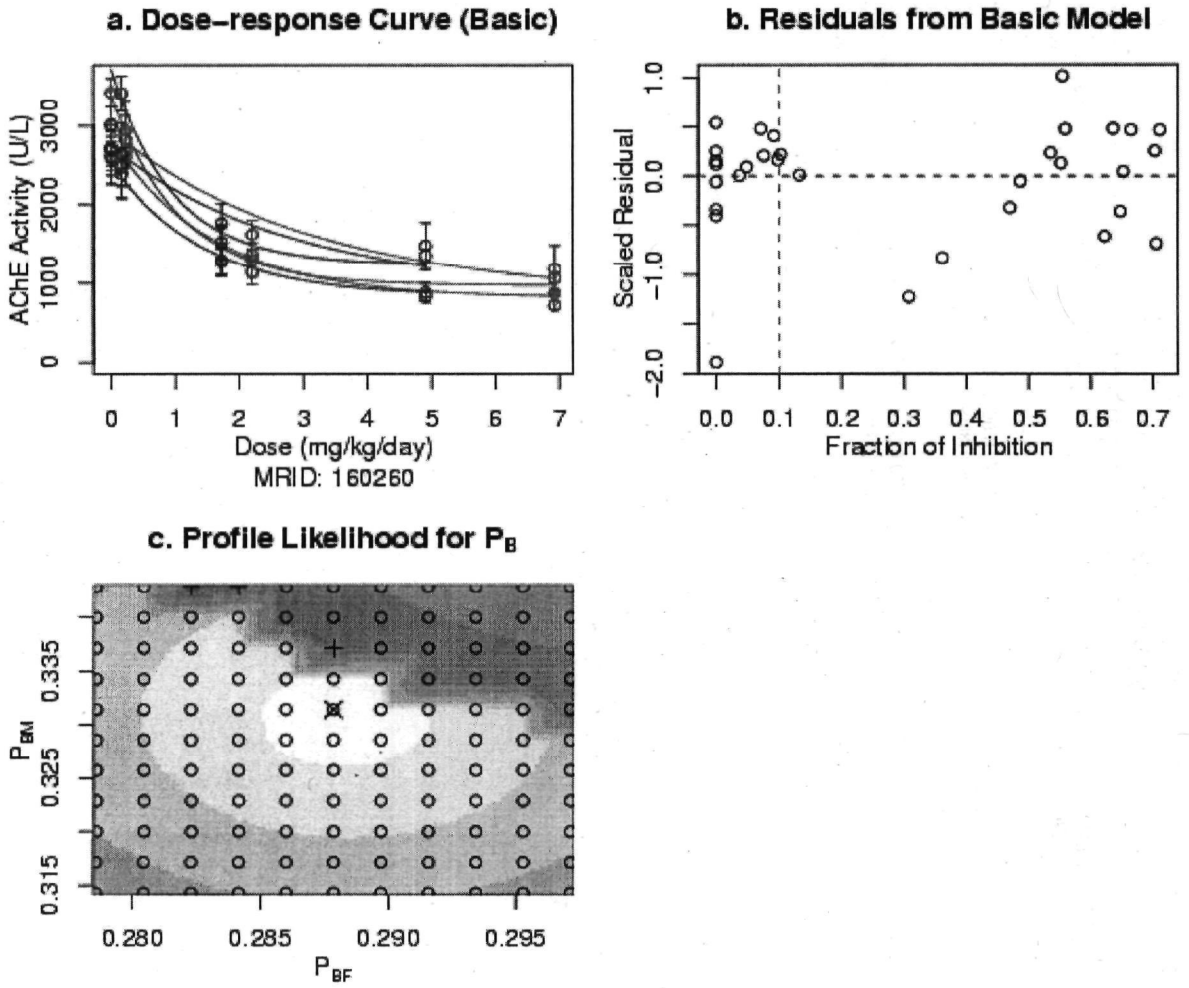


Table III.B.2-19. Methyl Parathion: Toxicology Profile Table

Methyl Parathion						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
00074299	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	001882	0/0, 0.20/0.16, 2.10/1.64, 6.90/5.90 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
41853801	83-1 (870.4100)	Chronic Oral Toxicity with Special Focus on Sciatic Nerve Effects	010333	0, 0.03/0.02, 0.14/0.11, 0.70/0.53, 3.09/2.21 mg/kg/day (females/males)	Nonguideline	Rat/ Sprague Dawley

Figure III.B.2-19. Methyl-parathion: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

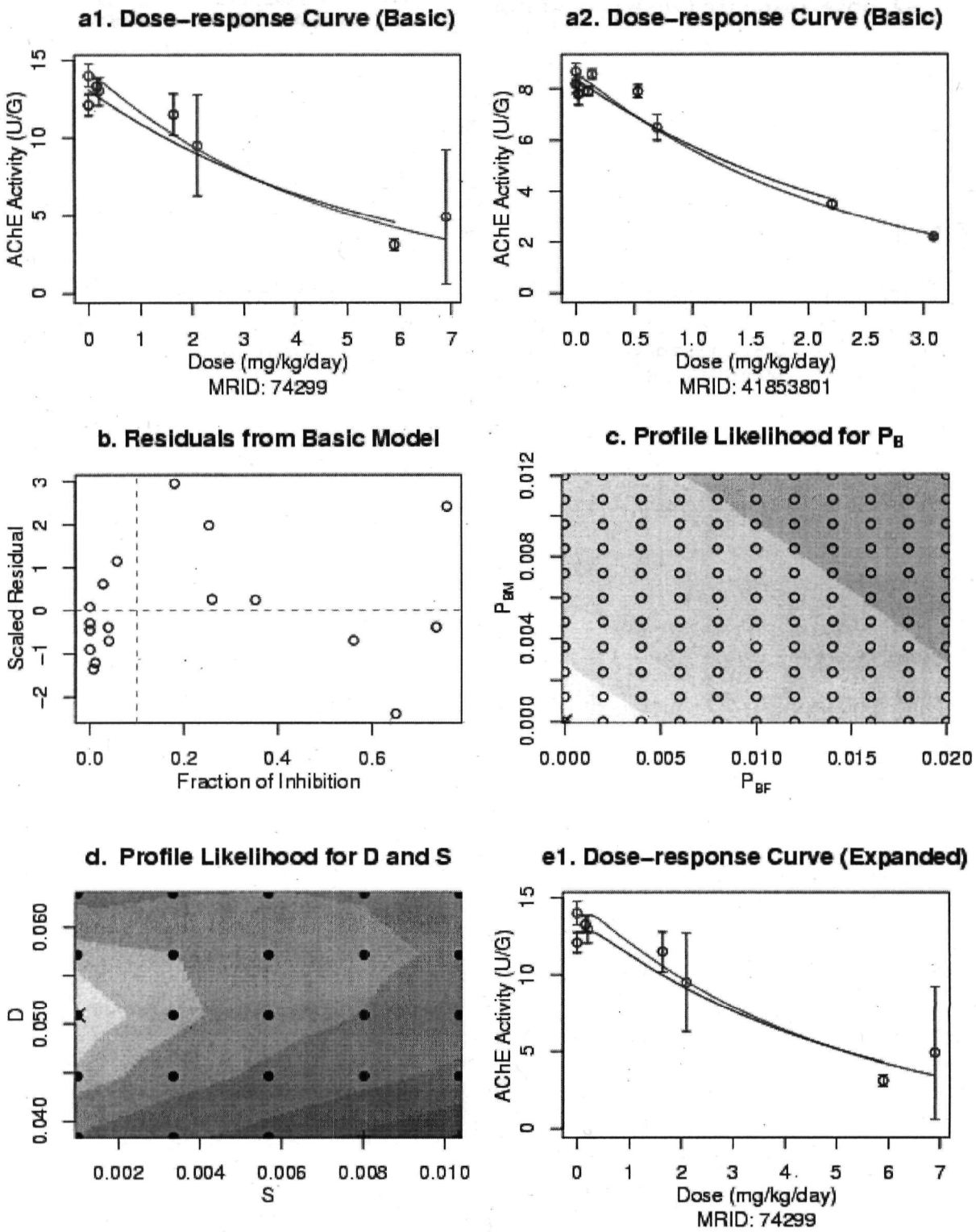


Figure III.B.2-19. Methyl-parathion con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

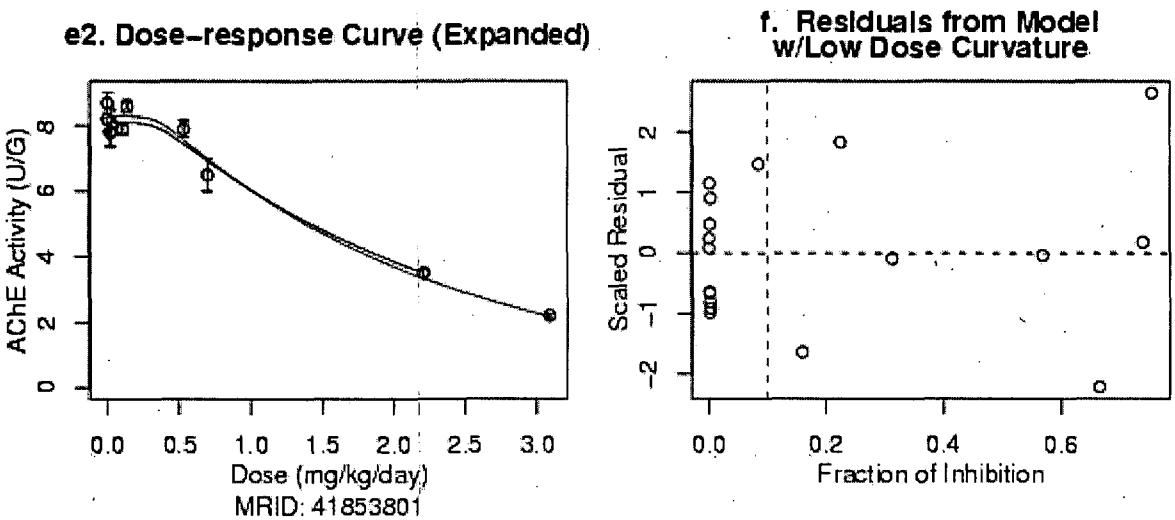


Table III.B.2-20. Mevinphos: Toxicology Profile Table

Mevinphos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
42588501	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	015801	0/0, 0.011/0.056, 0.056/0.56, 0.56/1.12, 0.84/1.67 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley

Figure III.B.2-20. Mevinphos: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

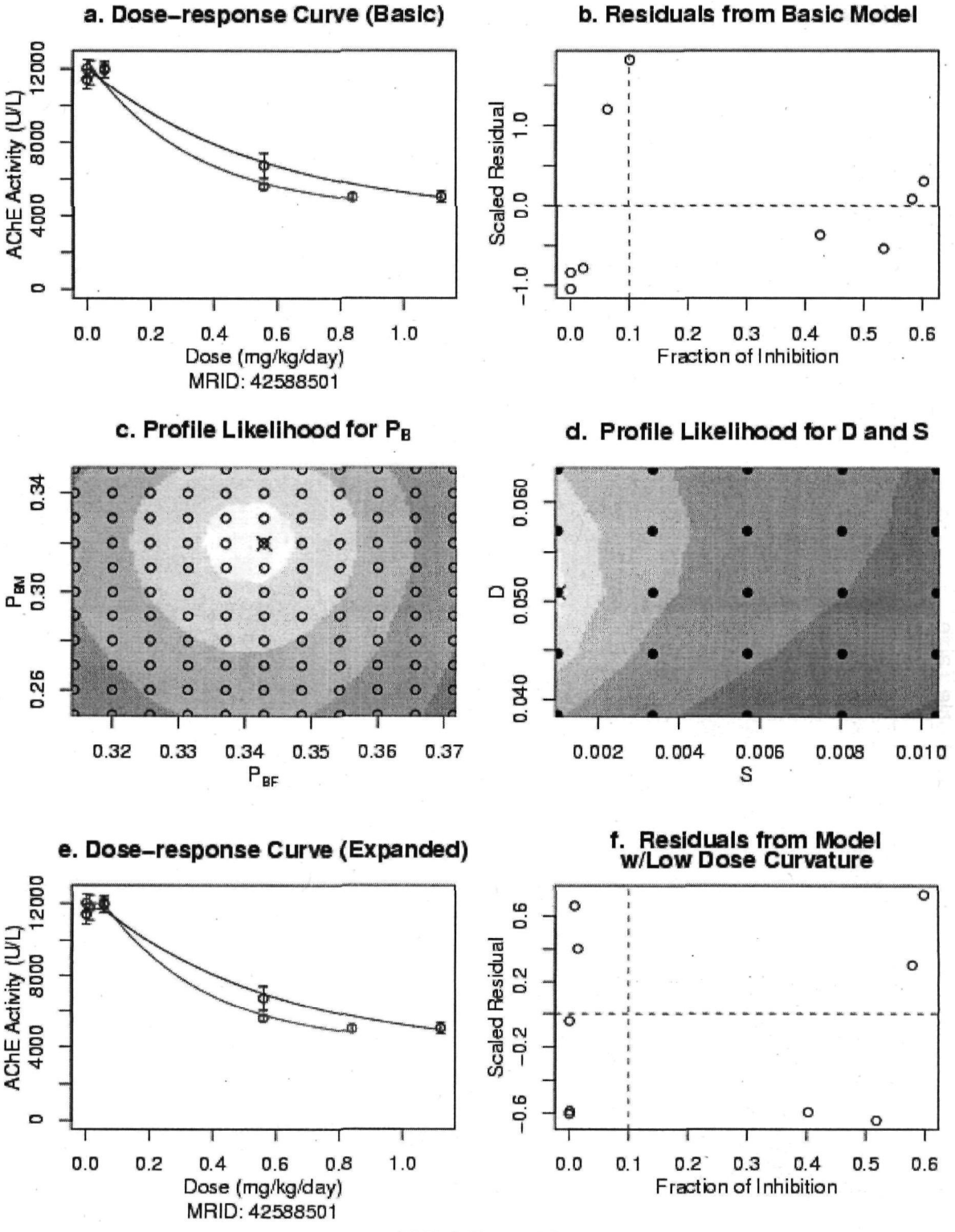


Table III.B.2-21. Naled: Toxicology Profile Table

Naled						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
00088871	82-1 (870.3100)	Four-Week Subchronic Oral (Gavage) Toxicity–Rat	1460	0, 0.25, 1, 10, 100 mg/kg/day (gavage)	Supplementary	Rat/Sprague Dawley
00141784	83-5 (870.4300)	Combined Chronic Oral (Gavage) Toxicity/Carcinogenicity–Rat	002997 004128 004521	0, 0.2, 2, 10 mg/kg/day (gavage)	Guideline	Rat/Sprague Dawley
45222001	82-2 (870.3200)	28-Day Dermal Toxicity–Rat	0144336	0, 5, 10, 40 mg/kg/day	Guideline	Rat/Sprague Dawley
00160750	82-2 (870.3200)	28-Day Dermal Toxicity–Rat	5774	0, 1, 20, 80 mg/kg/day	Guideline	Rat/Sprague Dawley
00164224	82-4 (870.3465)	Subchronic Inhalation Toxicity–Rat	5784	0, 0.2, 1.2, or 6 µg/L	Guideline	Rat/ Fischer
40087201	82-4 (870.3465)	21-Day Inhalation Toxicity–Rat	004580 006709	0 (air), 4, 8, 16 µg/L (nominal) actual chamber concentration: 0, 3.4, 7.2, 12.1 µg/L	Supplementary	Rat/ Fischer

Figure III.B.2-21. Naled: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

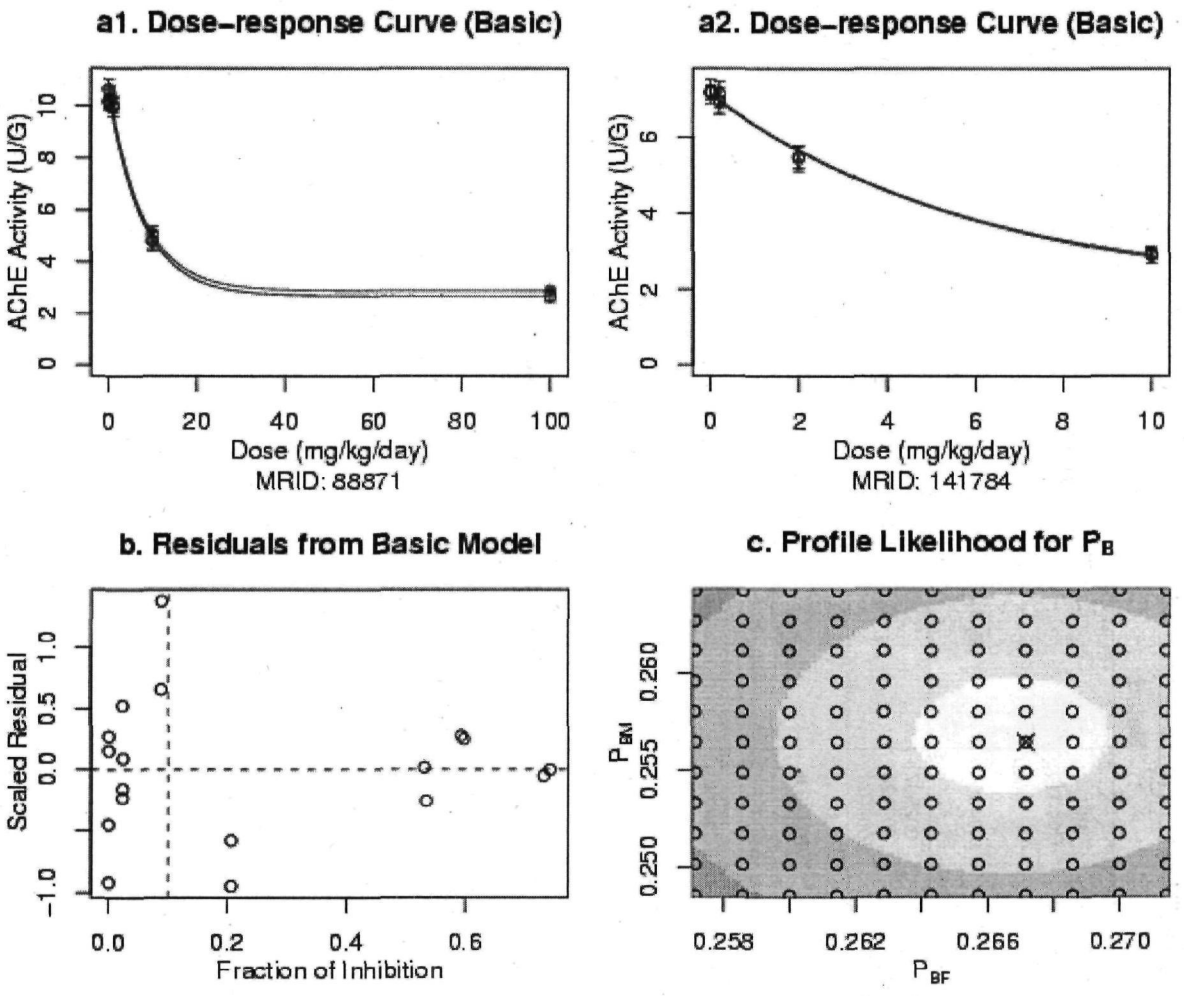


Table III.B.2-22. Omethoate: Toxicity Profile Table

Omethoate						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
ACP28Day (MRID Not assigned)*	NA	28-Day Feeding Study - Rat	NA	0, 0.01, 0.02, 0.04, 0.08, 0.4 mg/kg/day	NA	Rat/Nelson

*Fax and email communications from D. Allemang, Cheminova, Inc. to A. Lowit, EPA, 3/18/02, 3/20/02, 3/27/02
NA=Not applicable

Figure III.B.2-22. Omethoate: Dose-response Curve Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

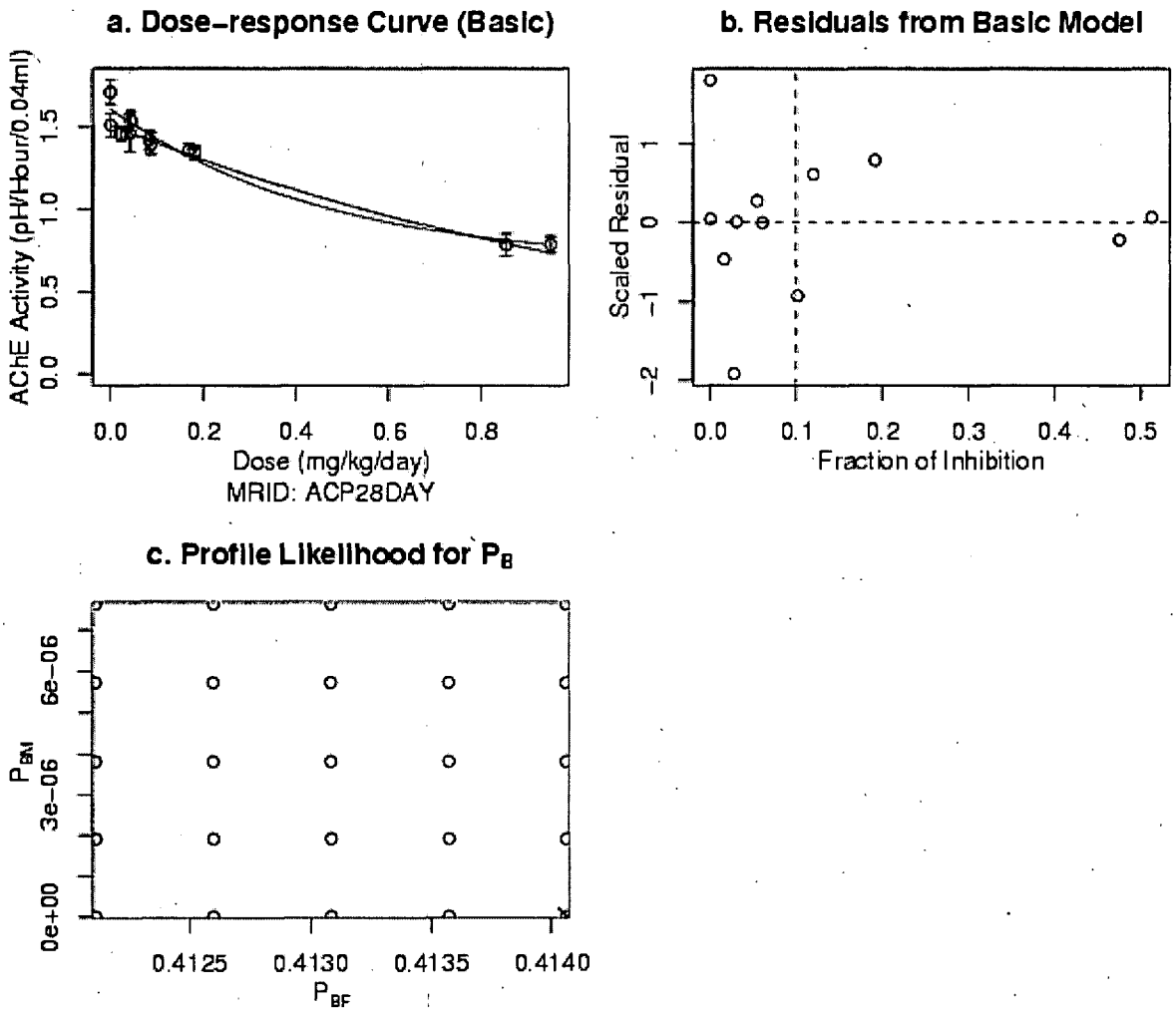


Table III.B.2-23. Oxydemeton-methyl: Toxicology Profile Table

Oxydemeton-methyl						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
00151806	83-5 870.4300	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	005174 005752 009544	0/0, 0.06/0.05, 0.62/0.49, 6.92/5.84 mg/kg/day (females/males)	Guideline	Rat/ Fischer
00143351	82-1 870.3100	Subchronic Oral Toxicity–Rat	005752	0/0, 0.09/0.08, 0.93/0.75, 13.22/8.25 mg/kg/day (females/males)	Supplementary	Rat/ SPF
41834002	Non-guideline	Special NTP Study	012221	0, 0.15, 0.45 or 2.5 mg/kg/day (males only)	Nonguideline	Rat/ Sprague Dawley
44141301	82-1 870.3100	Subchronic Oral Toxicity (13-week Cholinesterase Study)–Rat	012216	0/0, 0.0073/0.006,0.0224/0.0184, 0.074/0.0616, 0.7475/0.6201, 6.5697/5.3925 mg/kg/day (females/males)	Nonguideline	Rat/ Sprague Dawley

Figure III.B.2-23. Oxydemeton-methyl: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot For P_B

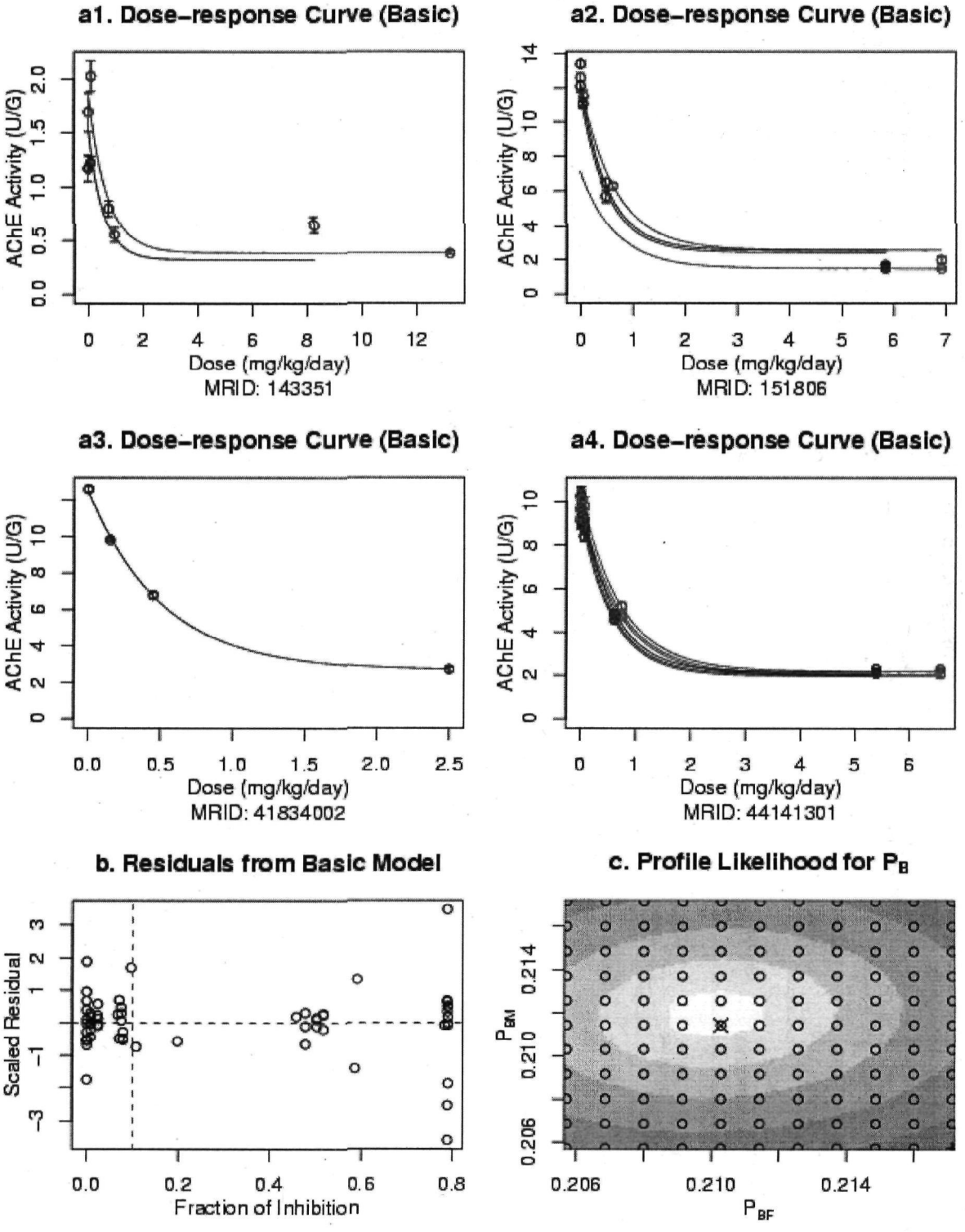


Table III.B.2-24. Phorate: Toxicology Profile Table

Phorate						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
44895301	82-1 (870.3100)	21-Day Rangefinding–Rat	137767	0/0, 0.10/0.09, 0.20/0.19, 0.52/0.69 mg/kg/day (females/males)	Supplementary	Rat/ Sprague Dawley
44895302	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	13767	0/0, 0.04/0.04, 0.08/0.07, 0.33/0.54 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley

Figure III.B.2-24. Phorate: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

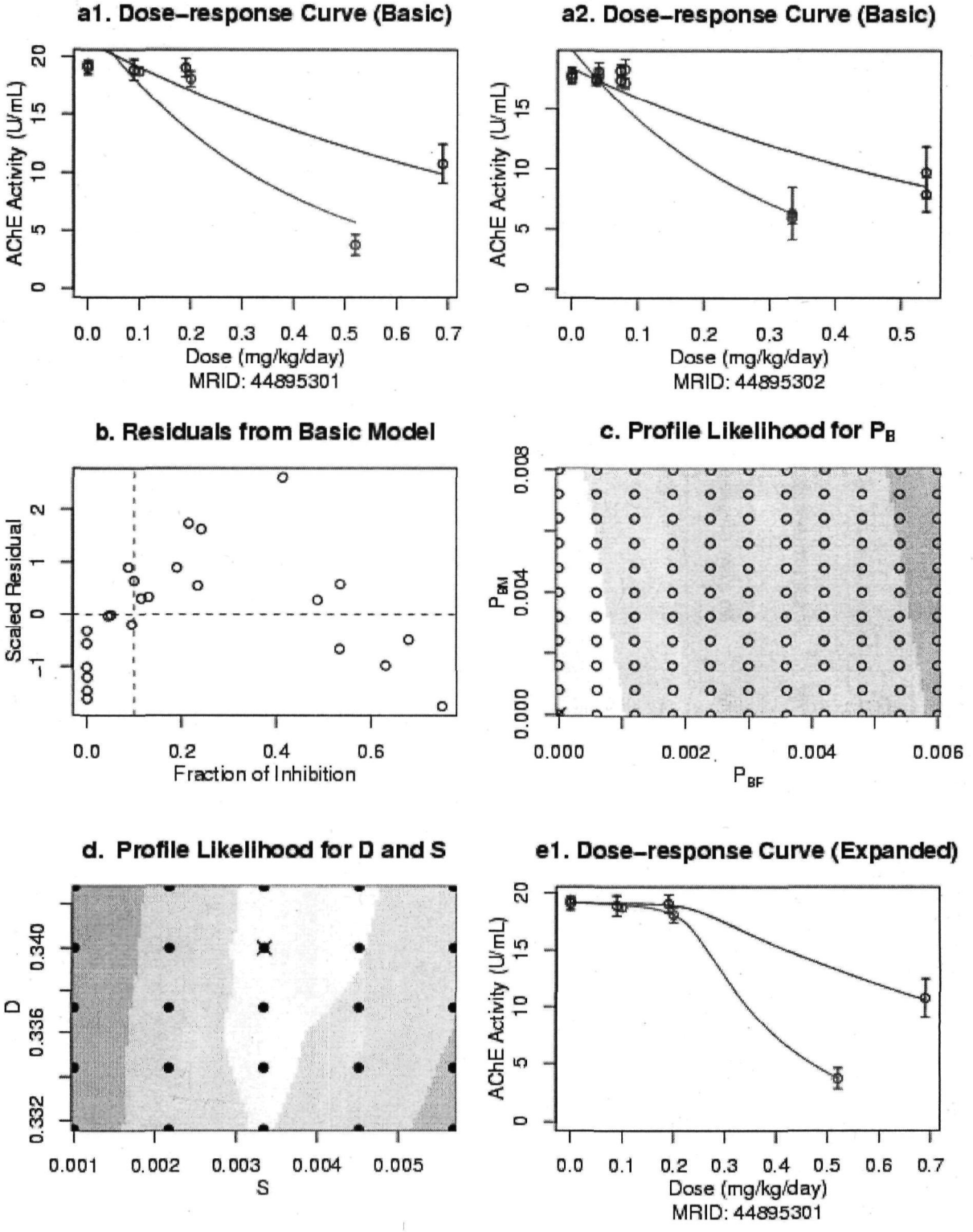


Figure III.B.2-24. Phorate con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

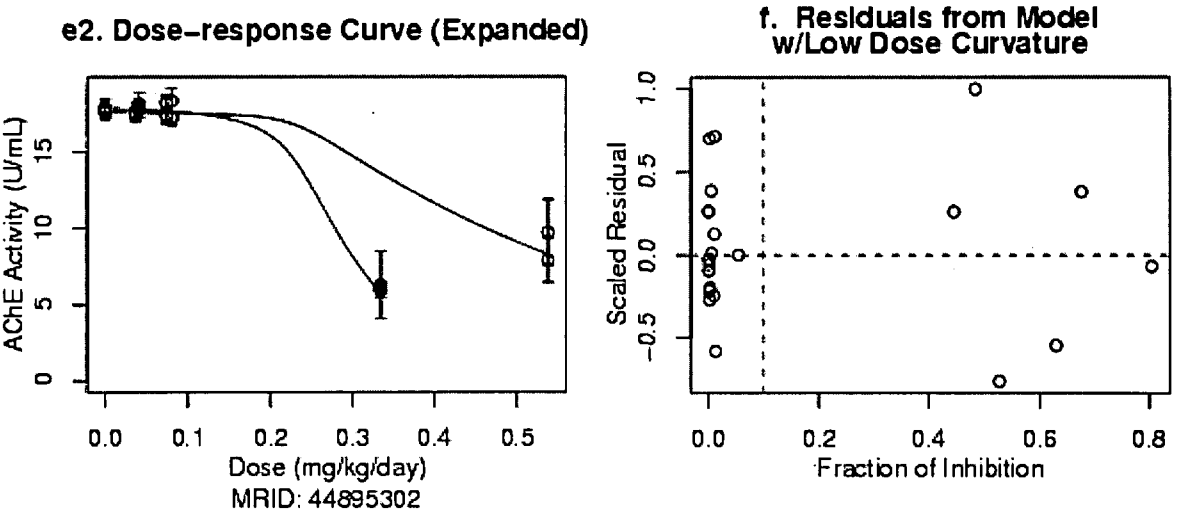


Table III.B.2-25. Phosalone: Toxicology Profile Table

Phosalone						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
44801002	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	13753	0/0, 0.28/0.23, 2.87/2.19, 46.54/31.82 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
45317902	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	13753	0/0, 5/4.6, 14.70/13.80, 61.90/55.80 mg/kg/day (females/males)	Guideline	Rat/ Crl:CD BR

Figure III.B.2-25. Phosalone: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

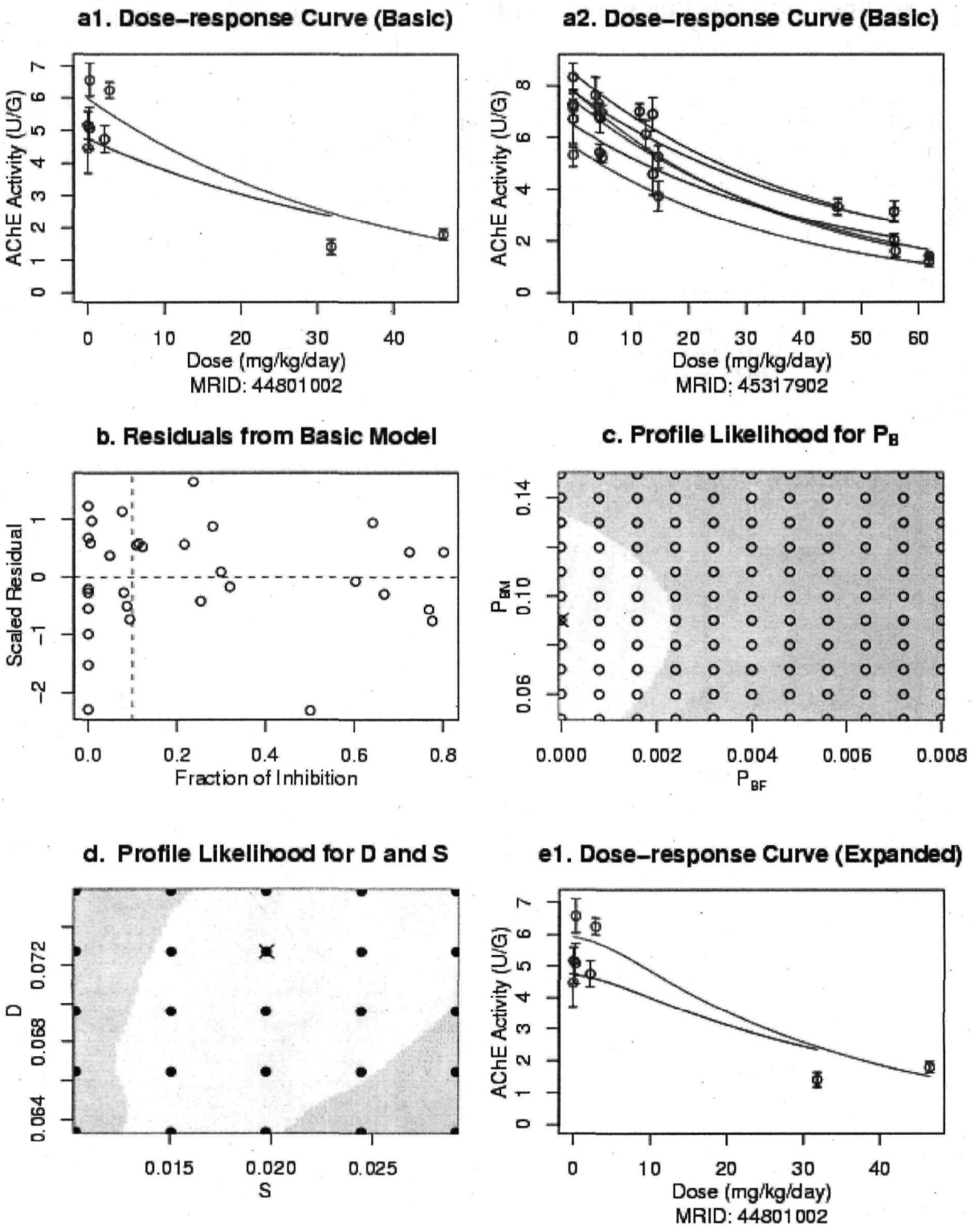


Figure III.B.2-25. Phosalone con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

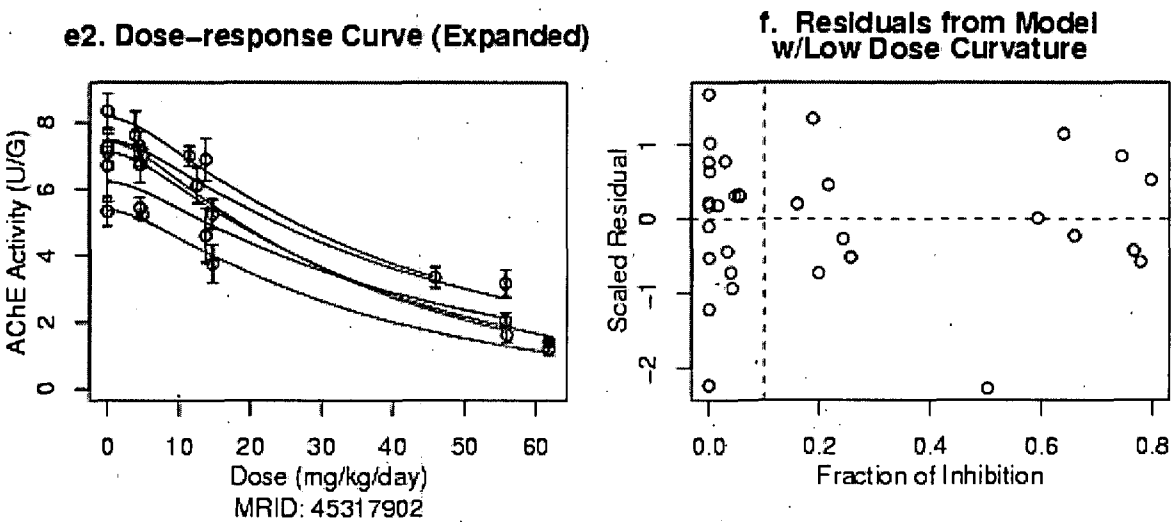


Table III.B.2-26. Phosmet: Toxicology Profile Table

Phosmet						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
41916401	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	9828 10756	0/0, 1.1/1.1, 2.1/1.8, 10.9/9.4, 27.1/22.7 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
44811801	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	13522	0/0, 1.9/1.7, 3.9/3.4, 12.1/10.4 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley

Figure III.B.2-26. Phosmet: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

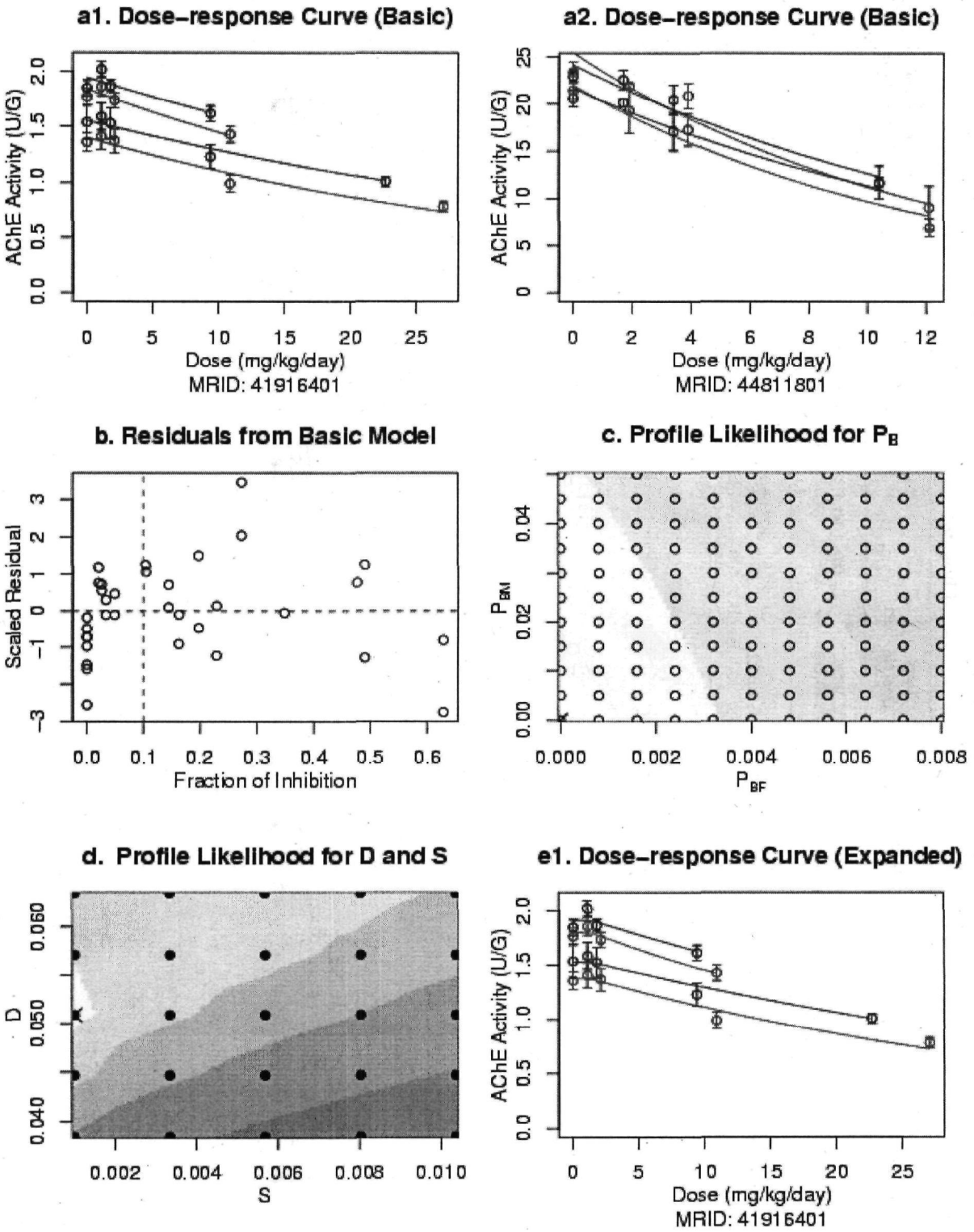


Figure III.B.2-26. Phosmet con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

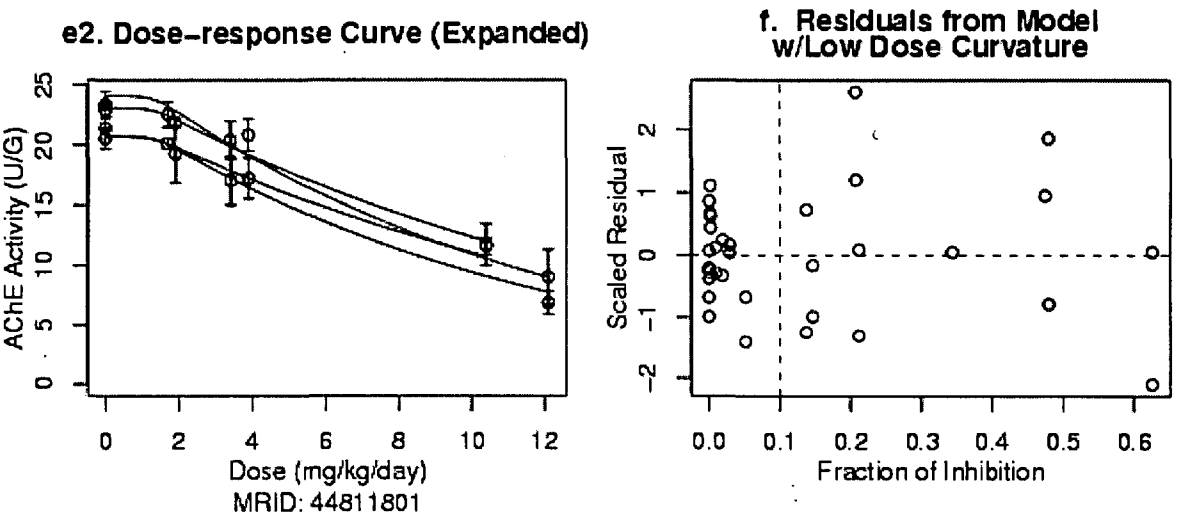


Table III.B.2-27. Phostebupirim: Toxicology Profile Table

Phostebupirim						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
43656302	82-7 (870.6200)	Subchronic Dietary Neurotoxicity - Rat	013283	0, 0.30/0.26, 0.96/1.2, and 3.6/4.4 mg/kg/day (females/males)	Guideline	Rat/Fischer
42005451	83-5 (870.4300)	Combined Chronic Oral Toxicity/Oncogenicity - Rat	009954	0/0, 0.08/0.06, 0.42/0.30, 2.37/1.71 mg/kg/day (females/males)	Minimum	Rat/Wistar
42005447	82-1 (870.3100)	Subchronic Oral Toxicity - Rat	009954	0/0, 0.2/0.2, 0.4/0.3, 1.2/1.0, 4.9/3.6 mg/kg/day (females/males)	Guideline	Rat/Wistar

Figure III.B.2-27. Phostebupirim: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

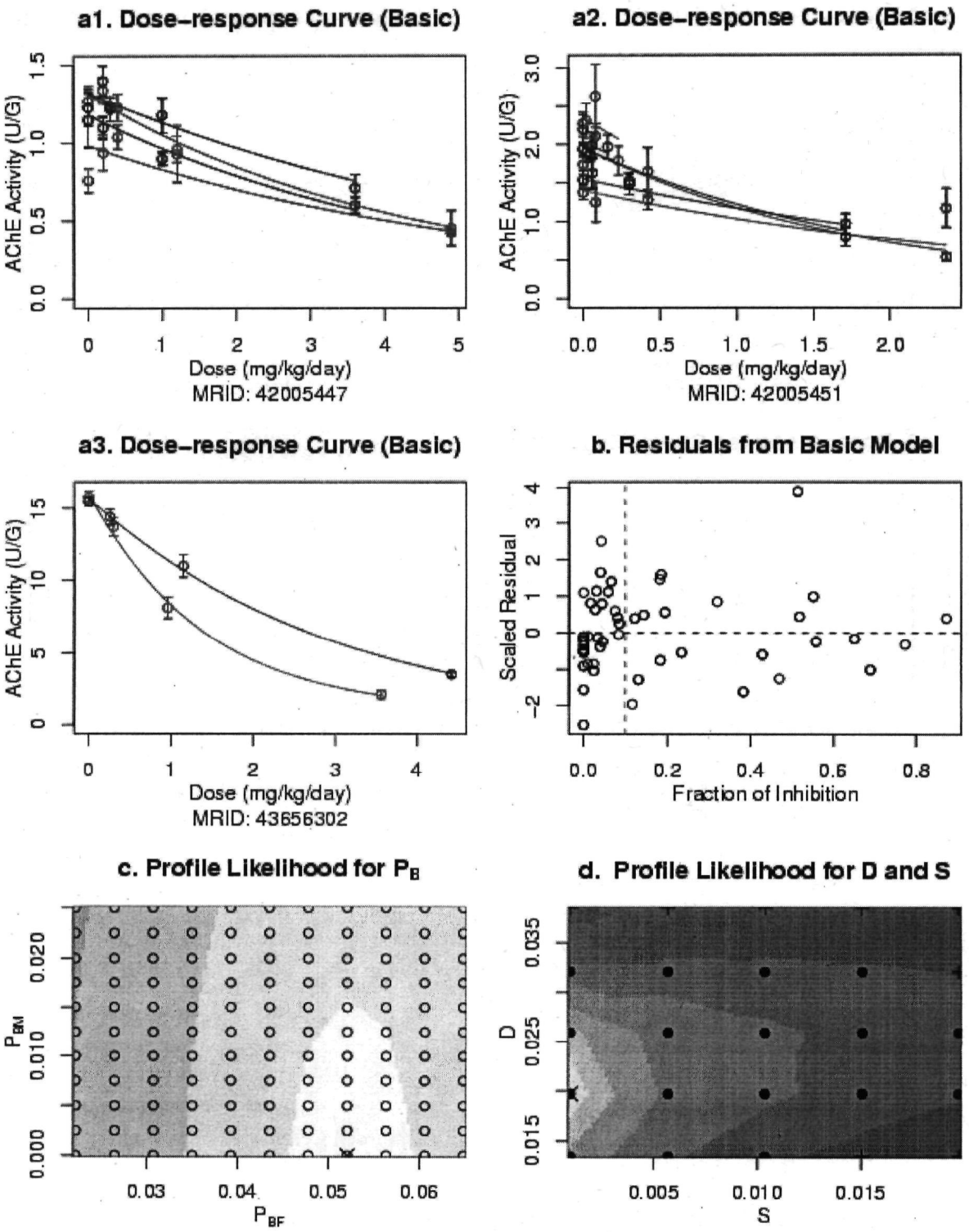


Figure III.B.2-27. Phostebupirim con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

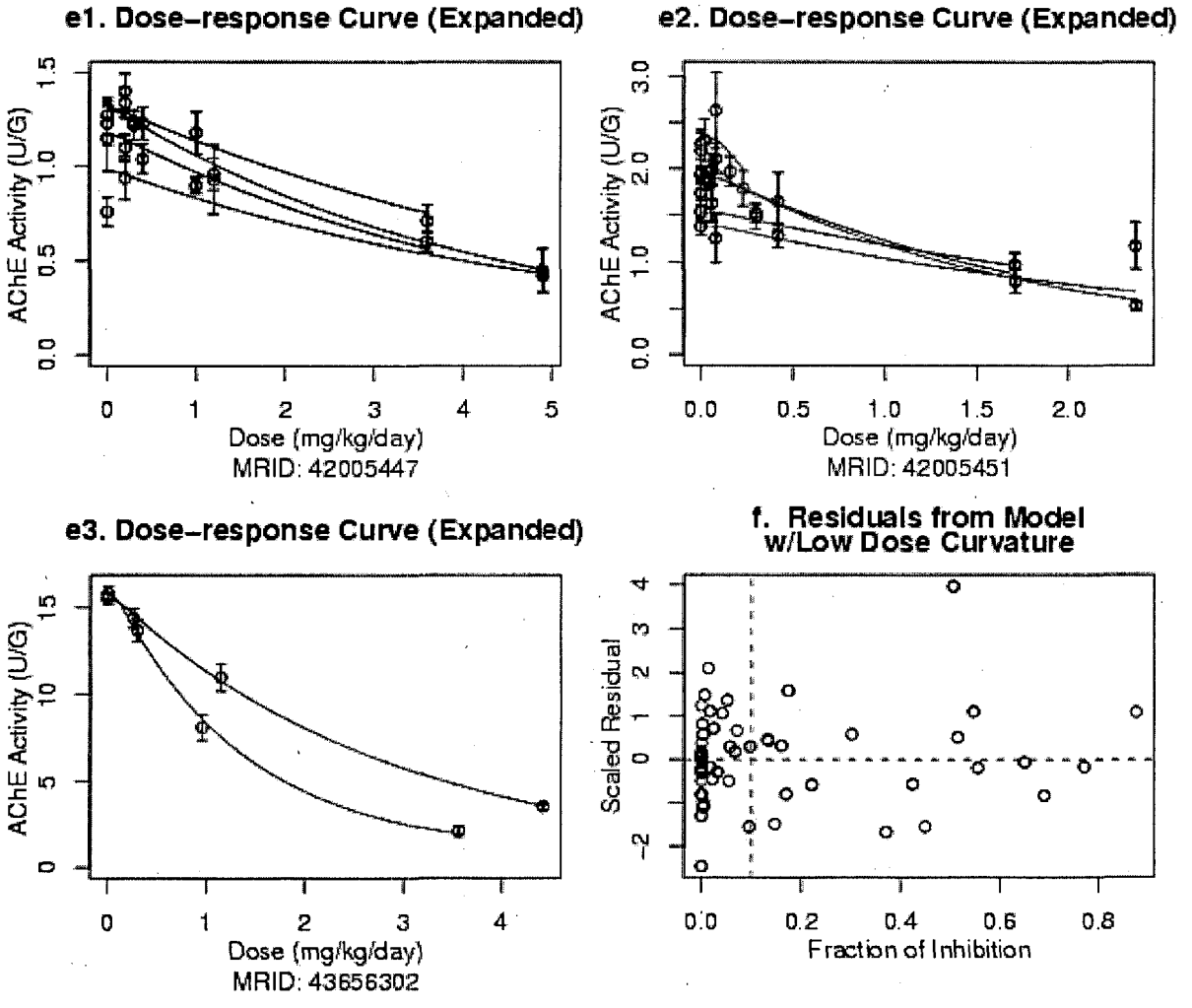


Table III.B.2-28. Pirimiphos-methyl: Toxicology Profile Table

Pirimiphos-methyl						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
00129343	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	014067 3582	0, 0.25, 0.40, 0.50, 2.50 mg/kg/day	Guideline	Rat/ Wistar
92147035	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	14067 3582 5105 8819	0/0,0.4/0.4, 2.1/2.1, 12.6/12.6 mg/kg/day (females/males)	Guideline	Rat/ Wistar

Figure III.B.2-28. Pirimiphos-methyl: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

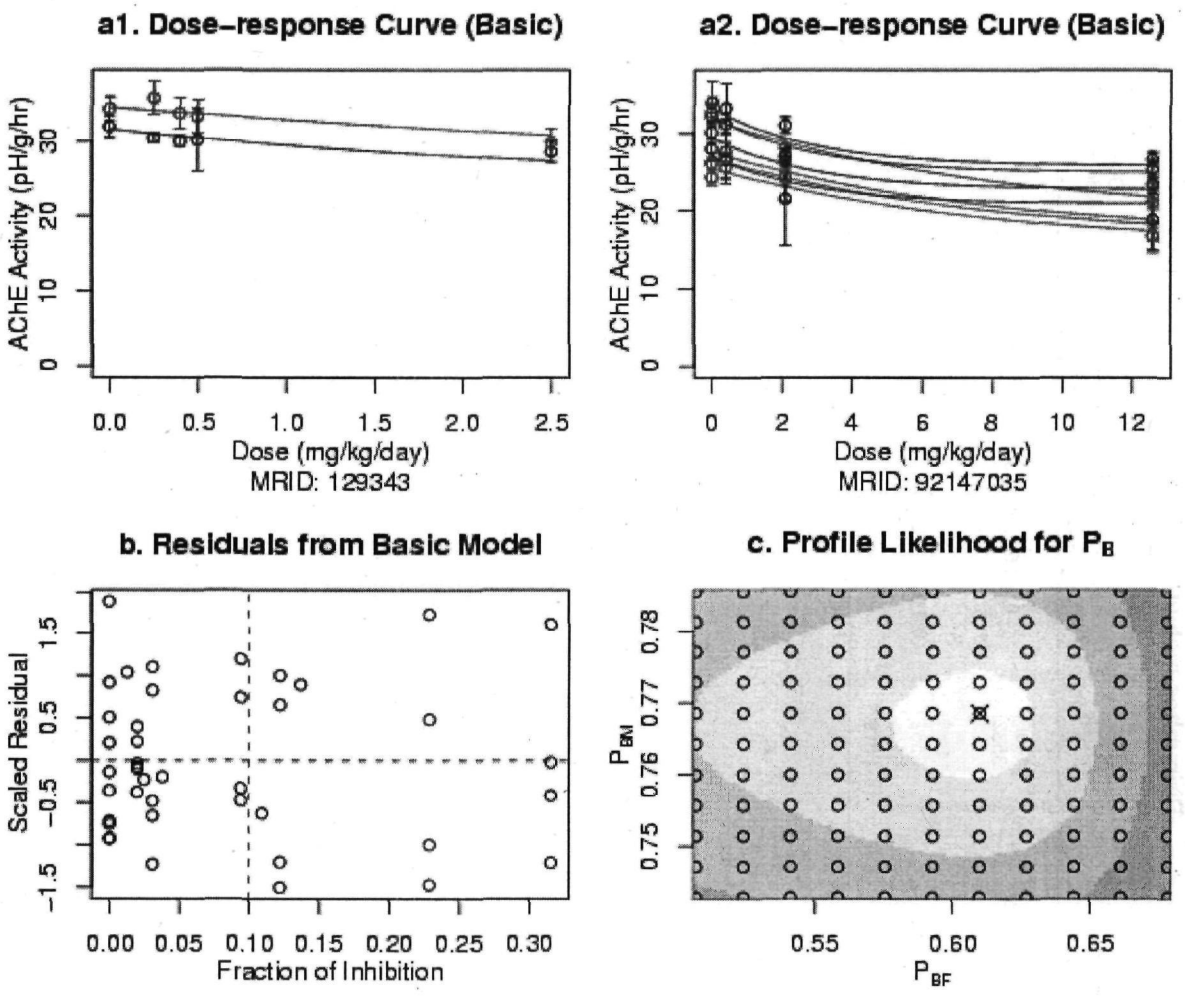


Table III.B.2-29. Profenofos: Toxicology Profile Table

Profenofos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
92148022	82-1 (870.3100)	Subchronic Oral Toxicity - Rat	NA	0/0, 0.001/0.001,0.003/0.003, 0.01/0.009,0.03/0.02, 0.09/0.09,0.25/0.21, 0.96/0.87, 2.6/2.1, 9.2/8.4, 24.8/21.1, 96.8/85.9 (females/males)	NA	Rat/Fischer
43213303	82-7 (870.6200)	Subchronic Dietary Neurotoxicity - Rat	011795	0/0, 1.84/1.7, 8.4/7.7, 37.9/36 mg/kg/day (females/males)	Acceptable	Rat/Sprague Dawley
92148031	83-5 (870.4300)	Combined Chronic Oral Toxicity/Oncogenicity - Rat	011916	0/0, 0.02/0.017, 0.694/0.559, 6.951/5.685 mg/kg/day (females/males)	Acceptable	Rat/Fischer

NA=Not available

Figure III.B.2-29. Profenofos: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

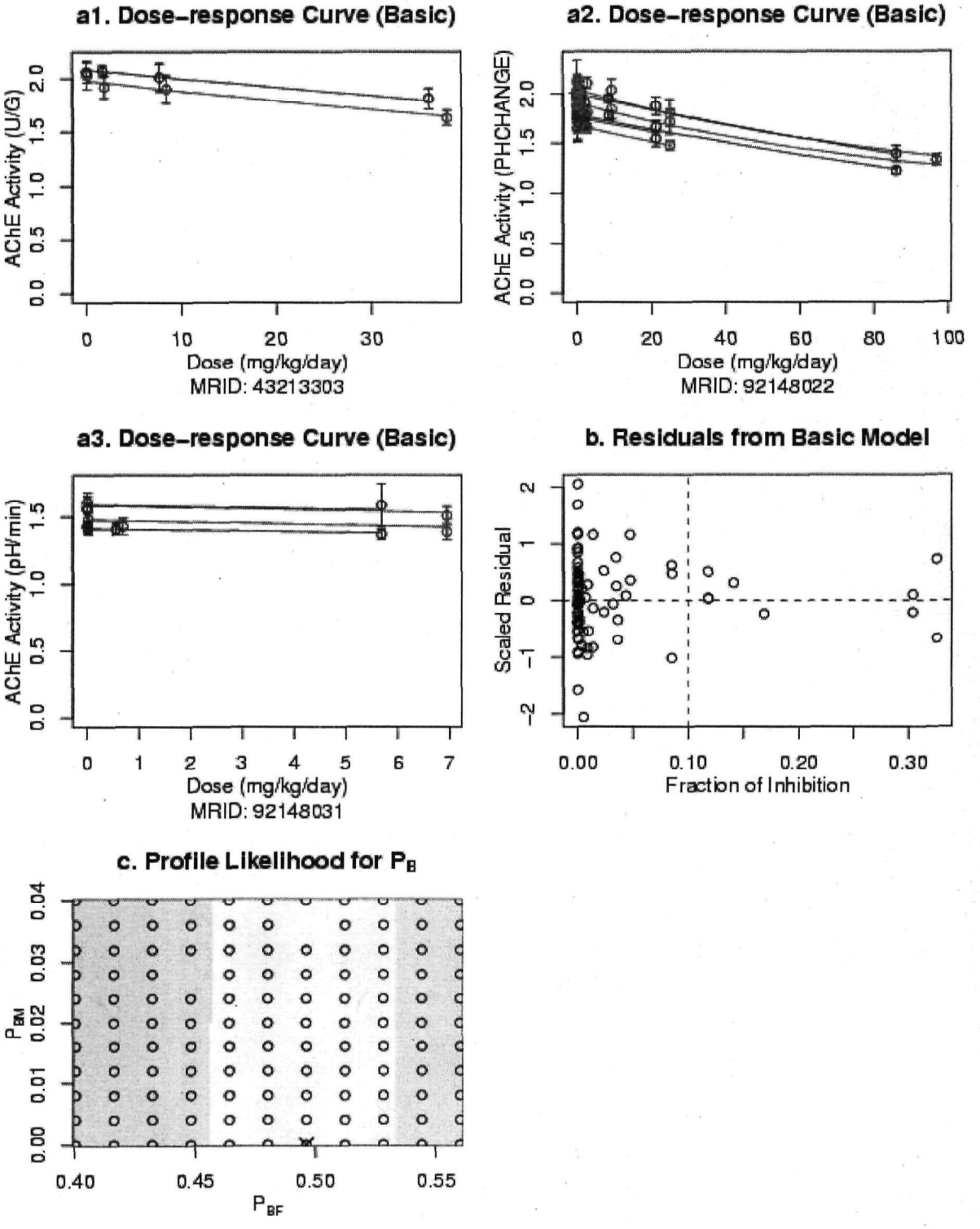


Table III.B.2-30. Terbufos: Toxicology Profile Table

Terbufos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
00109446	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	002377 005612	0/0, 0.01/0.01, 0.02/0.02, 0.05/0.04, 0.095/0.08 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
40089602	83-1 (870.4100)	Chronic Oral Toxicity–Rat	006352	0/0, 0.009/0.007, 0.04/0.03, 0.07/0.06 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley
00049236	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity–Rat	004898 003847 001514 005612 006352	0/0, 0.01/0.01, 0.05/0.04, 0.22/0.33 mg/kg/day (females/males)	Guideline	Rat/ Long Evans
44842302	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	013572	0/0, 0.04/0.04, 0.06/0,06, 0.25/0.37 mg/kg/day (females/males)	Guideline	Rat/ Sprague Dawley

Figure III.B.2-30. Terbufos: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

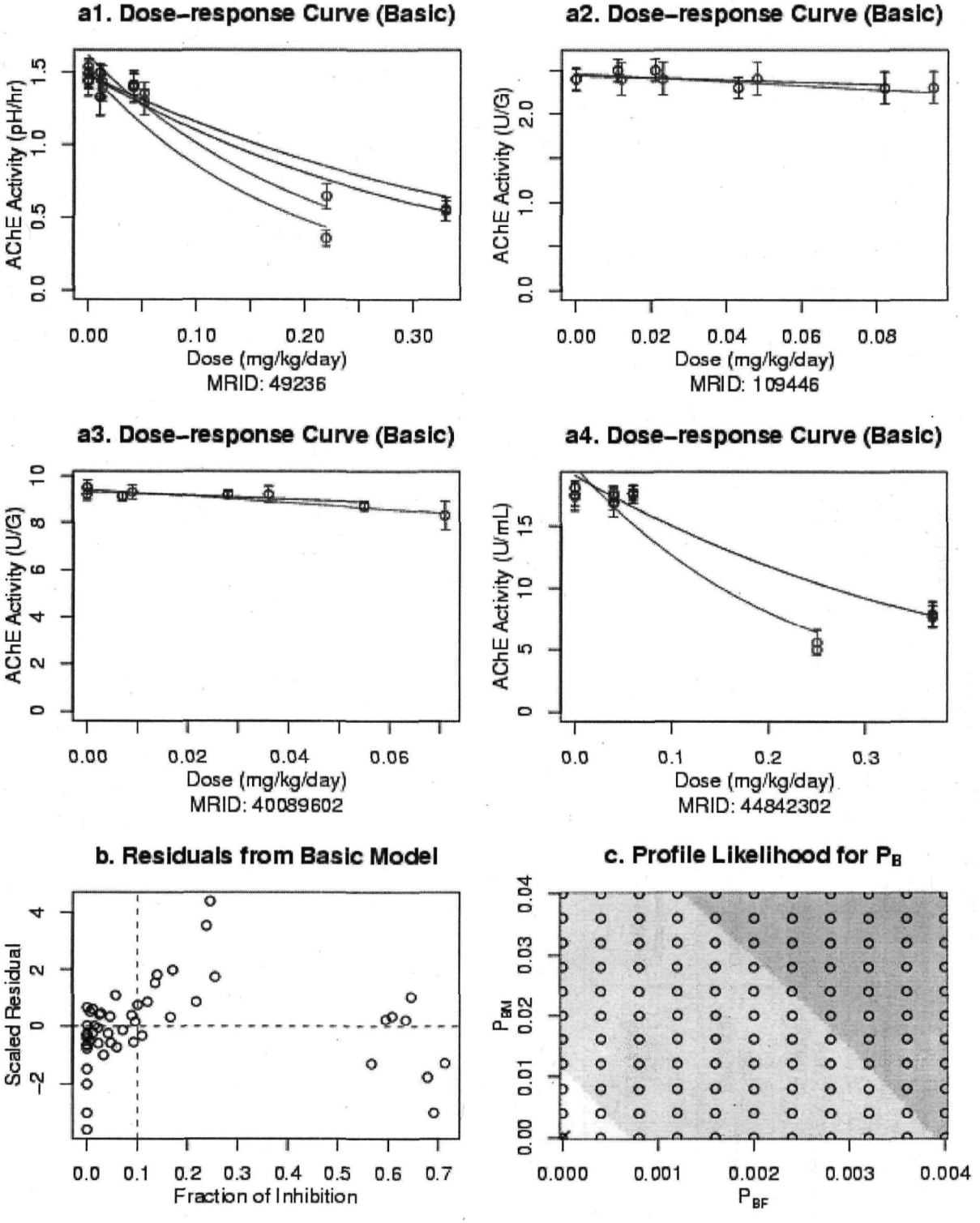


Figure III.B.2-30. Terbufos con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

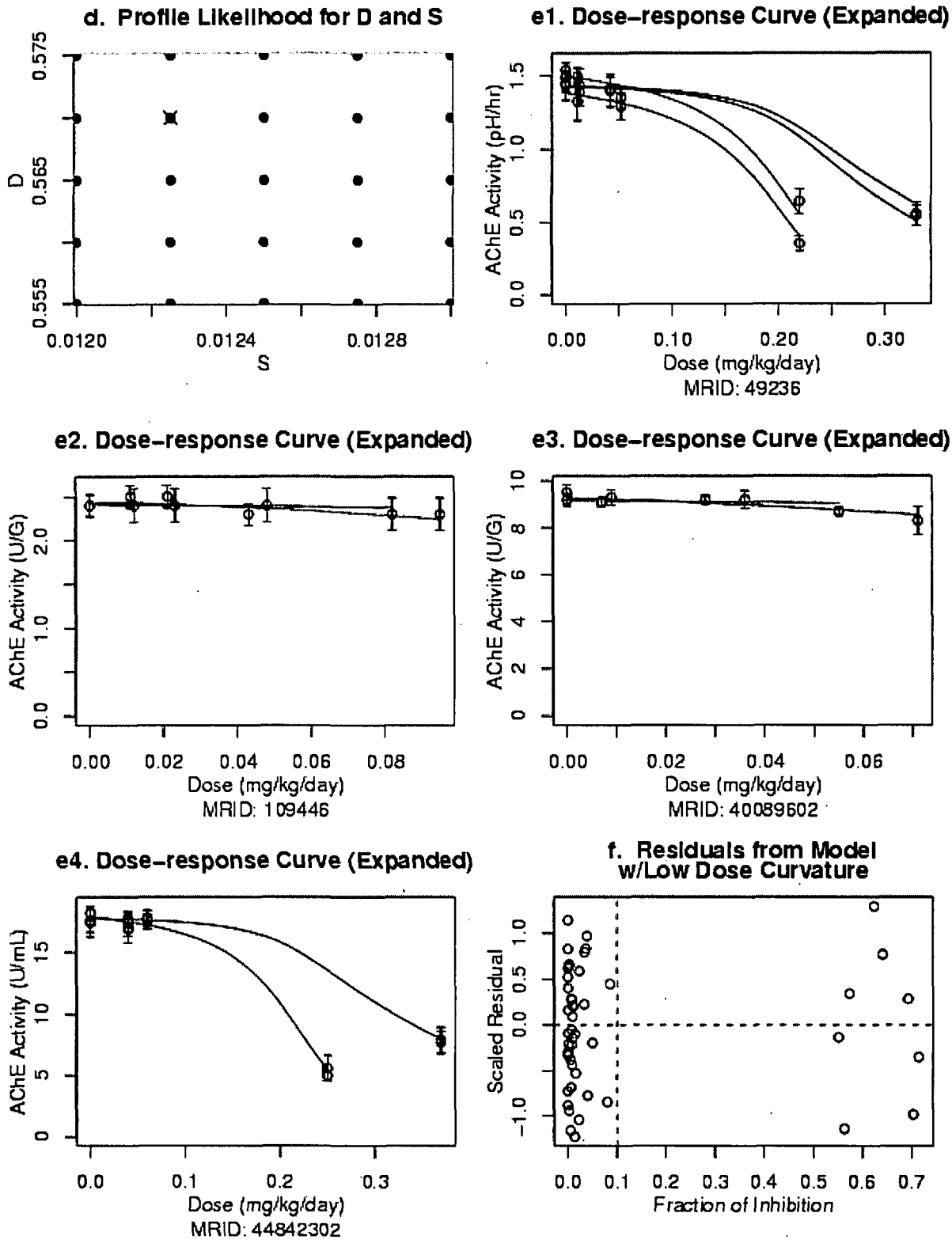


Table III.B.2-31. Tetrachlorvinphos: Toxicology Profile Table

Tetrachlorvinphos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
43371201	82-1 (870.3100)	Subchronic Oral Toxicity–Rat	11295	0, 5, 100, 250 mg/kg/day	Guideline	Rat/Sprague Dawley
00112525	83-2 (870.4200)	Chronic Oral Toxicity–Rat	002607 007181	0, 0.25, 1.25, 6.25, 100 mg/kg/day	Guideline	Rat/ Porton strain derived from Turnstall Lab
42980901	83-2 (870.4200)	Chronic Oral Toxicity–Rat	010884 010884 011295	0/0, 5.93/4.23, 62.7/43.2, 125.3/88.5 mg/kg/day (females/males)	Guideline	Rat/Sprague Dawley
45570601	Nonguideline	21-Day Cholinesterase Study–Rat	TXR No. 0050614	0, 8, 12, 50 mg/kg/day	Acceptable	Rat/Crl:CD®(SD)IGS BR
41342001	82-2 (870.3200)	21-Day Dermal Toxicity–Rat	7844	0, 10, 100, 1000 mg/kg/day	Guideline	Rat/Sprague Dawley

Figure III.B.2-31. Tetrachlorvinphos: Dose-response Curves Using the Basic Model, Plot of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plot for P_B

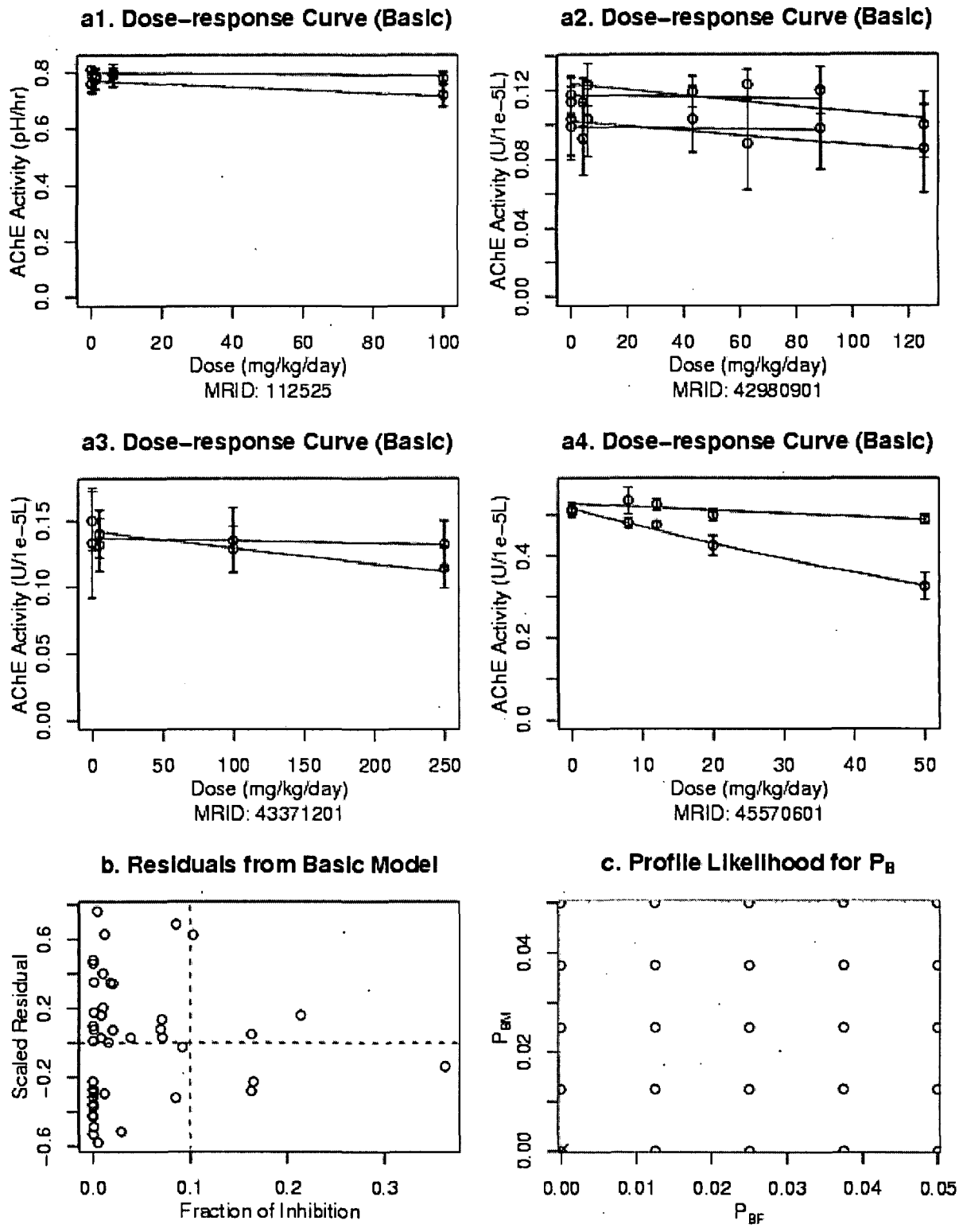


Table III.B.2-32. Tribufos: Toxicology Profile Table

Tribufos						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
42335101	83-5 (870.4300)	Combined Chronic Oral Toxicity/Carcinogenicity/Neurotoxicity–Rat	010119	0/0, 0.2/0.2, 2.3/1.8, 21.1/16.8 mg/kg/day (females/males)	Guideline	Rat/Fischer
45369101	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	NA	0/0, 0.17/0.14, 3.54/2.89, 46.2/36.8 mg/kg/day (females/males)	NA	Rat/ Wistar

NA=Not available

Figure III.B.2-32. Tribufos: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

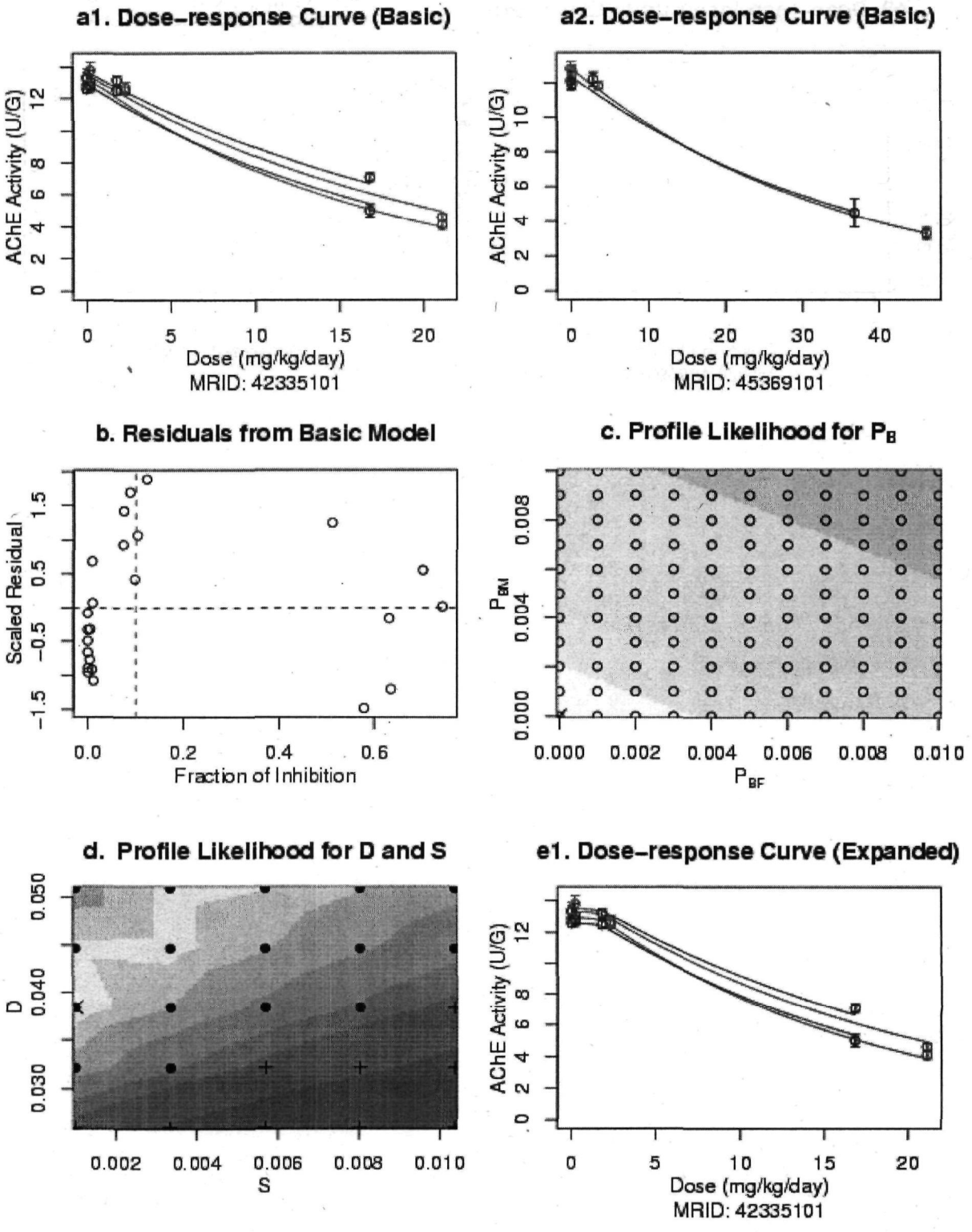


Figure III.B.2-32. Tribufos con't: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

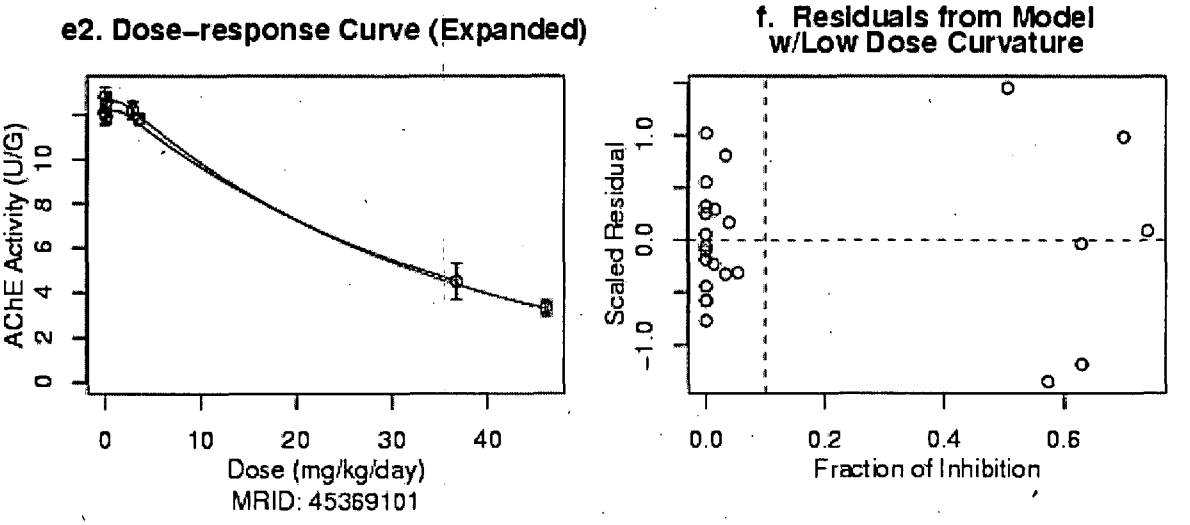


Table III.B.2-33. Trichlorfon: Toxicology Profile Table

Trichlorfon						
MRID #	Guideline No.	Study Type	HED Doc. No.	Dose	Guideline/ Nonguideline	Species/ Strain
43871701	82-7 (870.6200)	Subchronic Neurotoxicity–Rat	13967	0/0, 6.9/6.1, 35.4/31.2, 188.7/164.7 mg/kg/day (females/males)	Guideline	Rat/ Fischer
41056201	83-5 (870.4300)	Combined Chronic Oral Toxicity/ Carcinogenicity–Rat	9626	0/0, 5.8/4.5, 17.4/13.3, 109.2/85.7 mg/kg/day (females/males)	Guideline	Rat/ Fischer
41973001	83-5 (870.4300)	Combined Chronic Oral Toxicity/ Carcinogenicity–Rat	013703	0/0, 159/129 mg/kg/day (females/males)	NA	Rat/ Fischer
40306901	82-2 (870.3200)	21-Day Dermal Toxicity–Rabbit	6476	0, 100, 300, 1000 mg/kg/day	Guideline	Rabbit/New Zealand
00152137	82-4 (870.3465)	21-Day Inhalation Toxicity–Rat	004509 004915	0 (EtON/PEG), 12.7, 35.4, 103.5 mg/m^3	Guideline	Rat/ Wistar

NA=Not available

Figure III.B.2-33. Trichlorfon: Dose-response Curves Using the Basic and Expanded Models, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S

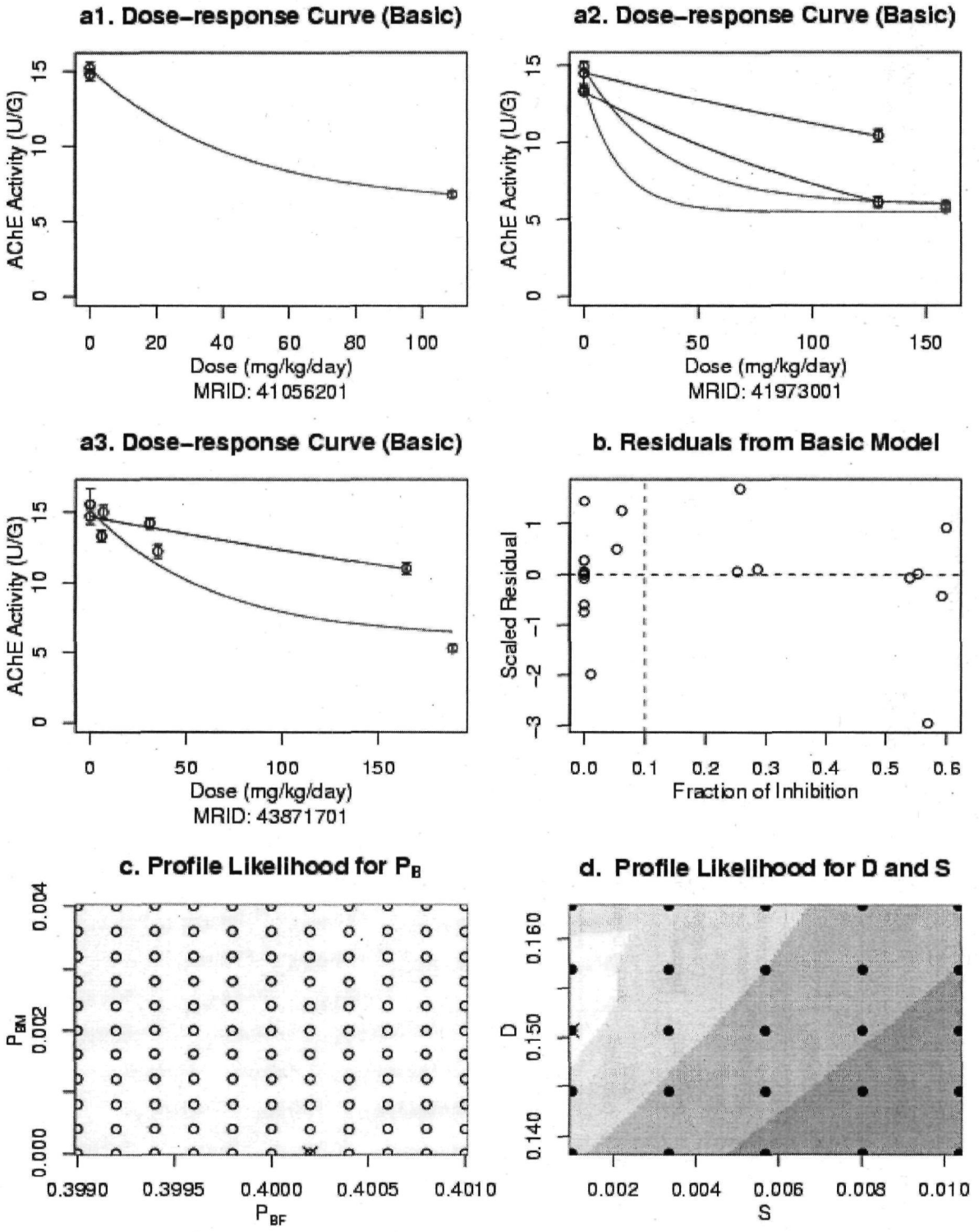
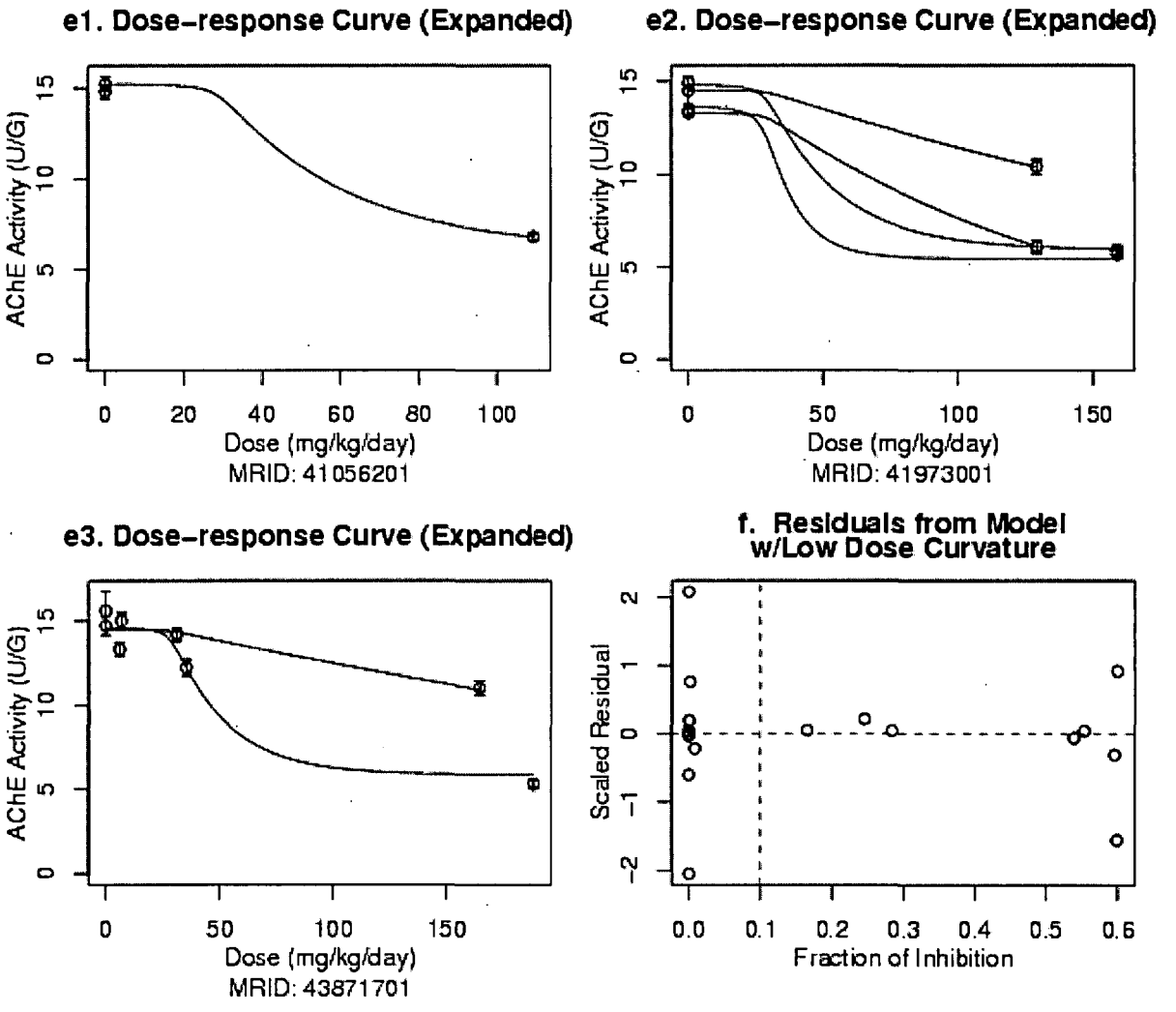


Figure III.B.2-33. Trichlorfon con't: Dose-response Curves Using the Basic Model, Plots of the Scaled Residuals Versus Predicted Inhibition, and the Profile Likelihood Plots for P_B, D, and S



III. Appendices

B. Hazard/RPF

3. Response to SAP Comments from September 2001 and March 2002 Reports

a. Response to SAP Comments from September 2001

OPP in collaboration with ORD presented its July 31st , 2001 document entitled, "Determination of Relative Potency and Points of Departure for Cholinesterase Inhibition" to the FIFRA SAP on September 5-6, 2001. The key recommendations from the September 2001 report (http://www.epa.gov/scipoly/sap/index.htm) and OPP's responses are given below:

i. Derivation of the Adjustment Factor "B" and Modification of Decision Tree for use of "B"

The SAP Report noted that a plot of the "scaled residuals" against "predicted % inhibition" indicates that the weighting strategy used for calculating the adjustment factor "B" does not adequately reflect how the variance changes with response. The SAP was specifically concerned EPA "focused the modeling effort on achieving fidelity with observations at the high end of the range of doses tested, to the likely detriment of fitting points at the low end of the dose response relationship."

In the current analysis, all available cholinesterase datasets for the brain compartment were analyzed using a fixed horizontal y-asymptote for each chemical. The weight function was changed from one in which the variance was presumed proportional to the square of the mean to one in which the variance is proportional to the mean. The revised methodology for the determination of the horizontal y- asymptote is described in I.B and III.B.1.

ii. Conduct a Formal Analysis of Residuals as a Function of Dose

Residual plots for the basic and expanded models for each chemical for the brain compartment are given in III.B.2.

iii. Accuracy of the "Chi Square Approximation" for the "Goodness of Fit" Statistic

In the July 31st document, a Chi-Square Approximation was calculated for each cholinesterase dataset. This statistic was used as a measure of the goodness-of-fit for the exponential function. The concern expressed by the SAP does not apply to the current methodology. Although the OPCumRisk program was not used to determine potency of OPs in the current analysis, the program was revised to deliver a warning message to the program user indicating possible calculation inaccuracy for this statistic. The revised version of the OPCumRisk is available for download at http://www.epa.gov/scipoly/sap/index.htm and http://www.epa.gov/pesticides/cumulative/

iv. Confidence Interval Calculations

The SAP report suggested that HED "reconsider the confidence interval calculations"and "perhaps try bootstrapping or some other more robust method . . . ." In the current analysis, HED has revised the calculation of the confidence intervals (See III.B.1). Bootstrapping is a very time and resource-intensive procedure. Although bootstrapping may be the preferred approach for calculating confidence intervals, due to limited availability of resources, the Agency has not conducted any bootstrapping procedures. At this time, the current method for calculating confidence intervals is adequate and satisfactory. Because it is important to evaluate the range of uncertainty around any potency or benchmark dose values used to extrapolate to human risk, the Agency will consider bootstrapping procedures in future assessments.

v. Deleting p- and t- values

The SAP Report recommended deleting the p- and t- values that are produced by the Agency's OPCumRisk program. As stated previously, the OPCumRisk program was not used in the current analysis to calculate potency or benchmark dose estimates. The requested deletions have been incorporated; the revised version of the OPCumRisk is available for download at http://www.epa.gov/scipoly/sap/index.htm and http://www.epa.gov/pesticides/cumulative/

vi. Estimates of Relative Potency

The SAP Report included considerable discussion regarding whether relative potency factors should be based on ratios of the "Benchmark Dose 10's" (BMD_{10}) or on ratios of the dose-scaling factors. OPP has derived potency in the present analysis on BMD_{10} (See I.B).

vii. Inhalation Dose

The SAP Report recommended that inhalation exposure be expressed in the same units as the oral doses and that the doses be adjusted for actual treatment durations. HED has calculated the inhalation doses as mg/kg/day using conversion factors that account for respiratory volume and body weight for the strain of rat used, as well as the duration of exposure in terms of hours exposed per day.

viii. Use of Individual Animal Data

The SAP Report from the September 2000 SAP meeting recommended that study data on individual animals be used in calculating relative potencies. Due to the fact that all the data on organophosphates are not in an electronic format, HED has not taken this step. However, the September, 2001 Report recognizes that "individual data would not be likely to change the results using current methods." In addition, by switching from RBC to the brain compartment, some of the concern about not using individual animal data should be reduced, since the experimental designs for the brain measurements do not include a repeated measures component, unlike the RBC data.

iv. Use of NOAEL's and LOAEL's for Inhalation and Dermal Routes

Several Panel members objected to EPA's use of No Observed Adverse Effect Levels ("NOAEL's") and Lowest Observed Adverse Effect Levels ("LOAEL's") for cholinesterase inhibition data by the dermal and inhalation routes of exposure instead of actual dose-response models as are used for the oral data set. HED does not intend to use dose-response modeling to determine relative potency estimates for dermal and inhalation exposure because the data are not sufficiently robust to justify the resources required.

However, it is to be noted that the current analysis uses Comparative Effect Levels (CEL's) for cholinesterase inhibition data for these two routes of exposure. The dermal and inhalation database was not suitable for dose-response analysis. Cholinesterase determinations in these studies were typically made at only one time point and several of the studies had no cholinesterase inhibition at the highest dose. For the current assessment, potencies by the dermal and inhalation routes were compared using brain cholinesterase inhibition at a dose causing a maximum of 15% brain cholinesterase inhibition.

v. Derivation of Doses from the Actual Dietary Intake Rates

The SAP Report recommends that "the doses used for evaluation of potencies at various ages within specific data sets should be derived from the actual dietary intake rates observed in the study for those ages where the consumption data are available."

In feeding toxicity studies, laboratory rats are exposed to the test compound via the diet. Generally, the test compound is mixed in the animal feed which the laboratory animals eat. Over the course of a toxicity study, as the animals age, they will not only gain weight and but they will naturally change their rate of food consumption. The data collected for the oral route and used in both the July and December 2001 preliminary cumulative risk assessments include average compound intake (mg of active ingredient per kg per day). HED has conducted a pilot analysis in response to this recommendation to evaluate the effect of age and food consumption rate on the potency estimates. In this pilot compound intake analysis, OP potency was determined for a subset of studies [≈10% of total studies in the dose-response assessment] using compound intake measured at or around the time of cholinesterase measurements [duration-specific compound intake].

Seventy-nine oral toxicity studies were included in the dose-response assessment for the December, 2001 Cumulative Risk Assessment for OPs. Of these 79 studies, the test article was administered via the diet for 73. For each of the seven OPs selected for this analysis, the calculated compound intake (mg/kg/day) given in the study report for a weekly, biweekly, or monthly time interval closest to the time of cholinesterase measurement was extracted from the feeding toxicity studies [duration-specific compound intakes]. For example, if brain cholinesterase was measured at a one-year interim sacrifice, the compound intake for the 50-52 week reported interval was collected. The potency values obtained were compared to those in the July, 2001 analysis, which utilized average compound intake values. Potency estimates given below (Table III.B.3-4) were calculated using the OPCumRisk program with the methodology described in the July 31st document *prior* to the completion of the current methodology for the joint analysis. The pilot analysis was performed in three stages : 1) impact of age on relative potency for chronic studies only; 2) impact of age on relative potency for complete database of subchronic and chronic studies; and 3) impact of age on the points of departure on the index chemical.

Stage 1: The purpose of this pilot analysis was to investigate the impact of age on food consumption and body weight, and ultimately OP potency. In order to maximize the age-related differences in body weight and food consumption, chronic studies were analyzed first. Seven chronic feeding studies were selected

randomly and analyzed as described above. Relative potency of each was calculated using the methamidophos chronic study. Results given in Table III.B.3-1.

In the chronic study analysis (Table III.B.3-1) comparing the RPFs calculated using the slope scale factor (m) and also the BMD_{10}s for ChE data using the average and duration-specific compound intakes, *the RBC and brain data for both sexes display comparable potency values.* For tribufos a 5-fold difference between the average and duration-specific intake assessments for male brain CHel was observed. This difference is an artifact of the decision tree for the determination B (horizontal asymptote) and not from differences in potency between the average and duration specific intakes. Two timepoints (364 and 721 days) are available for the male brain ChE data in MRID 42335101. In the duration specific analysis, the 364 day time point did not converge and was therefore not included in the potency estimates.

Table III.B.3-1a. Results of Dietary Intake Comparison [actual *vs* average] Using Chronic Studies

CHEMICAL	MRID	COMPARTMENT	SEX	Dietary Intake Calculation	Relative Potency using 'm'	Lower 95% CL	Upper 95% CL	BMD_{10}	BMDL	Relative Potency using BMD_{10}
BENSULIDE	44161101	BRAIN	F	average	0.005	0.004	0.006	14.11	12.40	0.005
				biweekly	0.004	0.004	0.005	14.04	12.17	0.004
DIAZINON	41942002	BRAIN	F	average	0.034	0.031	0.038	1.85	1.78	0.038
				biweekly	0.031	0.028	0.035	1.85	1.80	0.034
DICROTOPHOS	44527802	BRAIN	F	average	1.77	1.41	2.22	0.041	0.035	1.74
				biweekly	1.89	1.51	2.38	0.035	0.030	1.79
METHAMIDOPHOS	00148452	BRAIN	F	average	1.00	1.00	1.00	0.071	0.063	1.00
				biweekly	1.00	1.00	1.00	0.063	0.058	1.00
PHOSALONE	44801002	BRAIN	F	average	0.015	0.013	0.018	4.13	3.70	0.017
				biweekly	0.024	0.020	0.029	2.40	2.14	0.026
PHOSMET	41916401	BRAIN	F	average	0.023	0.010	0.053	4.41	3.74	0.016
				biweekly	0.021	0.016	0.027	2.76	2.33	0.023
TRIBUFOS	42335101	BRAIN	F	average	0.018	0.007	0.048	3.26	1.88	0.022
				biweekly	0.017	0.007	0.045	3.14	1.83	0.020
BENSULIDE	44161101	BRAIN	M	average	0.002	0.002	0.003	24.69	19.37	0.003
				biweekly	0.002	0.001	0.003	24.93	19.54	0.002
DIAZINON	41942002	BRAIN	M	average	0.011	0.003	0.041	3.38	1.83	0.018
				biweekly	0.011	0.003	0.035	3.31	1.83	0.016
DICROTOPHOS	44527802	BRAIN	M	average	2.06	1.70	2.38	0.028	0.026	2.23
				biweekly	2.32	2.03	2.67	0.022	0.020	2.45
METHAMIDOPHOS	00148452	BRAIN	M	average	1.00	1.00	1.00	0.062	0.057	1.00
				biweekly	1.00	1.00	1.00	0.055	0.049	1.00
PHOSALONE	44801002	BRAIN	M	average	0.021	0.018	0.025	2.58	2.37	0.024
				biweekly	0.038	0.033	0.044	1.29	1.18	0.042
PHOSMET	41916401	BRAIN	M	average	0.011	0.008	0.015	5.35	4.33	0.012
				biweekly	0.013	0.009	0.018	3.71	2.98	0.015
TRIBUFOS	42335101	BRAIN	M	average	0.020	0.017	0.022	4.22	2.51	0.015
				biweekly	0.004	0.001	0.020	15.64	6.19	0.003

Table III.B.3-1b. Results of Dietary Intake Comparison [actual *vs* average] Using Chronic Studies

CHEMICAL	MRID	COMPARTMENT	SEX	Dietary Intake Calculation	Relative Potency using 'm'	Lower 95% CL	Upper 95% CL	BMD_{10}	BMDL	Relative Potency using BMD_{10}
BENSULIDE	44161101	RBC	F	average	0.012	0.005	0.025	5.53	3.69	0.012
				biweekly	0.011	0.005	0.024	5.35	3.55	0.012
DIAZINON	41942002	RBC	F	average	0.12	0.037	0.38	0.28	0.17	0.24
				biweekly	0.11	0.036	0.33	0.29	0.18	0.21
DICROTOPHOS	44527802	RBC	F	average	2.77	1.88	4.08	0.039	0.030	1.71
				biweekly	2.89	1.95	4.29	0.035	0.027	1.78
METHAMIDOPHOS	00148452	RBC	F	average	1.00	1.00	1.00	0.067	0.063	1.00
				biweekly	1.00	1.00	1.00	0.062	0.058	1.00
PHOSALONE	44801002	RBC	F	average	0.068	0.027	0.17	0.71	0.48	0.094
				biweekly	0.076	0.035	0.17	0.64	0.44	0.097
PHOSMET	41916401	RBC	F	average	0.080	0.058	0.11	0.84	0.75	0.080
				biweekly	0.083	0.065	0.11	0.70	0.57	0.089
TRIBUFOS	42335101	RBC	F	average	0.095	0.048	0.19	0.61	0.48	0.11
				biweekly	0.089	0.045	0.18	0.60	0.46	0.10
BENSULIDE	44161101	RBC	M	average	0.013	0.006	0.026	7.56	6.34	0.008
				biweekly	0.013	0.006	0.027	7.55	6.33	0.007
DIAZINON	41942002	RBC	M	average	0.040	0.013	0.13	2.36	1.92	0.025
				biweekly	0.042	0.013	0.13	2.09	1.57	0.025
DICROTOPHOS	44527802	RBC	M	average	1.33	1.10	1.61	0.039	0.035	1.51
				biweekly	1.55	1.26	1.91	0.033	0.030	1.60
METHAMIDOPHOS	00148452	RBC	M	average	1.00	1.00	1.00	0.059	0.056	1.00
				biweekly	1.00	1.00	1.00	0.053	0.047	1.00
PHOSALONE	44801002	RBC	M	average	0.053	0.021	0.13	0.96	0.56	0.062
				biweekly	0.067	0.032	0.14	1.49	1.31	0.035
PHOSMET	41916401	RBC	M	average	0.079	0.055	0.11	0.81	0.72	0.073
				biweekly	0.10	0.077	0.14	0.58	0.53	0.091
TRIBUFOS	42335101	RBC	M	average	0.14	0.090	0.21	0.49	0.40	0.12
				biweekly	0.10	0.050	0.21	0.57	0.42	0.094

Stage 2: Out of the seven OPs analyzed in Stage 1, the entire oral databases; i.e., both chronic and subchronic studies, of three randomly selected OPs were analyzed as in Stage 1. Relative potency was calculated using all available methamidophos studies (Table III.B.3-2).

In the pilot analysis of the complete oral database for three OPs (diazinon, dimethoate, and phosalone; Table III.B.3-2) comparing the RPFs calculated with slope scale factors and BMD_{10}s for ChE data using the average and duration-specific compound intakes, *the RBC and brain data for both sexes display comparable potency values*. For phosalone RBC male *only*, a 7-fold difference between the average and duration-specific intake assessments was observed.

Graphs of potency vs. time are shown in Figures III.B.3-1,2 for the analyzes of average chemical intake and for duration specific chemical intake. The patterns observed in the graphs for the average intake analyzes are similar to those of the duration specific intakes.

Table III.B.3-2. Results of Dietary Intake [actual *vs* average] Using All Available Studies

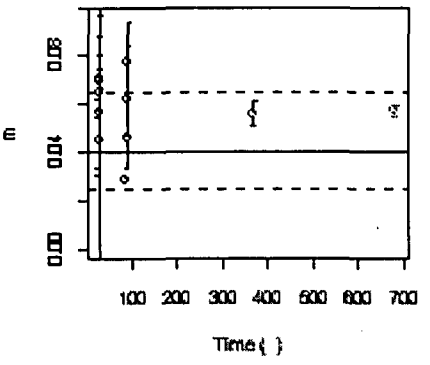
CHEMICAL	MRID	COMPARTMENT	SEX	Dietary Intake Calculation	Relative Potency using 'm'	Lower 95% CL	Upper 95% CL	BMD_{10}	BMDL	Relative Potency using BMD_{10}
DIAZINON	43543901 43543902 40815003 41942002	BRAIN	F	average	0.031	0.018	0.053	2.48	1.78	0.036
				biweekly	0.033	0.019	0.058	2.08	1.51	0.038
DIMETHOATE	43128201 164177	BRAIN	F	average	0.531	0.41	0.69	0.25	0.23	0.36
				biweekly	0.58	0.45	0.75	0.20	0.18	0.40
METHAMIDOPHOS	41867201 00148452 43197901	BRAIN	F	average	1.00	1.00	1.00	0.09	0.08	1.00
				biweekly	1.00	1.00	1.00	0.08	0.07	1.00
PHOSALONE	44852504 44801002	BRAIN	F	average	0.019	0.014	0.025	5.05	3.83	0.018
				biweekly	0.021	0.010	0.040	3.37	2.24	0.024
DIAZINON	43543901 43543902 40815003 41942002	BRAIN	M	average	0.005	0.002	0.012	24.77	24.15	0.003
				biweekly	0.005	0.002	0.010	18.28	17.83	0.004
DIMETHOATE	43128201 164177	BRAIN	M	average	0.71	0.53	0.94	0.10	0.08	0.80
				biweekly	0.83	0.60	1.15	0.08	0.06	0.88
METHAMIDOPHOS	41867201 148452 43197901	BRAIN	M	average	1.00	1.00	1.00	0.08	0.07	1.00
				biweekly	1.00	1.00	1.00	0.07	0.06	1.00
PHOSALONE	44852504 44801002	BRAIN	M	average	0.019	0.011	0.032	3.49	2.49	0.023
				biweekly	0.028	0.012	0.063	1.96	1.22	0.036
DIAZINON	43543901 43543902 40815003 41942002	RBC	F	average	0.38	0.22	0.65	0.24	0.22	0.38
				biweekly	0.41	0.27	0.62	0.18	0.17	0.44
DIMETHOATE	43128201 164177	RBC	F	average	0.32	0.14	0.73	0.29	0.14	0.31
				biweekly	0.27	0.14	0.53	0.33	0.16	0.24

CHEMICAL	MRID	COMPARTMENT	SEX	Dietary Intake Calculation	Relative Potency using 'm'	Lower 95% CL	Upper 95% CL	BMD_{10}	BMDL	Relative Potency using BMD_{10}
METHAMIDOPHOS	41867201 148452 43197901	RBC	F	average	1.00	1.00	1.00	0.09	0.07	1.00
				biweekly	1.00	1.00	1.00	0.08	0.06	1.00
PHOSALONE	44852504 44801002	RBC	F	average	0.044	0.015	0.13	1.45	0.77	0.062
				biweekly	0.048	0.017	0.14	1.31	0.68	0.061
DIAZINON	43543901 43543902 40815003 41942002	RBC	M	average	0.12	0.024	0.63	0.40	0.22	0.18
				biweekly	0.14	0.027	0.68	0.34	0.18	0.18
DIMETHOATE	43128201 164177	RBC	M	average	0.27	0.15	0.48	0.36	0.20	0.19
				biweekly	0.25	0.13	0.47	0.40	0.22	0.15
METHAMIDOPHOS	41867201 148452 43197901	RBC	M	average	1.00	1.00	1.00	0.07	0.05	1.00
				biweekly	1.00	1.00	1.00	0.06	0.05	1.00
PHOSALONE	44852504 44801002	RBC	M	average	0.054	0.022	0.13	18.07	9.81	0.004
				biweekly	0.072	0.032	0.16	2.72	1.40	0.023

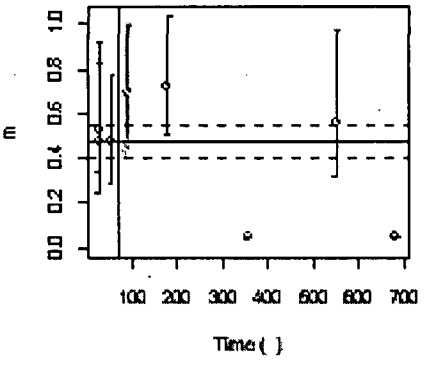
Figure III.B.3-1a. Plots of potency versus time for brain cholinesterase measured in rats exposed to diazinon

Average Dose

Female

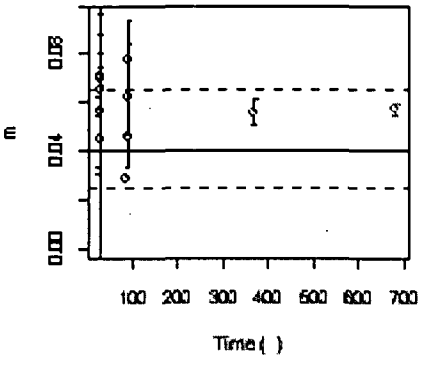


Male



Duration Specific Dose

Female



Male

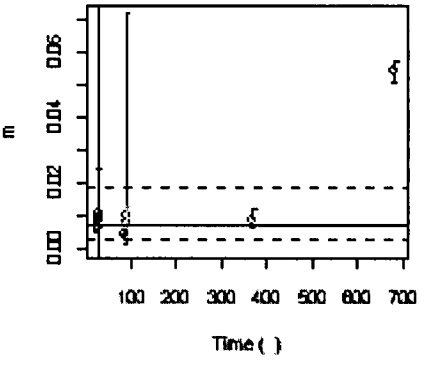
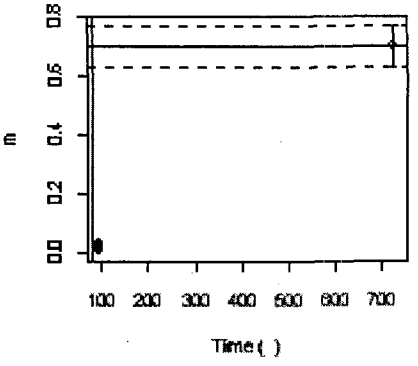


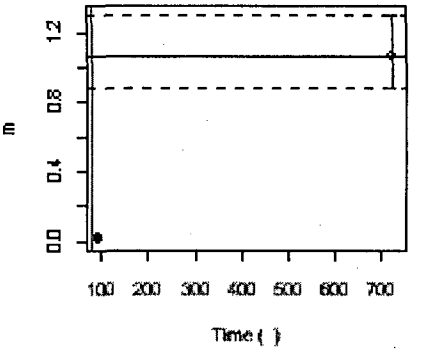
Figure III.B.3-1b. Plots of potency versus time for brain cholinesterase measured in rats exposed to dimethoate

Average Dose

Female

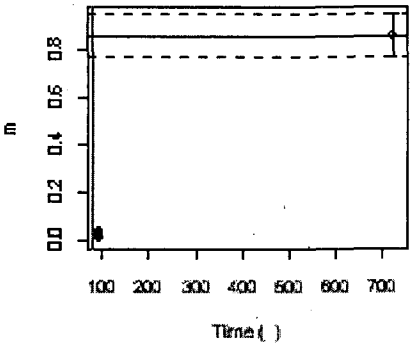


Male



Duration Specific Dose

Female



Male

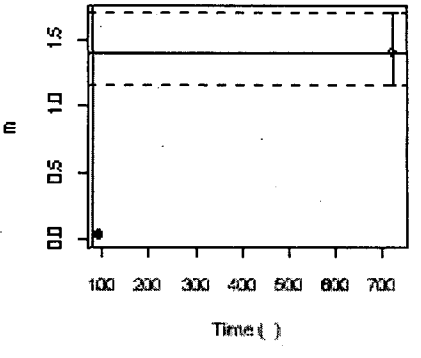
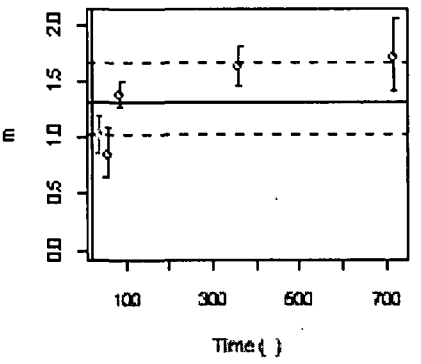


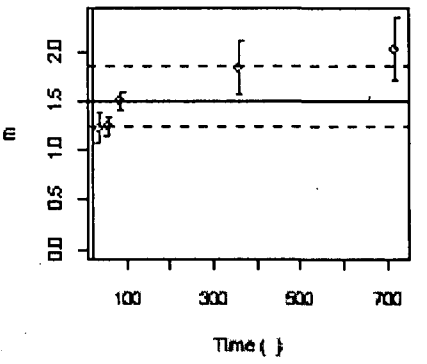
Figure III.B.3-1c. Plots of potency versus time for brain cholinesterase measured in rats exposed to methamidophos

Average Dose

Female

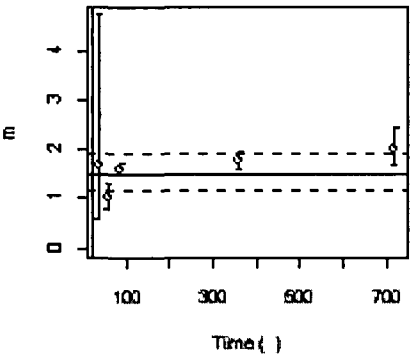


Male



Duration Specific Dose

Female



Male

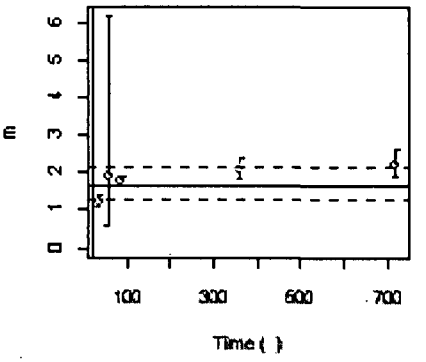
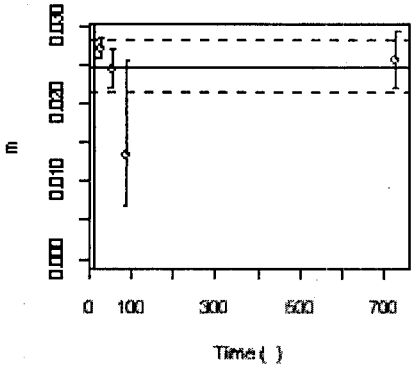


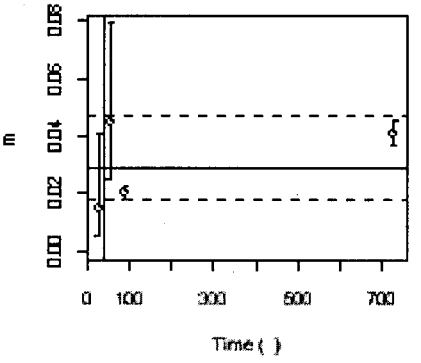
Figure III.B.3-1d. Plots of potency versus time for brain cholinesterase measured in rats exposed to phosalone

Average Dose

Female

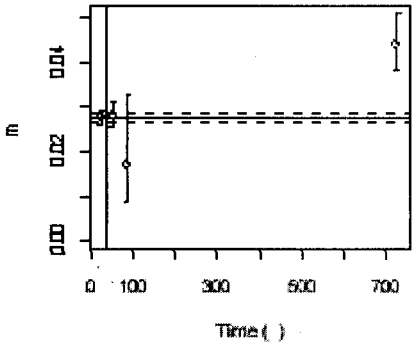


Male



Duration Specific Dose

Female



Male

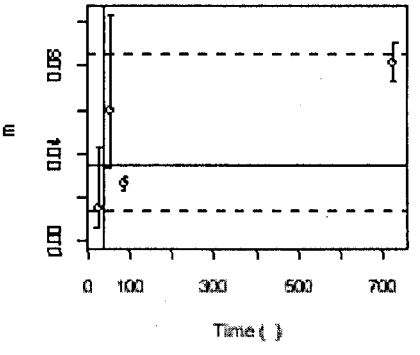
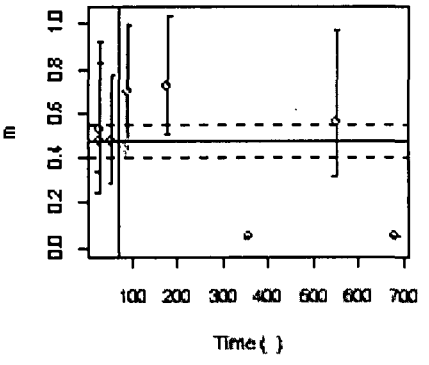


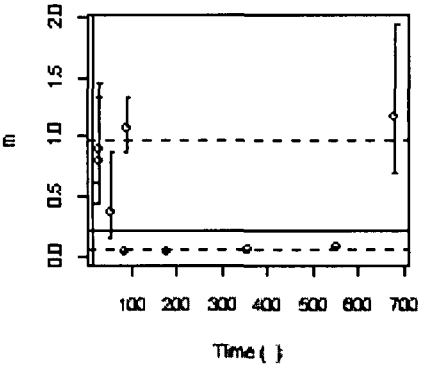
Figure III.B.3-2a. Plots of potency versus time for RBC cholinesterase measured in rats exposed to diazinon

Average Dose

Female

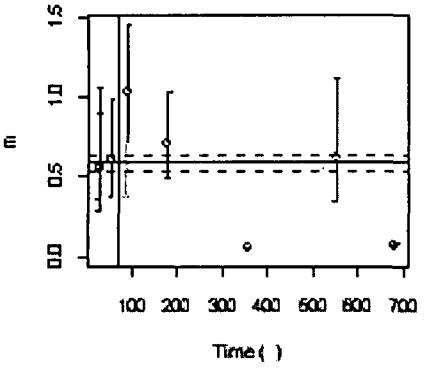


Male



Duration Specific Dose

Female



Male

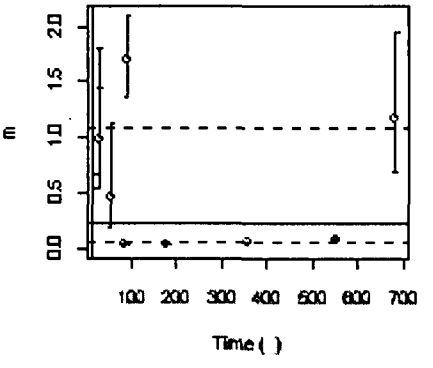
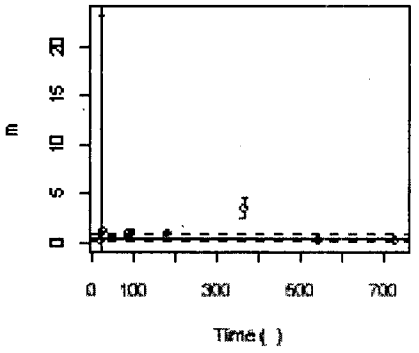


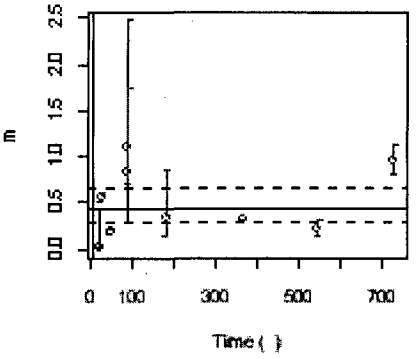
Figure III.B.3-2b. Plots of potency versus time for RBC cholinesterase measured in rats exposed to dimethoate

Average Dose

Female

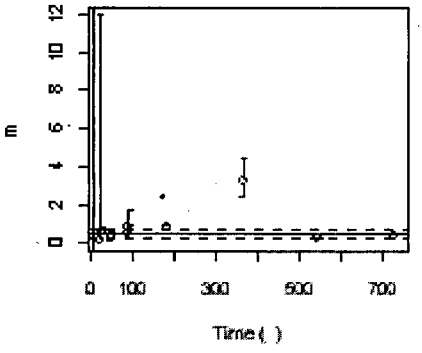


Male



Duration Specific Dose

Female



Male

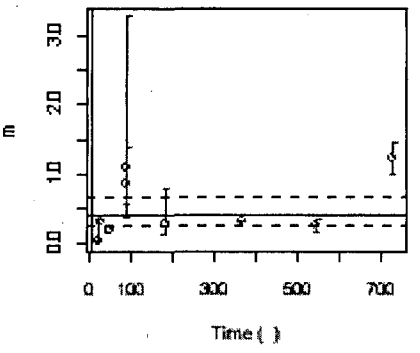
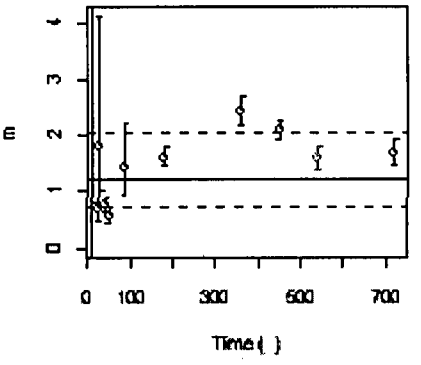


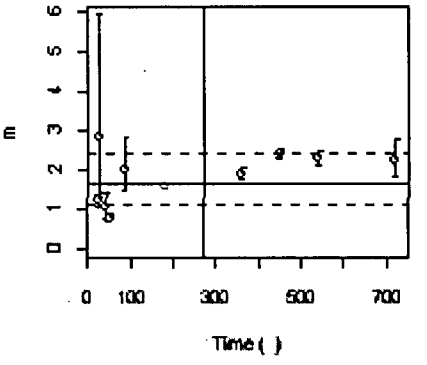
Figure III.B.3-2c. Plots of potency versus time for RBC cholinesterase measured in rats exposed to methamidophos

Average Dose

Female

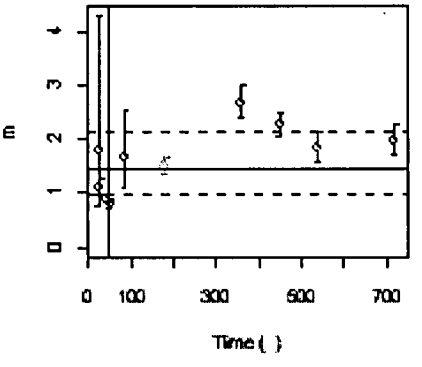


Male



Duration Specific Dose

Female



Male

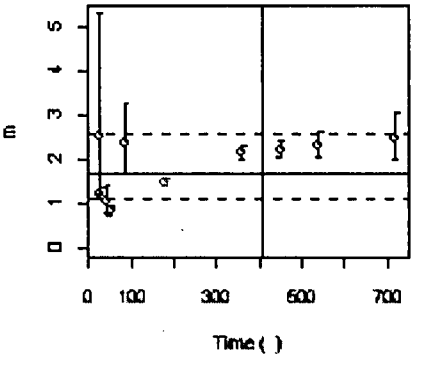
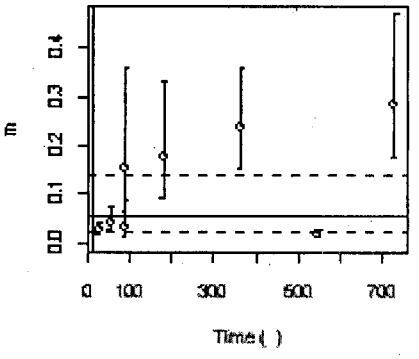


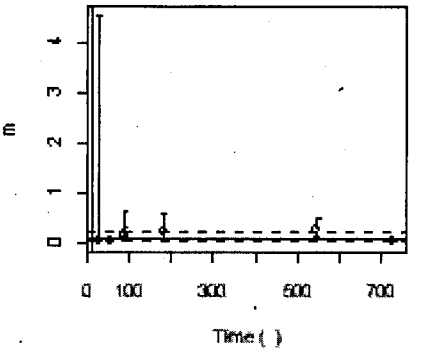
Figure III.B.3-2d. Plots of potency versus time for RBC cholinesterase measured in rats exposed to phosalone

Average Dose

Female

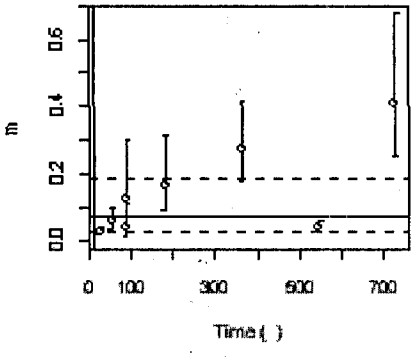


Male

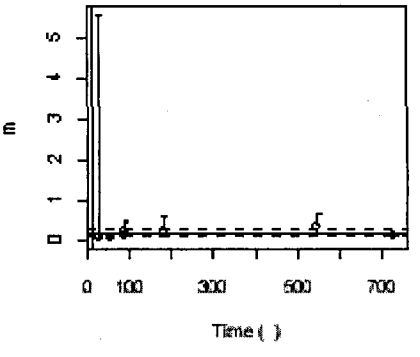


Duration Specific Dose

Female



Male



Stage 3: Compare the BMD_{10} 's and BMDL's of the index chemical calculated from the average compound intakes and the duration-specific compound intakes (Table III.B.3-3).

As shown in Table III.B.3-3, BMD_{10} and BMDL calculated using the average compound intake from July analysis are similar to but slightly smaller those calculated with the July methods with duration-specific compound intakes. BMD_{10} and BMDL calculated using the average compound intake from July analysis are similar those calculated with the December methods with duration-specific compound intakes.

Table III.B.3-3. Comparison of Average Intake *vs* Duration-Specific Intake BMD_{10}s and BMDLs

Compartment Sex	JULY				DECEMBER		
	Average Intake		Duration-Specific Intake		Compartment Sex	BMD_{10}	BMDL
	BMD_{10}	BMDL	BMD_{10}	BMDL			
FEMALE RBC	0.09	0.07	0.08	0.06	FEMALE brain	0.08	0.07
FEMALE brain	0.09	0.08	0.08	0.07			
MALE RBC	0.07	0.05	0.06	0.05	MALE brain	0.07	0.06
MALE brain	0.08	0.07	0.07	0.06			

Conclusions: The pilot analysis of compound intakes using duration specific values showed that relative potency estimates calculated from slope-scaling factors and BMD_{10}s are similar to those calculated using the average study compound intake. Based on this analysis, it is reasonable for OPP to continue using the average compound intake for its potency estimates. Concerning the PODs for the index chemical, although the values are very similar, the PODs calculated from duration-specific intake values result in slightly smaller BMD_{10}s.

b. Response to SAP Comments from March 2002

The following analyses were performed following discussion and recommendations from the February 5-8. 2002 meeting the FIFRA SAP meeting on the "Methods Used to Conduct a Preliminary Cumulative Risk Assessment for Organophosphate Pesticides":

i. Selecting the Benchmark Response Level

At the February 5-8, 2002 meeting of the FIFRA SAP, some panel members and some Public Commenters discussed the Agency's selection

of the BMD_{10} as the benchmark response level. In response to this discussion, the Agency analyzed the detection limits of the studies assessing female brain cholinesterase levels used in the Preliminary Cumulative Risk Assessment of the OPs. This analysis has shown that generally these studies can reliably detect around 10% cholinesterase inhibition and that such levels were generally achieved in the studies. *Therefore, the Agency's use of the BMD_{10} as the benchmark response is appropriate.*

According the Agency's draft benchmark dose guidance (USEPA, 2000a), generally, the response level selected to calculate the benchmark dose should lie in the low end of the range of the responses but within assay detectability. Figure III.B.3-3 shows a plot of the range of mean brain cholinesterase inhibition observed in all treatment groups (i.e., controls were not included). That figure shows that all chemicals include at least one dose level that yields approximately 10% inhibition. Thus, it is possible to directly assess the fit of the model to data in this critical region.

The ability of a study to detect a given amount of change is measured by the power of the study. In general, the power of a study depends on the sample size and the variability of the observations, measured as the standard deviation among individual measurements. Both of these factors vary among datasets in this risk assessment. The power for each study to detect a difference between control and a single treatment group of mean brain cholinesterase activity by 1%, 5%, 7.5%, 10%, 15%, and 20% has been calculated. In Figure III.B.3-4, the proportion of datasets with at least *x* power is plotted against *x* for effect levels ranging from 1-20% inhibition, and the median power (that is, the power level such that half the datasets have greater than that level of power) among those data sets to detect each change is indicated on the axis. Only at the level of a 10% change is the median power greater than 0.80, which has been a conventional goal in designing experiments. Thus, a 10% change in mean cholinesterase activity is indeed in the low end of detectability of assays for brain cholinesterase activity as they were conducted in the studies used in this risk assessment.

Figure III.B.3-3. Observed levels of inhibition relative to concurrent control for all dose-groups. The solid vertical line indicates 10% inhbition.

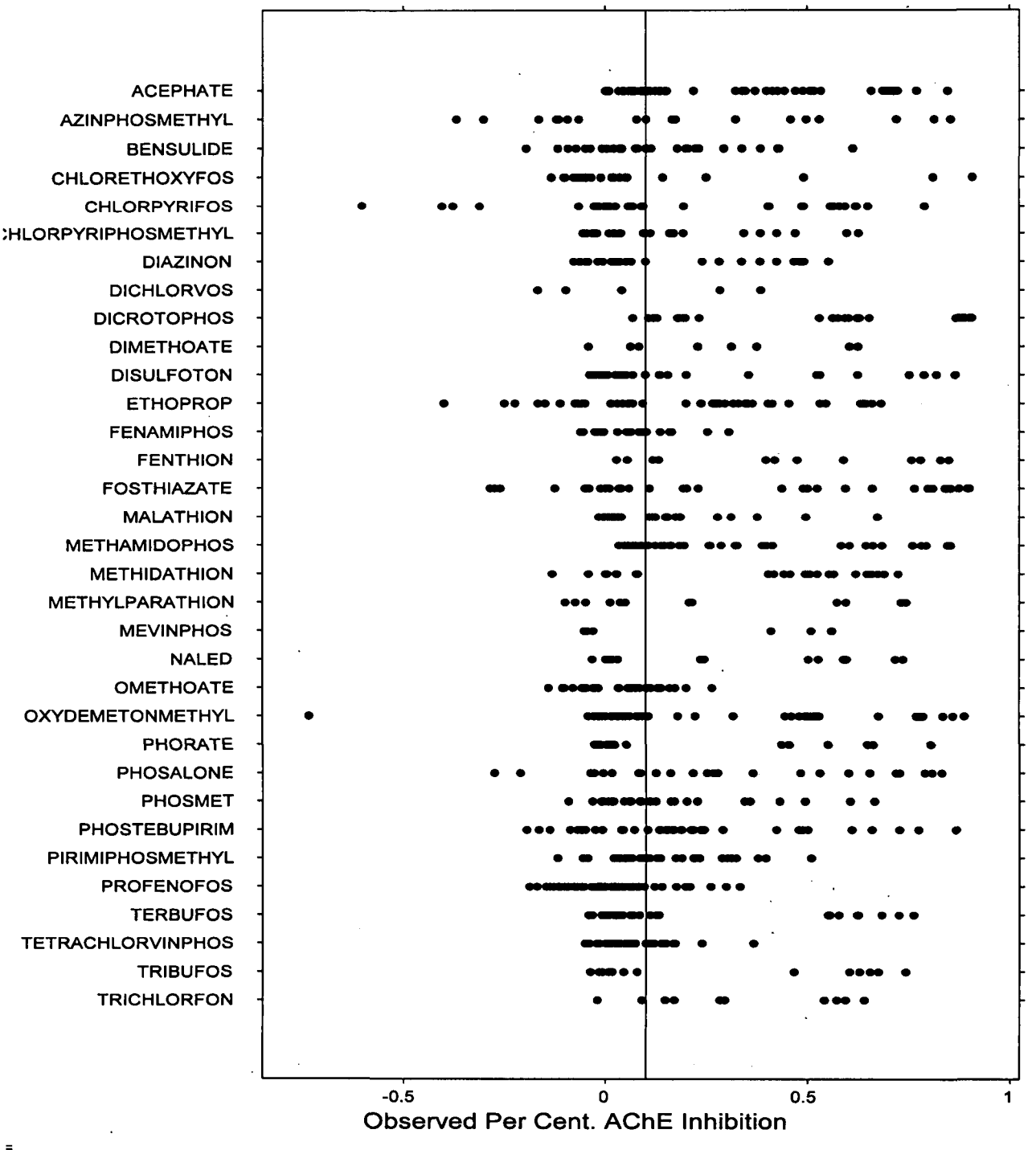
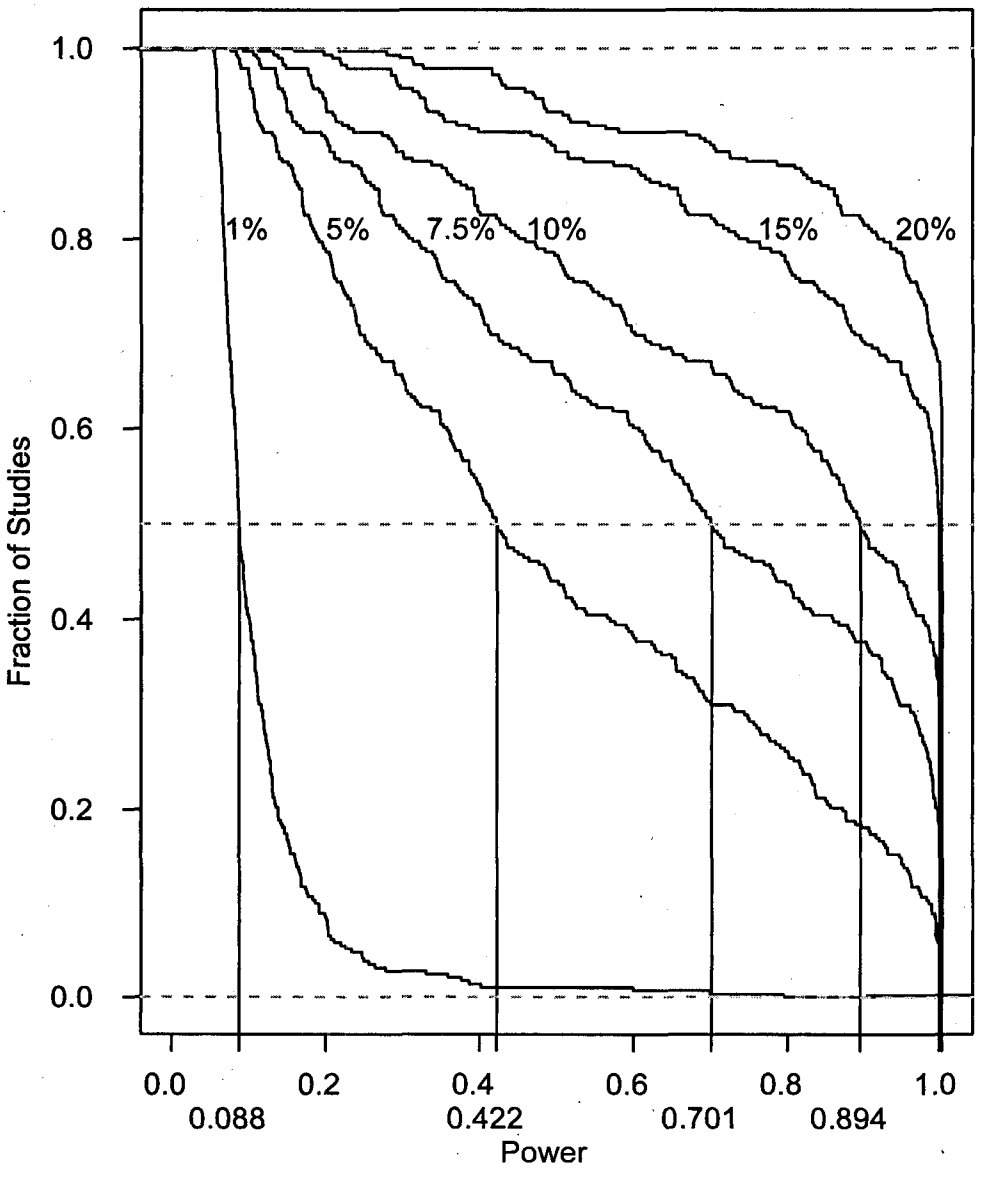


Figure III.B.3-4. Distribution of the power to detect a 1%, 5%, 7.5%, 10%, 15%, and 20% change in mean cholinesterase activity among datasets in the risk assessment. For each effect of treatment, the curves represent the fraction of datasets for which the power is at least the value on the x-axis to detect that effect. For example, half the studies have at least a power of 0.894 to detect a 10% change in mean cholinesterase activity.



ii. Standard and formal definition of the full mathematical exponential model

A formal presentation of the exponential model is included in the Appendix III.B.1.

iii. Individual Animal Data: Consequences of Aggregating Data

At the February 5-8, 2002 meeting of the FIFRA SAP, some members of the panel discussed the fact that the dose-response modeling of cholinesterase inhibition was based solely on dose group means, standard deviations, and sample sizes. The discussion centered about the issue: to what extent would the results of the analysis have differed if individual animal data had been used? The answer to that question has two parts.

1. **The statistical methods used in the analysis depend on the data only through their dose group means, standard deviations, and sample sizes.**

   Thus, applying the same analysis to individual animal data would result in the same numerical estimates as the current analysis. The following argument shows why this is so. Whether the model fit uses generalized least squares or is a nonlinear mixed effects model (See III.B.1), the parameter estimates are the result of optimizing expressions that depend on the individual data through quadratic forms like:

$$(\mathbf{y} - \boldsymbol{\mu})' \mathbf{V} (\mathbf{y} - \boldsymbol{\mu})$$

   Here, **y** is a column vector of the individual observations $\{y_{ij}\}$, i indexes dose group (in this discussion, "dose group" refers to the observations on animals of the same sex exposed at the same time and dose to the same chemical) and j indexes individual within that dose group. The vector **μ** is the vector of fitted values. Since all individuals in the same dose group were exposed to the same dose, the fitted values for each individual in a dose group are all identical. Finally, the matrix **V** is symmetric, and has the form **D** + **M**, where **D** is diagonal, and partitioned such that the values corresponding to the same dose group are identical to each other. **M** is symmetric and partitioned into blocks that correspond to the dose groups. The values within any given block are identical to each other. The partitioning of the components of **V** is due to the fact that all the individuals of the same sex given the same dose in the same study are treated identically by the model. A direct consequence of the partitioning of **μ**

and **V** is that the value of the above quadratic form can be expressed solely in terms of group means, standard deviations, and sample sizes.

2. Distribution of the brain cholinesterase data.

The methods used in the dose-response analysis assume the data is normally distributed. If the individual cholinesterase activity measurements were distinctly non-normal, it would be of interest to determine the impact of transformed or trimmed data on the benchmark dose estimates used to estimate relative potency.

Individual animal data for female and male rat brain cholinesterase activity were available for a small subset of the studies used in the the Draft Revised Cumulative Risk Assessment for the OPs. Individual animal data were available from 15 studies representing 11 chemicals (see Table III.B.3-4). Each study included several dose-response data sets in both males and females; each dose-response data set included several dose groups. (Note to the reader: Individual animal data form male and female brain cholinesterase activity used in the following analysis have **NOT** been released to the public).

i. Test for normality.

Each individual dose group (sample sizes ranging from about 5 to 50) was tested for deviations from normality using the Shapiro-Wilk test for normality (Shapiro and Wilk, 1965). The P-values for each dose group in a study were then combined using Fisher's method (Sokal and Rohlf, 1981; section 18.1), giving an overall P-value for deviation from normality for each MRID. Table III.B.3-4 gives the results of this initial test for normality.

The result of combining all the P-values over all studies was highly significant: the P-value is 9×10^{-8}. Thus, there is evidence of some deviation from normality, though, given the amount of data available for the test, and the relatively few chemicals for which the overall P-value is significant (only 2/15 MRIDs have a significant deviation from normality), the overall deviation from normality does not seem excessive

ii. Identify the nature of the deviations from normality.

Two possibilities were explored: that the data were such that a power transformation (in the form of the Box-Cox transformation; Sokal and Rohlf, 1981, section 13.9) would result in a normal distribution, and that the data were "contaminated", that is, the bulk of the observations are from a normal distribution, with an

occasional too large or too small value (Rosenberger and Gasko, 1983). The approach taken in this analysis was to use maximum likelihood to estimate the parameters in two models:

1). An observation y is sampled from a normal distribution with mean μ and standard deviation σ with probabilty p, and from a normal distribution with mean μ and standard deviation $a \times \sigma$, where $a > 1$, with probability $1 - p$. Here the mean and standard deviation are specific to each dose group, but a is the same value for all dose groups in a study.

2) If the data y were transformed to z by the Box-Cox transformation: $z = \frac{y^t - 1}{t}$ (if $t \neq 0$) or $z = \log(y)$ (if $t = 0$), the transformed data would be normally distributed, with separate mean and standard deviation for each dose group (but only one power parameter t for each study). When $t = 1$, then $z = y - 1$, and the original variable y is normally distributed.

The Akaike Information Coefficient (AIC; Burnham and Anderson, 1998) was calculated for each of the two hypothetical distributions for each study. AIC is useful for comparing different probability models fit to the same data sets: smaller AIC values indicate better fits. Table III.B.3-5 shows the AIC values that resulted from fitting the two models just described to the individual animal data from each study. In addition, the power parameter estimated in the Box-Cox model was tested for significant difference from one.

For eight of the fifteen studies, the AIC for the Box-Cox transformed data was less than that for the contaminated normal. Only two of those studies had a Box-Cox parameter significantly different from one, indicating that a Box-Cox transformation would result in a significantly more normal distribution. In the remaining seven of the fifteen studies, including the two with significant Shapiro-Wilk tests, the contaminated normal model provides a better description of the data. The overall AIC for the contaminated normal distribution is less than that for the Box-Cox transformed data, showing that the contaminated normal model is superior to the Box-Cox model as a single overall probability model for these data.

iii. Impact of non-normality on the BMD estimates.

BMD_{10}s were calculated for trimmed and untrimmed data. Table III.B.3-6 shows the results of applying the Shapiro-Wilk test to the trimmed individual data. The overall P-value for all the data taken together is 0.056, indicating a substantial improvement

Aggregated datasets were produced from the original (untrimmed) individual data and the trimmed individual data, and both the basic and expanded models fit to each set of data for each chemical (See I.B and III.B.1). Four chemicals were affected by the trimming: dicrotophos, methamidophos, phorate, and phosalone. Thus, comparisons between untrimmed and trimmed data is limited to nine studies from four OPs.

Table III.B.3-7 compares the BMD_{10} calculated from the original data to that calculated using the trimmed data, for both basic and expanded models. The largest difference is less than 20% of the untrimmed value, which is reasonably small. *The current dose-response analysis used in the Draft Revised Cumulative Risk Assessment of the OPs, based solely on aggregated data, is relatively robust to the kinds of deviations from normality identified here.*

In summary, since the statistical methods used to fit dose-response models to the data depend on the data only through their means, standard deviations, and sample sizes, the only way an analysis of individual data might differ from that of aggregated data would be if the distribution of the data were substantially non-normal. The distributions of a subset of the data were examined, resulting in evidence that some studies did produce data that deviated from normality. When extreme observations were omitted, the overall distribution of the data became closer to a normal distribution. However, benchmark doses calculated using the trimmed data, were quite similar. to those using all the data. *Thus, it is unlikely that using aggregated data has substantially distorted the estimates of benchmark doses that would obtain had the analysis been based on individual animal data.*

Table III.B.3-4. Chemicals and studies used in individual animal analysis.

Chemical	Study (MRID no.)	Number of Dose Groups	Number Failed	Proportion Failed	Combined Shapiro-Wilks P-value
Methamidophos	148452	20	8	0.400	1.63e-08
Methamidophos	41867201	20	2	0.100	1.08e-01
Methamidophos	43197901	8	1	0.125	1.37e-01
Fenamiphos	44051401	8	1	0.125	1.60e-01
Bensulide	44161101	32	4	0.125	8.14e-02
ODM	44189501	36	1	0.028	8.09e-01
Fosthiazate	44269905	14	1	0.071	3.23e-01
Dicrotophos	44527802	16	4	0.250	1.01e-03
Phosalone	44801002	8	1	0.125	6.52e-02
Phosmet	44811801	16	1	0.063	4.63e-01
Terbufos	44842302	8	1	0.125	8.98e-02
Phosalone	44852504	24	0	0.000	2.60e-01
Phorate	44895301	8	1	0.125	7.55e-02
Phorate	44895302	10	0	0.000	2.71e-01
Chlorpyrifos-methyl	44906902	10	1	0.100	5.47e-02

"Number of Groups" is the total number of dose groups available; "Number Failed" is the number of individual dose groups for which the Shapiro-Wilks test reported a P-value less than 0.05; "Proportion Failed" is the proportion of dose groups that failed the test (Number Failed/Number of Groups); "Combined Shapiro-Wilks P-value" is the overall P-value for each MRID, resulting from using Fisher's method to combine the P-values for the individual dose-group tests.

Table III.B.3-5. AIC values for the Box-Cox and the contaminated normal models.

Chemical	Study (MRID no.)	AIC	
		Contaminated Normal	Box Cox Transformed
Methamidophos	**148452**	***362.46***	386.90
Methamidophos	41867201	***250.84***	251.01
Methamidophos	43197901	64.15	***62.26***
Fenamiphos	44051401	***105.23***	106.66
Bensulide	44161101	5586.60	***5578.49***
ODM	44189501	527.11	***524.59***
Fosthiazate	44269905	2242.37	***2239.92***
Dicrotophos	**44527802**	***213.37***	228.71
Phosalone	44801002	174.91	***164.19*** *
Phosmet	44811801	382.29	***380.07***
Terbufos	44842302	***339.71***	340.77
Phosalone	44852504	***204.14***	209.10
Phorate	44895301	***108.88***	119.00
Phorate	44895302	366.05	***359.44*** *
Chlorpyrifos-methyl	44906902	211.33	***208.39***
Sum:		**11139.44**	**11159.50**

MRID numbers for data that were significantly non-normal by the Shapiro-Wilks test (see Table III.B.3-4) are written in bold. The smaller of the two AIC values for each MRID is written in bold italics. When the Box-Cox power parameter is significantly different from 1, the Box-Cox AIC is followed by an asterisk.

Table III.B.3-6. P-values for the Shapiro-Wilks test, combined over all dose groups in a study for the trimmed individual data.

Chemical	Study (MRID no.)	P.value
Methamidophos	148452	0.046
Methamidophos	41867201	0.293
Methamidophos	43197901	0.137
Fenamiphos	44051401	0.160
Bensulide	44161101	0.081
ODM	44189501	0.809
Fosthiazate	44269905	0.323
Dicrotophos	44527802	0.937
Phosalone	44801002	0.065
Phosmet	44811801	0.463
Terbufos	44842302	0.090
Phosalone	44852504	0.863
Phorate	44895301	0.831
Phorate	44895302	0.271

Table III.B.3-7. Benchmark doses from the basic and expanded models for untrimmed (original) and trimmed data.

Chemical	Data Treatment	Female BMD_{10}	
		Expanded Model	Basic Model
Dicrotophos	original	NA	0.032
	trimmed	NA	0.026
Methamidophos	original	NA	0.080
	trimmed	NA	0.079
Phorate	original	0.215	0.036
	trimmed	0.201	0.037
Phosalone	original	6.426	3.843
	trimmed	6.313	3.847

NA: As shown in I.B, the basic model was used to estimate potency for methamidophos and dicrotophos.

References

Burnham, K. P. and Anderson, D. R. 1998. *Model Selection and Inference. A Practical information-Theoretic Approach.* Springer. New York.

Rosenberger, J. L. and Gasko, M. 1983. Comparing location estimators: trimmed means, medians, and trimean. Chapter 10 in *Understanding Robust and Exploratory Data Analysis*, David C. Hoaglin, Frederick Mosteller, and John W. Tukey, eds. Wiley. New York.

Shapiro, S. S. and Wilk, M. B. (1965). "An analysis of variance test for normality (complete samples)", Biometrika, 52: 591-611.

Sokal, R. R. and Rohlf, F. James. (1981). *Biometry, Second Edition.* Freeman. San Francisco.

III. Appendices

B. Hazard/RPF

4. R Programs for the Revised Analysis

a. Part 1: Package RBMDS

RBMDS is a package of utility functions written to facilitate the analysis of the cholinesterase activity dose-response data. These functions are made available to the scripts that carry out the data analysis by including the call "require(RBMDS)" or "library(RBMDS)" at the beginning of the scripts.

```
### Generalization of exponential decreasing model
### A, B, and m are constrained to be strictly positive
CexpB <- function (Dose, A, B, m)
{
  ## exp(A)*(1/(1 + exp(-B)) + (exp(-B)/(1 + exp(-B)) * exp(-exp(m)*Dose)))
  .expr1 <- exp(A)
  .expr3 <- exp(-B)
  .expr4 <- 1 + .expr3
  .expr6 <- exp(m)
  .expr9 <- exp(-.expr6 * Dose)
  .expr10 <- .expr3 * .expr9
  .expr11 <- .expr10/.expr4
  .expr13 <- .expr1 * (1/.expr4 + .expr11)
  .expr14 <- .expr4^2
  .value <- .expr13
  .grad <- array(0, c(length(.value), 3), list(NULL, c("A",
                                                       "B", "m")))
  .grad[, "A"] <- .expr13
  .grad[, "B"] <- .expr1 * (.expr3/.expr14 - (.expr11 - .expr10 *
                                                .expr3/.expr14))
  .grad[, "m"] <- -.expr1 * (.expr3 * (.expr9 * (.expr6 *
                                                   Dose))/.expr4)
  attr(.value, "gradient") <- .grad
  .value
}

### CexpB2: same as above, but gradient includes deriv wrt dose.

CexpB2 <-
function (Dose, A, B, m)
{
    .expr1 <- exp(A)
    .expr3 <- exp(-B)
    .expr4 <- 1 + .expr3
    .expr6 <- .expr3/.expr4
    .expr7 <- exp(m)
    .expr10 <- exp(-.expr7 * Dose)
    .expr13 <- .expr1 * (1/.expr4 + .expr6 * .expr10)
    .expr18 <- .expr4^2
    .value <- .expr13
    .grad <- array(0, c(length(.value), 4), list(NULL, c("Dose",
        "A", "B", "m")))
    .grad[, "Dose"] <- -.expr1 * (.expr6 * (.expr10 * .expr7))
    .grad[, "A"] <- .expr13
```

```
    .grad[, "B"] <- .expr1 * (.expr3/.expr18 - (.expr6 - .expr3 *
        .expr3/.expr18) * .expr10)
    .grad[, "m"] <- -.expr1 * (.expr6 * (.expr10 * (.expr7 *
        Dose)))
    attr(.value, "gradient") <- .grad
    .value
}


### CpkexpB: above model, with the additional assumption that there is
### saturable detoxification.  This is implemented by composing CexpB
### (now assuming that Dose in CexpB refers to internal dose), with
### a model that relates administered dose to internal dose:
###
### idose = 0.5*((Dose - S - D) + sqrt((Dose - S - D)^2 + 4*Dose*S))
###
### This model approaches a line with slope 1 and y-intercept -D as Dose
### increases to infinity.  The parameter 'S' controls the shape of the
### low-dose part of the curve.  For values close to 0, the curve looks
### very threshold-like; as S increases, the curve becomes more gradual.
### S and D must be positive
###
### The final function is (Dose refers to administered dose; exponentiation
### used to force positive parameters):
###
###
## exp(A)*(1/(1 + exp(-B)) + (exp(-B)/(1 + exp(-B)) * exp(-exp(m)*0.5*((Dose
- exp(S) - exp(D)) + sqrt((Dose - exp(S) - exp(D))^2 + 4*Dose*exp(S))))))
##


CpkexpB <- function (Dose, A, B, m, S, D)
{
  idose <- CpkB(Dose, S, D)
  .value <- CexpB2(idose, A, B, m)
  .grad <- array(0, c(length(.value), 5),
                 list(NULL,
                      c("A","B","m","S","D")))
  .grad[,c("A","B","m")] <- attr(.value,"gradient")[,c("A","B","m")]
  .grad[,c("S","D")] <- attr(.value,"gradient")[,"Dose"] *
    attr(idose, "gradient")[,c("S", "D")]
  attr(.value, "gradient") <- .grad
  .value
}

### CexpBS(dose, A, B, m, sex, fixed=NULL) allows fixed, a named list of
### vectors of fixed values for the model CexpB.  They can differ by sex.
### used, for example:
### nlme(model=chei ~ CexpBS(dose, A, m, sex,
###      fixed=list(B=c(F=-3.01, B=-2.78))), data=mydata, random=A+m~1 |
mrid)


CexpBS <- function(dose, A, B, m, sex, fixed=NULL) {
  callList <- vector("list",4)
  names(callList) <- c("Dose","A","B","m")
  sex <- switch(length(fixed) + 1,
                sex,
                m,
                B)

  callList[["Dose"]] <- dose
  ### Assume we will always estimate A
  callList[["A"]] <- A
  if ("B" %in% names(fixed)) {
```

```
    callList[["B"]] <- fixed[["B"]][as.character(sex)]
    callList[["m"]] <- if ("m" %in% names(fixed)) {
      fixed[["m"]][as.character(sex)]
    } else {
      B
    }
  } else {
    callList[["B"]] <- B
    callList[["m"]] <- if ("m" %in% names(fixed)) {
      fixed[["m"]][as.character(sex)]
    } else {
      m
    }
  }
  .value <- do.call("CexpB",callList)
  .grad <- attr(.value, "gradient")
  .grad <- .grad[,-match(names(fixed),colnames(.grad)),drop=FALSE]
  attr(.value,"gradient") <- .grad
  .value
}

### This is the same model, reparameterized so that, instead of 'm', we
### estimate 'BMD', for a BMR*100% reduction in mean response relative to
### control.
###
### Model is:
### ~ exp(A)*(1/(1 + exp(-B)) + exp(-B)/(1 + exp(-B))*exp(log(1 - BMR*(1 +
exp(B))) * Dose*exp(-BMD)))

CexpBwD <- function (Dose, A, B, BMD, BMR=0.10)
{
    .expr1 <- exp(A)
    .expr3 <- exp(-B)
    .expr4 <- 1 + .expr3
    .expr6 <- .expr3/.expr4
    .expr7 <- exp(B)
    .expr10 <- 1 - BMR * (1 + .expr7)
    .expr14 <- exp(-BMD)
    .expr15 <- log(.expr10) * Dose * .expr14
    .expr16 <- exp(.expr15)
    .expr19 <- .expr1 * (1/.expr4 + .expr6 * .expr16)
    .expr20 <- .expr4^2
    .value <- .expr19
    .grad <- array(0, c(length(.value), 3), list(NULL, c("A",
        "B", "BMD")))
    .grad[, "A"] <- .expr19
    .grad[, "B"] <- .expr1 * (.expr3/.expr20 - (.expr6 * (.expr16 *
        (BMR * .expr7/.expr10 * Dose * .expr14)) + (.expr6 -
        .expr3 * .expr3/.expr20) * .expr16))
    .grad[, "BMD"] <- -.expr1 * (.expr6 * (.expr16 * .expr15))
    attr(.value, "gradient") <- .grad
    .value
}
## The following returns the hessian, too (for calculating covariances
## from nlme models).
CexpBwDH <- function (Dose, A, B, BMD, BMR=0.10)
{
    .expr1 <- exp(A)
    .expr3 <- exp(-B)
    .expr4 <- 1 + .expr3
    .expr6 <- .expr3/.expr4
    .expr7 <- exp(B)
    .expr10 <- 1 - BMR * (1 + .expr7)
    .expr14 <- exp(-BMD)
    .expr15 <- log(.expr10) * Dose * .expr14
```

```
    .expr16 <- exp(.expr15)
    .expr19 <- .expr1 * (1/.expr4 + .expr6 * .expr16)
    .expr20 <- .expr4^2
    .expr21 <- .expr3/.expr20
    .expr22 <- BMR * .expr7
    .expr23 <- .expr22/.expr10
    .expr25 <- .expr23 * Dose * .expr14
    .expr26 <- .expr16 * .expr25
    .expr28 <- .expr3 * .expr3
    .expr30 <- .expr6 - .expr28/.expr20
    .expr34 <- .expr1 * (.expr21 - (.expr6 * .expr26 + .expr30 *
        .expr16))
    .expr35 <- .expr16 * .expr15
    .expr38 <- -.expr1 * (.expr6 * .expr35)
    .expr40 <- 2 * (.expr3 * .expr4)
    .expr42 <- .expr20^2
    .expr55 <- .expr30 * .expr26
    .value <- .expr19
    .grad <- array(0, c(length(.value), 3), list(NULL, c("A",
        "B", "BMD")))
    .hessian <- array(0, c(length(.value), 3, 3), list(NULL,
        c("A", "B", "BMD"), c("A", "B", "BMD")))
    .grad[, "A"] <- .expr19
    .hessian[, "A", "A"] <- .expr19
    .hessian[, "A", "B"] <- .hessian[, "B", "A"] <- .expr34
    .hessian[, "A", "BMD"] <- .hessian[, "BMD", "A"] <- .expr38
    .grad[, "B"] <- .expr34
    .hessian[, "B", "B"] <- -.expr1 * (.expr21 - .expr3 * .expr40/.expr42 +
        .expr6 * (.expr16 * ((.expr23 + .expr22 * .expr22/.expr10^2) *
            Dose * .expr14) - .expr26 * .expr25) - .expr55 -
            (.expr55 + (.expr30 - ((.expr28 + .expr28)/.expr20 -
                .expr28 * .expr40/.expr42)) * .expr16))
    .hessian[, "B", "BMD"] <- .hessian[, "BMD", "B"] <- .expr1 *
        (.expr30 * .expr35 + .expr6 * (.expr26 + .expr35 * .expr25))
    .grad[, "BMD"] <- .expr38
    .hessian[, "BMD", "BMD"] <- .expr1 * (.expr6 * (.expr35 +
        .expr35 * .expr15))
    attr(.value, "gradient") <- .grad
    attr(.value, "hessian") <- .hessian
    .value
}

### Include the derivative wrt Dose, to combine with the Pk model:

CexpBwD2 <- function (Dose, A, B, BMD, BMR=0.10)
{
    .expr1 <- exp(A)
    .expr3 <- exp(-B)
    .expr4 <- 1 + .expr3
    .expr6 <- .expr3/.expr4
    .expr7 <- exp(B)
    .expr10 <- 1 - BMR * (1 + .expr7)
    .expr11 <- log(.expr10)
    .expr14 <- exp(-BMD)
    .expr15 <- .expr11 * Dose * .expr14
    .expr16 <- exp(.expr15)
    .expr19 <- .expr1 * (1/.expr4 + .expr6 * .expr16)
    .expr24 <- .expr4^2
    .value <- .expr19
    .grad <- array(0, c(length(.value), 4), list(NULL, c("Dose",
        "A", "B", "BMD")))
    .grad[, "Dose"] <- .expr1 * (.expr6 * (.expr16 * (.expr11 *
        .expr14)))
    .grad[, "A"] <- .expr19
    .grad[, "B"] <- .expr1 * (.expr3/.expr24 - (.expr6 * (.expr16 *
```

```
        (BMR * .expr7/.expr10 * Dose * .expr14)) + (.expr6 -
        .expr3 * .expr3/.expr24) * .expr16))
    .grad[, "BMD"] <- -.expr1 * (.expr6 * (.expr16 * .expr15))
    attr(.value, "gradient") <- .grad
    .value
}

### Again, with the hessian:
CexpBwD2H <- function (Dose, A, B, BMD, BMR=0.1)
{
    .expr1 <- exp(A)
    .expr3 <- exp(-B)
    .expr4 <- 1 + .expr3
    .expr6 <- .expr3/.expr4
    .expr7 <- exp(B)
    .expr10 <- 1 - BMR * (1 + .expr7)
    .expr11 <- log(.expr10)
    .expr14 <- exp(-BMD)
    .expr15 <- .expr11 * Dose * .expr14
    .expr16 <- exp(.expr15)
    .expr19 <- .expr1 * (1/.expr4 + .expr6 * .expr16)
    .expr20 <- .expr11 * .expr14
    .expr21 <- .expr16 * .expr20
    .expr23 <- .expr1 * (.expr6 * .expr21)
    .expr27 <- BMR * .expr7
    .expr28 <- .expr27/.expr10
    .expr32 <- .expr28 * Dose * .expr14
    .expr33 <- .expr16 * .expr32
    .expr37 <- .expr3 * .expr3
    .expr38 <- .expr4^2
    .expr40 <- .expr6 - .expr37/.expr38
    .expr45 <- .expr16 * .expr15
    .expr51 <- .expr3/.expr38
    .expr56 <- .expr1 * (.expr51 - (.expr6 * .expr33 + .expr40 *
        .expr16))
    .expr59 <- -.expr1 * (.expr6 * .expr45)
    .expr61 <- 2 * (.expr3 * .expr4)
    .expr63 <- .expr38^2
    .expr76 <- .expr40 * .expr33
    .value <- .expr19
    .grad <- array(0, c(length(.value), 4), list(NULL, c("Dose",
        "A", "B", "BMD")))
    .hessian <- array(0, c(length(.value), 4, 4), list(NULL,
        c("Dose", "A", "B", "BMD"), c("Dose", "A", "B", "BMD")))
    .grad[, "Dose"] <- .expr23
    .hessian[, "Dose", "Dose"] <- .expr1 * (.expr6 * (.expr21 *
        .expr20))
    .hessian[, "Dose", "A"] <- .hessian[, "A", "Dose"] <- .expr23
    .hessian[, "Dose", "B"] <- .hessian[, "B", "Dose"] <- -.expr1 *
        (.expr6 * (.expr16 * (.expr28 * .expr14) + .expr33 *
            .expr20) + .expr40 * .expr21)
    .hessian[, "Dose", "BMD"] <- .hessian[, "BMD", "Dose"] <- -.expr1 *
        (.expr6 * (.expr21 + .expr45 * .expr20))
    .grad[, "A"] <- .expr19
    .hessian[, "A", "A"] <- .expr19
    .hessian[, "A", "B"] <- .hessian[, "B", "A"] <- .expr56
    .hessian[, "A", "BMD"] <- .hessian[, "BMD", "A"] <- .expr59
    .grad[, "B"] <- .expr56
    .hessian[, "B", "B"] <- -.expr1 * (.expr51 - .expr3 * .expr61/.expr63 +
        .expr6 * (.expr16 * ((.expr28 + .expr27 * .expr27/.expr10^2) *
            Dose * .expr14) - .expr33 * .expr32) - .expr76 -
            (.expr76 + (.expr40 - ((.expr37 + .expr37)/.expr38 -
                .expr37 * .expr61/.expr63)) * .expr16))
    .hessian[, "B", "BMD"] <- .hessian[, "BMD", "B"] <- .expr1 *
        (.expr40 * .expr45 + .expr6 * (.expr33 + .expr45 * .expr32))
```

```
    .grad[, "BMD"] <- .expr59
    .hessian[, "BMD", "BMD"] <- .expr1 * (.expr6 * (.expr45 +
        .expr45 * .expr15))
    attr(.value, "gradient") <- .grad
    attr(.value, "hessian") <- .hessian
    .value
}


### CexpB2wD(dose, A, PB, BMD, BMR=0.10) Same as CexpBwD, but PB is on
### original scale.
### Model is:
### ~ exp(A)*(PB + (1-PB)*exp(log((1 - BMR - PB)/(1 - PB)) *
Dose*exp(-BMD)))
### Primarily used to be called from CexpBwDS; grad[,"PB"] is not returned.
CexpB2wD <- function (dose, A, PB, BMD, BMR=0.1)  {
  .expr1 <- exp(A)
  .expr2 <- 1 - PB
  .expr4 <- 1 - BMR - PB
  .expr5 <- .expr4/.expr2
  .expr9 <- exp(-BMD)
  .expr10 <- log(.expr5) * dose * .expr9
  .expr11 <- exp(.expr10)
  .expr14 <- .expr1 * (PB + .expr2 * .expr11)
  .value <- .expr14
  .grad <- array(0, c(length(.value), 2), list(NULL, c("A", "BMD")))
  .grad[, "A"] <- .expr14
  .grad[, "BMD"] <- -.expr1 * (.expr2 * (.expr11 * .expr10))
  attr(.value, "gradient") <- .grad
  .value
}

### Same as above, but return gradient and hessian, and include PB in both
### (for computing standard errors)

CexpB2wDH <- function (Dose, A, PB, BMD, BMR)
{
    .expr1 <- exp(A)
    .expr2 <- 1 - PB
    .expr4 <- 1 - BMR - PB
    .expr5 <- .expr4/.expr2
    .expr6 <- log(.expr5)
    .expr9 <- exp(-BMD)
    .expr10 <- .expr6 * Dose * .expr9
    .expr11 <- exp(.expr10)
    .expr14 <- .expr1 * (PB + .expr2 * .expr11)
    .expr15 <- .expr6 * .expr9
    .expr16 <- .expr11 * .expr15
    .expr18 <- .expr1 * (.expr2 * .expr16)
    .expr23 <- .expr2^2
    .expr25 <- 1/.expr2 - .expr4/.expr23
    .expr26 <- .expr25/.expr5
    .expr30 <- .expr26 * Dose * .expr9
    .expr31 <- .expr11 * .expr30
    .expr38 <- .expr11 * .expr10
    .expr47 <- .expr1 * (1 - (.expr2 * .expr31 + .expr11))
    .expr50 <- -.expr1 * (.expr2 * .expr38)
    .expr51 <- 1/.expr23
    .value <- .expr14
    .grad <- array(0, c(length(.value), 4), list(NULL, c("Dose",
        "A", "PB", "BMD")))
    .hessian <- array(0, c(length(.value), 4, 4), list(NULL,
        c("Dose", "A", "PB", "BMD"), c("Dose", "A", "PB", "BMD")))
    .grad[, "Dose"] <- .expr18
    .hessian[, "Dose", "Dose"] <- .expr1 * (.expr2 * (.expr16 *
```

```
        .expr15))
    .hessian[, "Dose", "A"] <- .hessian[, "A", "Dose"] <- .expr18
    .hessian[, "Dose", "PB"] <- .hessian[, "PB", "Dose"] <- -.expr1 *
        (.expr2 * (.expr11 * (.expr26 * .expr9) + .expr31 * .expr15) +
            .expr16)
    .hessian[, "Dose", "BMD"] <- .hessian[, "BMD", "Dose"] <- -.expr1 *
        (.expr2 * (.expr16 + .expr38 * .expr15))
    .grad[, "A"] <- .expr14
    .hessian[, "A", "A"] <- .expr14
    .hessian[, "A", "PB"] <- .hessian[, "PB", "A"] <- .expr47
    .hessian[, "A", "BMD"] <- .hessian[, "BMD", "A"] <- .expr50
    .grad[, "PB"] <- .expr47
    .hessian[, "PB", "PB"] <- -.expr1 * (.expr2 * (.expr11 *
        (((.expr51 + .expr51 - .expr4 * (2 * .expr2)/.expr23^2)/.expr5 +
            .expr25 * .expr25/.expr5^2) * Dose * .expr9) - .expr31 *
        .expr30) - .expr31 - .expr31)
    .hessian[, "PB", "BMD"] <- .hessian[, "BMD", "PB"] <- .expr1 *
        (.expr38 + .expr2 * (.expr31 + .expr38 * .expr30))
    .grad[, "BMD"] <- .expr50
    .hessian[, "BMD", "BMD"] <- .expr1 * (.expr2 * (.expr38 +
        .expr38 * .expr10))
    attr(.value, "gradient") <- .grad
    attr(.value, "hessian") <- .hessian
    .value
}


### Above, but including derivative of dose, for the pk model

### CexpBwDS(dose, A, B, BMD, sex, fixed=NULL) allows fixed, a named list of
### vectors of fixed values for the model CexpBwD.  They can differ by sex.
### used, for example:
### nlme(model=chei ~ CexpBwDS(dose, A, BMD, sex,
###      fixed=list(PB=c(F=0.05, M=0.06)), data=mydata, random=A+m~1 | mrid)
### This implementation assumes PB is the only fixed parameter, and is on
### its original scale (0 <= PB < 1).

CexpBwDS <- function(dose, A, BMD, sex, fixed=NULL, BMR=0.10) {
  callList <- vector("list",5)
  names(callList) <- c("dose","A","PB","BMD","BMR")
  callList[["dose"]] <- dose
  callList[["A"]] <- A
  callList[["PB"]] <- if ("B" %in% names(fixed)) {
    B <- fixed[["B"]][as.character(sex)]
    1/(1 + exp(-B))
  } else {
    fixed[["PB"]][as.character(sex)]
  }
  callList[["BMD"]] <- BMD
  callList[["BMR"]] <- BMR
  do.call("CexpB2wD",callList)
}

### CpkexpBwD: pk combined with the exp model in terms of BMD:

CpkexpBwD <- function (Dose, A, B, BMD, S, D, BMR=0.10)
{
  idose <- CpkB(Dose, S, D)
  .value <- CexpBwD2(idose, A, B, BMD, BMR=BMR)
  .grad <- array(0, c(length(.value), 5),
                 list(NULL,
                      c("A","B","BMD","S","D")))
  .grad[,c("A","B","BMD")] <- attr(.value,"gradient")[,c("A","B","BMD")]
  .grad[,c("S","D")] <- attr(.value,"gradient")["Dose"] *
    attr(idose, "gradient")[,c("S", "D")]
```

```
  attr(.value, "gradient") <- .grad
  .value
}
###: same as above, but include hessian
CpkexpBwDH <- function (Dose, A, B, BMD, S, D, BMR=0.10)
{
  idose <- CpkBH(Dose, S, D)
  .value <- CexpBwD2H(idose, A, B, BMD, BMR=BMR)
  .grad <- array(0, c(length(.value), 5),
                 list(NULL,
                      c("A","B","BMD","S","D")))
  .grad[,c("A","B","BMD")] <- attr(.value,"gradient")[,c("A","B","BMD")]
  .grad[,c("S","D")] <- attr(.value,"gradient")[,"Dose"] *
    attr(idose, "gradient")[,c("S", "D")]
  .hessian <- array(0, c(length(.value),5,5),
                    c("A","B","BMD","S","D"),
                    c("A","B","BMD","S","D"))
  .hessian[,c("A","B","BMD"),c("A","B","BMD")] <-
    attr(.value,"hessian")[,c("A","B","BMD"),c("A","B","BMD")]
  .hessian[,c("S","D"),c("S","D")] <-
    attr(.value,"hessian")[,"Dose","Dose"] *
      (attr(.idose,"gradient")[,c("S","D"),c("S","D")])^2 +
      attr(.idose,"gradient")[,"Dose"] *
        attr(.idose,"hessian")[,c("S","D"),c("S","D")]
  .hessian[,"A","S"] <- .hessian[,"S","A"] <-
    attr(.value,"hessian")[,"Dose","A"] * attr(.idose,"gradient")[,"S"]
  .hessian[,"A","D"] <- .hessian[,"S","D"] <-
    attr(.value,"hessian")[,"Dose","A"] * attr(.idose,"gradient")[,"D"]
  .hessian[,"B","S"] <- .hessian[,"S","B"] <-
    attr(.value,"hessian")[,"Dose","B"] * attr(.idose,"gradient")[,"S"]
  .hessian[,"B","D"] <- .hessian[,"S","D"] <-
    attr(.value,"hessian")[,"Dose","B"] * attr(.idose,"gradient")[,"D"]
  .hessian[,"BMD","S"] <- .hessian[,"S","BMD"] <-
    attr(.value,"hessian")[,"Dose","BMD"] * attr(.idose,"gradient")[,"S"]
  .hessian[,"BMD","D"] <- .hessian[,"S","D"] <-
    attr(.value,"hessian")[,"Dose","BMD"] * attr(.idose,"gradient")[,"D"]

  attr(.value, "gradient") <- .grad
  attr(.value, "hessian") <- .hessian
  .value
}
CpkexpB2wDH <- function (Dose, A, PB, BMD, S, D, BMR=0.10)
{
  idose <- CpkBH(Dose, S, D)
  .value <- CexpB2wDH(idose, A, PB, BMD, BMR=BMR)
  .grad <- array(0, c(length(.value), 5),
                 list(NULL,
                      c("A","PB","BMD","S","D")))
  .grad[,c("A","PB","BMD")] <- attr(.value,"gradient")[,c("A","PB","BMD")]
  .grad[,c("S","D")] <- attr(.value,"gradient")[,"Dose"] *
    attr(idose, "gradient")[,c("S", "D")]
  .hessian <- array(0, c(length(.value),5,5),
                    list(NULL,c("A","PB","BMD","S","D"),
                    c("A","PB","BMD","S","D")))
  .hessian[,c("A","PB","BMD"),c("A","PB","BMD")] <-
    attr(.value,"hessian")[,c("A","PB","BMD"),c("A","PB","BMD")]
  .hessian[,"S","S"] <-
    attr(.value,"hessian")[,"Dose","Dose"] *
      attr(idose,"gradient")[,"S"] * attr(idose,"gradient")[,"S"] +
      attr(.value,"gradient")[,"Dose"] *
        attr(idose,"hessian")[,"S","S"]
  .hessian[,"D","D"] <-
    attr(.value,"hessian")[,"Dose","Dose"] *
      attr(idose,"gradient")[,"D"] * attr(idose,"gradient")[,"D"] +
      attr(.value,"gradient")[,"Dose"] *
```

```
        attr(idose,"hessian")[,"D","D"]
  .hessian[,"S","D"] <- .hessian[,"D","S"] <-
    attr(.value,"hessian")[,"Dose","Dose"] *
      attr(idose,"gradient")[,"S"] * attr(idose,"gradient")[,"D"] +
      attr(.value,"gradient")[,"Dose"] *
        attr(idose,"hessian")[,"S","D"]

  .hessian[,"A","S"] <- .hessian[,"S","A"] <-
    attr(.value,"hessian")[,"Dose","A"] * attr(idose,"gradient")[,"S"]
  .hessian[,"A","D"] <- .hessian[,"S","D"] <-
    attr(.value,"hessian")[,"Dose","A"] * attr(idose,"gradient")[,"D"]
  .hessian[,"PB","S"] <- .hessian[,"S","PB"] <-
    attr(.value,"hessian")[,"Dose","PB"] * attr(idose,"gradient")[,"S"]
  .hessian[,"PB","D"] <- .hessian[,"S","D"] <-
    attr(.value,"hessian")[,"Dose","PB"] * attr(idose,"gradient")[,"D"]
  .hessian[,"BMD","S"] <- .hessian[,"S","BMD"] <-
    attr(.value,"hessian")[,"Dose","BMD"] * attr(idose,"gradient")[,"S"]
  .hessian[,"BMD","D"] <- .hessian[,"S","D"] <-
    attr(.value,"hessian")[,"Dose","BMD"] * attr(idose,"gradient")[,"D"]

  attr(.value, "gradient") <- .grad
  attr(.value, "hessian") <- .hessian
  .value
}

### CpkexpB2wD: pk combined with the exp model in terms of BMD.  Assume PB,
### S, and D
### are fixed, and do not return their components of the gradient.

CpkexpB2wD <- function (dose, A, PB, BMD, S, D, BMR=0.10)
{
  idose <- CpkB(dose, S, D)
  .value <- CexpB2wD(idose, A, PB, BMD, BMR=BMR)
  .grad <- array(0, c(length(.value), 2),
                 list(NULL,
                      c("A","BMD")))
  .grad[,c("A","BMD")] <- attr(.value,"gradient")[,c("A","BMD")]
  attr(.value, "gradient") <- .grad
  .value
}

### CpkexpBS: pk combined with exp model, with fixed B and/or S values:
CpkexpBS <- function(dose, A, B, m, S, D, sex, fixed=NULL) {
  callList <- vector("list",6)
  names(callList) <- c("Dose","A","B","m","S","D")
  callList[["Dose"]] <- dose
  callList[["A"]] <- A
  if (length(fixed) == 0) sx <- sex
  if (length(fixed) == 1) sx <- D
  if (length(fixed) == 2) sx <- S
  if (length(fixed) == 3) sx <- m
  if ("B" %in% names(fixed)) {
    callList[["B"]] <- fixed[["B"]][as.character(sx)]
    callList[["m"]] <- B
    if ("S" %in% names(fixed)) {
      callList[["S"]] <- fixed[["S"]][as.character(sx)]
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
      } else {
        callList[["D"]] <- m
      }
    } else {
      callList[["S"]] <- m
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
```

```
      } else {
        callList[["D"]] <- S
      }
    }
  } else {
    callList[["B"]] <- B
    callList[["m"]] <- m
    if ("S" %in% names(fixed)) {
      callList[["S"]] <- fixed[["S"]][as.character(sx)]
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
      } else {
        callList[["D"]] <- S
      }
    } else {
      callList[["S"]] <- S
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
      } else {
        callList[["D"]] <- D
      }
    }
  }
  do.call("CpkexpB",callList)
}
### CpkexpBwDS: pk combined with exp model, with fixed B and/or S values:
CpkexpBwDS <- function(dose, A, B, BMD, S, D, sex, fixed=NULL) {
  callList <- vector("list",6)
  names(callList) <- c("Dose","A","B","BMD","S","D")
  callList[["Dose"]] <- dose
  callList[["A"]] <- A
  if (length(fixed) == 0) sx <- sex
  if (length(fixed) == 1) sx <- D
  if (length(fixed) == 2) sx <- S
  if (length(fixed) == 3) sx <- BMD
  if ("B" %in% names(fixed)) {
    callList[["B"]] <- fixed[["B"]][as.character(sx)]
    callList[["BMD"]] <- B
    if ("S" %in% names(fixed)) {
      callList[["S"]] <- fixed[["S"]][as.character(sx)]
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
      } else {
        callList[["D"]] <- BMD
      }
    } else {
      callList[["S"]] <- BMD
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
      } else {
        callList[["D"]] <- S
      }
    }
  } else {
    callList[["B"]] <- B
    callList[["BMD"]] <- BMD
    if ("S" %in% names(fixed)) {
      callList[["S"]] <- fixed[["S"]][as.character(sx)]
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
      } else {
        callList[["D"]] <- S
      }
    } else {
      callList[["S"]] <- S
```

```
      if ("D" %in% names(fixed)) {
        callList[["D"]] <- fixed[["D"]][as.character(sx)]
      } else {
        callList[["D"]] <- D
      }
    }
  }
  do.call("CpkexpBwD",callList)
}

### CpkexpB2wDS: pk combined with exp model, with fixed PB, S, and D values:

CpkexpB2wDS <- function(dose, A, BMD, sex, fixed=NULL) {
  callList <- vector("list",7)
  names(callList) <- c("dose","A","PB", "BMD","S","D","BMR")
  callList[["dose"]] <- dose
  callList[["A"]] <- A
  callList[["PB"]] <- fixed[["PB"]][as.character(sex)]
  callList[["BMD"]] <- BMD
  callList[["S"]] <- fixed[["S"]][as.character(sex)]
  callList[["D"]] <- fixed[["D"]][as.character(sex)]
  callList[["BMR"]] <- 0.1

  do.call("CpkexpB2wD",callList)
}

### Compute log(BMD) (for a BMR * 100% decrease in the mean) for the
### exponential
### model, based on the parameter transformations used here.
### This returns the gradient of log(BMD), to use in computing standard
### errors.
###
### log(BMD) <- expression(log(-log(1 - BMR*(1 + exp(B)))) - m)
### in terms of the transformation used in the CexpB models.
### fixed is a vector of strings, listing the parameters that should not
### be in the gradient.

CexplBMD <- function (BMR, m, B, fixed=NULL)
{
    .expr1 <- exp(B)
    .expr4 <- 1 - BMR * (1 + .expr1)
    .expr5 <- log(.expr4)
    .value <- log(-.expr5) - m
    .grad <- array(0, c(length(.value), 2), list(NULL, c("m",
        "B")))
    .grad[, "m"] <- -1
    .grad[, "B"] <- -BMR * .expr1/.expr4/.expr5
    if (!is.null(fixed) > 0) {
      .grad <- .grad[,-match(fixed,colnames(.grad)),drop=FALSE]
    }
    attr(.value, "gradient") <- .grad
    .value
}

### Essentially the same function, but takes as input our model object
### (xx) and returns a list with elements lBMD and lBMD.se,
### each with components for "F" and "M".


explBMD <- function(object, BMR) {
  fit <- object$Fitm
  if (inherits(fit, "nlme")) {
    Sigma <- fit$varFix
    Coefs <- fit$coefficients$fixed
  } else {
```

```
    Sigma <- fit$varBeta
    Coefs <- fit$coefficients
  }
### Fixed gives the fixed parameters
  tmp <- names(fit$call$model[[3]])
  Fixed <- NULL
  if ("fixed" %in% tmp) {
    tmp <- names(fit$call$model[[3]][["fixed"]])
    Fixed <- c(Fixed,c("m","B")[c("m","B") %in% tmp])
  }
  m.F <- if ("m" %in% Fixed)
    fit$call$model[[3]][["fixed"]][["m"]][["F"]]
  else
    if ("m.sexF" %in% names(Coefs)) Coefs["m.sexF"] else Coefs["m"]
  m.M <- if ("m" %in% Fixed)
    fit$call$model[[3]][["fixed"]][["m"]][["M"]]
  else
    if ("m.sexM" %in% names(Coefs)) Coefs["m.sexM"] else Coefs["m"]

  B.F <- if ("B" %in% Fixed)
    fit$call$model[[3]][["fixed"]][["B"]][["F"]]
  else
    if ("B.sexF" %in% names(Coefs)) Coefs["B.sexF"] else Coefs["B"]
  B.M <- if ("B" %in% Fixed)
    fit$call$model[[3]][["fixed"]][["B"]][["M"]]
  else
    if ("B.sexM" %in% names(Coefs)) Coefs["B.sexM"] else Coefs["B"]
  ### Females
  lBMD.F <- CexplBMD(BMR=BMR,m=m.F,B=B.F,fixed=Fixed) +
log(object$Dosescale)
  ### Males
  lBMD.M <- CexplBMD(BMR=BMR,m=m.M,B=B.M,fixed=Fixed) +
log(object$Dosescale)

  ## Get the standard errors
  ## 1) Get the 'Female' and 'Male' Sigmas
  indx <- match(c("m.sexF","B.sexF"),rownames(Sigma),nomatch=0)
  Sigma.F <- Sigma[indx,indx,drop=FALSE]

  ## Strip off the '.F' from the rownames and colnames
  rownames(Sigma.F) <- colnames(Sigma.F) <-
sub("\\.sexF","",rownames(Sigma.F))

  indx <- match(c("m.sexM","B.sexM"),rownames(Sigma),nomatch=0)
  Sigma.M <- Sigma[indx,indx,drop=FALSE]

  ## Strip off the '.M' from the rownames and colnames
  rownames(Sigma.M) <- colnames(Sigma.M) <-
sub("\\.sexM","",rownames(Sigma.M))

  ## Get the 'Female' and 'Male' gradients

  grad.F <- attr(lBMD.F,"gradient")
  grad.F <- grad.F[,rownames(Sigma.F)]
  grad.M <- attr(lBMD.M,"gradient")
  grad.M <- grad.M[,rownames(Sigma.M)]
  list(lBMD = c(F=c(lBMD.F),M=c(lBMD.M)),
       lBMD.se = c(F=sqrt(t(grad.F) %*% Sigma.F %*% grad.F),
         M=sqrt(t(grad.M) %*% Sigma.M %*% grad.M)))
}

### Compute BMD (for a BMR * 100% decrease in the mean) for the
### exponential
### model, based on the parameter transformations used here.
### This returns the gradient of BMD, to use in computing standard
```

```
### errors.
###
### BMD <- expression(-log(1 - BMR*(1 + exp(B)))*exp( - m ))
### in terms of the transformation used in the CexpB models.
### fixed is a vector of strings, listing the parameters that should not
### be in the gradient.

CexpBMD <- function (BMR, m, B, fixed)
{
    .expr1 <- exp(B)
    .expr4 <- 1 - BMR * (1 + .expr1)
    .expr5 <- log(.expr4)
    .expr8 <- exp(-m)
    .value <- -.expr5 * .expr8
    .grad <- array(0, c(length(.value), 2), list(NULL, c("m",
        "B")))
    .grad[, "m"] <- .expr5 * .expr8
    .grad[, "B"] <- BMR * .expr1/.expr4 * .expr8
    if (!is.null(fixed) > 0) {
      .grad <- .grad[,-match(fixed,colnames(.grad)),drop=FALSE]
    }
    attr(.value, "gradient") <- .grad
    .value
}

### Compute lBMD and its standard error for pkexpBS
### This calculates the value for both sexes at the same time

pkexpSlBMD.se <- function(object,BMR) {
  fitpk <- object$Fitpk
  if (inherits(fitpk, "nlme")) {
    Sigma <- fitpk$varFix
    Coefs <- fitpk$coefficients$fixed
  } else {
    Sigma <- fitpk$varBeta
    Coefs <- fitpk$coefficients
  }

### Fixed1 gives the fixed parameters for the CexpBMD part (i.e., 'B')
### Fixed2 gives the fixed parameters for the CpkBMD part (i.e., 'S')

  tmp <- names(fitpk$call$model[[3]])
  Fixed1 <- Fixed2 <- NULL
  if ("fixed" %in% tmp) {
    tmp <- names(fitpk$call$model[[3]][["fixed"]])
    Fixed1 <- c(Fixed1,c("m","B")[c("m","B") %in% tmp])
    Fixed2 <- c(Fixed2,c("S","D")[c("S","D") %in% tmp])
  }

  S.F <- if ("S" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["S"]][["F"]]
  else
    if ("S.sexF" %in% names(Coefs)) Coefs["S.sexF"] else Coefs["S"]
  S.M <- if ("S" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["S"]][["M"]]
  else
    if ("S.sexM" %in% names(Coefs)) Coefs["S.sexM"] else Coefs["S"]

  D.F <- if ("D" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["D"]][["F"]]
  else
    if ("D.sexF" %in% names(Coefs)) Coefs["D.sexF"] else Coefs["D"]
  D.M <- if ("D" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["D"]][["M"]]
  else
```

```
    if ("D.sexM" %in% names(Coefs)) Coefs["D.sexM"] else Coefs["D"]

  m.F <- if ("m" %in% Fixed1)
    fitpk$call$model[[3]][["fixed"]][["m"]][["F"]]
  else
    if ("m.sexF" %in% names(Coefs)) Coefs["m.sexF"] else Coefs["m"]
  m.M <- if ("m" %in% Fixed1)
    fitpk$call$model[[3]][["fixed"]][["m"]][["M"]]
  else
    if ("m.sexM" %in% names(Coefs)) Coefs["m.sexM"] else Coefs["m"]

  B.F <- if ("B" %in% Fixed1)
    fitpk$call$model[[3]][["fixed"]][["B"]][["F"]]
  else
    if ("B.sexF" %in% names(Coefs)) Coefs["B.sexF"] else Coefs["B"]
  B.M <- if ("B" %in% Fixed1)
    fitpk$call$model[[3]][["fixed"]][["B"]][["M"]]
  else
    if ("B.sexM" %in% names(Coefs)) Coefs["B.sexM"] else Coefs["B"]

  ### Females
  idose.F <- CexpBMD(BMR,m.F,B.F,fixed=Fixed1)
  lBMD.F <- CpkBi(as.vector(idose.F),S.F,D.F,fixed=Fixed2) +
    log(object$Dosescale)

  ### Males
  idose.M <- CexpBMD(BMR,m.M,B.M,fixed=Fixed1)
  lBMD.M <- CpkBi(as.vector(idose.M),S.M,D.M,fixed=Fixed2) +
    log(object$Dosescale)

  ## Get the standard errors
  ## 1) Get the 'Female' and 'Male' Sigmas
  indx <- match(c("m.sexF","B.sexF","S","D"),rownames(Sigma),nomatch=0)
  Sigma.F <- Sigma[indx,indx,drop=FALSE]

  ## Strip off the '.F' from the rownames and colnames
  rownames(Sigma.F) <- colnames(Sigma.F) <-
sub("\\.sexF","",rownames(Sigma.F))

  indx <- match(c("m.sexM","B.sexM","S","D"),rownames(Sigma),nomatch=0)
  Sigma.M <- Sigma[indx,indx,drop=FALSE]

  ## Strip off the '.M' from the rownames and colnames
  rownames(Sigma.M) <- colnames(Sigma.M) <-
sub("\\.sexM","",rownames(Sigma.M))

  ## 2) Get the 'Female' and 'Male' gradients

  grad.F <- c(attr(idose.F,"gradient")*attr(lBMD.F,"gradient")[,"idose"],
              attr(lBMD.F,"gradient")[,-match("idose",
                                              colnames(attr(lBMD.F,
                                                            "gradient"))),
                                      drop=FALSE])
  nm <- c(colnames(attr(idose.F,"gradient")),
          colnames(attr(lBMD.F,"gradient")))
  names(grad.F) <- nm[-match("idose",nm)]

  ## Make sure the elements are in the right order for Sigma.F

  grad.F <- grad.F[rownames(Sigma.F)]

  grad.M <- c(attr(idose.M,"gradient")*attr(lBMD.M,"gradient")[,"idose"],
              attr(lBMD.M,"gradient")[,-match("idose",
                                              colnames(attr(lBMD.M,
                                                            "gradient"))),
```

```
                                          drop=FALSE])
  nm <- c(colnames(attr(idose.M,"gradient")),
          colnames(attr(lBMD.M,"gradient")))
  names(grad.M) <- nm[-match("idose",nm)]
  ##Make sure the elements are in the right order for Sigma.M

  grad.M <- grad.M[rownames(Sigma.M)]

  list(lBMD=c(F=as.vector(lBMD.F),M=as.vector(lBMD.M)),
       lBMD.se=c(F = sqrt(t(grad.F) %*% Sigma.F %*% grad.F),
         M = sqrt(t(grad.M) %*% Sigma.M %*% grad.M)))
}



### Compute lBMD and its standard error for pkexpBwDS
### This calculates the value for both sexes at the same time

pkexpwDSlBMD.se <- function(object,BMR) {
  fitpk <- object$Fitpk
  if (inherits(fitpk, "nlme")) {
    Sigma <- fitpk$varFix
    Coefs <- fitpk$coefficients$fixed
  } else {
    Sigma <- fitpk$varBeta
    Coefs <- fitpk$coefficients
  }

### Fixed1 gives the fixed parameters for the CexpBMD part (i.e., 'B')
### Fixed2 gives the fixed parameters for the CpkBMD part (i.e., 'S')

  tmp <- names(fitpk$call$model[[3]])
  Fixed1 <- Fixed2 <- NULL
  if ("fixed" %in% tmp) {
    tmp <- names(fitpk$call$model[[3]][["fixed"]])
    Fixed1 <- c(Fixed1,c("BMD","B")[c("BMD","B") %in% tmp])
    Fixed2 <- c(Fixed2,c("S","D")[c("S","D") %in% tmp])
  }
## Extract the parameter estimates
  S.F <- if ("S" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["S"]][["F"]]
  else
    if ("S.sexF" %in% names(Coefs)) Coefs["S.sexF"] else Coefs["S"]
  S.M <- if ("S" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["S"]][["M"]]
  else
    if ("S.sexM" %in% names(Coefs)) Coefs["S.sexM"] else Coefs["S"]

  D.F <- if ("D" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["D"]][["F"]]
  else
    if ("D.sexF" %in% names(Coefs)) Coefs["D.sexF"] else Coefs["D"]
  D.M <- if ("D" %in% Fixed2)
    fitpk$call$model[[3]][["fixed"]][["D"]][["M"]]
  else
    if ("D.sexM" %in% names(Coefs)) Coefs["D.sexM"] else Coefs["D"]

  BMD.F <- if ("BMD" %in% Fixed1)
    fitpk$call$model[[3]][["fixed"]][["BMD"]][["F"]]
  else
    if ("BMD.sexF" %in% names(Coefs)) Coefs["BMD.sexF"] else Coefs["BMD"]
  BMD.M <- if ("BMD" %in% Fixed1)
    fitpk$call$model[[3]][["fixed"]][["BMD"]][["M"]]
```

```
  else
    if ("BMD.sexM" %in% names(Coefs)) Coefs["BMD.sexM"] else Coefs["BMD"]


### Calculate the log BMDs
  ## Females
  idose.F <- exp(BMD.F)
  lBMD.F <- CpkBi(as.vector(idose.F),S.F,D.F,fixed=Fixed2) +
    log(object$Dosescale)

  ### Males
  idose.M <- exp(BMD.M)
  lBMD.M <- CpkBi(as.vector(idose.M),S.M,D.M,fixed=Fixed2) +
    log(object$Dosescale)

### Calculate the standard errors
  ## 1) Get the 'Female' and 'Male' Sigmas
  indx <- match(c("BMD.sexF","S","D"),rownames(Sigma),nomatch=0)
  Sigma.F <- Sigma[indx,indx,drop=FALSE]

  ## Strip off the '.F' from the rownames and colnames
  rownames(Sigma.F) <- colnames(Sigma.F) <-
sub("\\.sexF","",rownames(Sigma.F))

  indx <- match(c("BMD.sexM","S","D"),rownames(Sigma),nomatch=0)
  Sigma.M <- Sigma[indx,indx,drop=FALSE]

  ## Strip off the '.M' from the rownames and colnames
  rownames(Sigma.M) <- colnames(Sigma.M) <-
sub("\\.sexM","",rownames(Sigma.M))

  ## 2) Get the 'Female' and 'Male' gradients

  grad.F <- c(idose.F*attr(lBMD.F,"gradient")[,"idose"],
              attr(lBMD.F,"gradient")[,-match("idose",
                                              colnames(attr(lBMD.F,
                                                            "gradient"))),
                                      drop=FALSE])
  nm <- c("BMD",
          colnames(attr(lBMD.F,"gradient")))
  names(grad.F) <- nm[-match("idose",nm)]

  ## Make sure the elements are in the right order for Sigma.F

  grad.F <- grad.F[rownames(Sigma.F)]

  grad.M <- c(idose.M*attr(lBMD.M,"gradient")[,"idose"],
              attr(lBMD.M,"gradient")[,-match("idose",
                                              colnames(attr(lBMD.M,
                                                            "gradient"))),
                                      drop=FALSE])
  nm <- c("BMD",
          colnames(attr(lBMD.M,"gradient")))
  names(grad.M) <- nm[-match("idose",nm)]
  ##Make sure the elements are in the right order for Sigma.M

  grad.M <- grad.M[rownames(Sigma.M)]

  list(lBMD=c(F=as.vector(lBMD.F),M=as.vector(lBMD.M)),
       lBMD.se=c(F = sqrt(t(grad.F) %*% Sigma.F %*% grad.F),
         M = sqrt(t(grad.M) %*% Sigma.M %*% grad.M)))
}

### CpkB: a low-dose modification to simulate the effect of first-pass
### metabolism on the relationship between administered dose and
```

```
### internal dose.
### Model expression is:
###  ~0.5*(dose - exp(S) - exp(D) + sqrt((dose - exp(S) - exp(D))^2 +
4*dose*exp(S)))
### This maps administered dose (dose) to internal dose (CpkB)
"CpkB" <-
function (dose, S, D)
{
    .expr1 <- exp(S)
    .expr3 <- exp(D)
    .expr4 <- dose - .expr1 - .expr3
    .expr7 <- 4 * dose * .expr1
    .expr8 <- .expr4^2 + .expr7
    .expr15 <- .expr8^-0.5
    .value <- 0.5 * (.expr4 + sqrt(.expr8))
    .grad <- array(0, c(length(.value), 2), list(NULL, c("S",
        "D")))
    .grad[, "S"] <- 0.5 * (0.5 * ((.expr7 - 2 * (.expr1 * .expr4)) *
        .expr15) - .expr1)
    .grad[, "D"] <- -0.5 * (0.5 * (2 * (.expr3 * .expr4) * .expr15) +
        .expr3)
    attr(.value, "gradient") <- .grad
    .value
}

### CpkBH: just like CpkB, but also returns the hessian
CpkBH <- function (dose, S, D)
{
    .expr1 <- exp(S)
    .expr3 <- exp(D)
    .expr4 <- dose - .expr1 - .expr3
    .expr7 <- 4 * dose * .expr1
    .expr8 <- .expr4^2 + .expr7
    .expr12 <- .expr1 * .expr4
    .expr14 <- .expr7 - 2 * .expr12
    .expr15 <- .expr8^-0.5
    .expr25 <- .expr8^-1.5
    .expr36 <- .expr3 * .expr4
    .expr37 <- 2 * .expr36
    .expr39 <- -0.5 * (.expr37 * .expr25)
    .value <- 0.5 * (.expr4 + sqrt(.expr8))
    .grad <- array(0, c(length(.value), 2), list(NULL, c("S",
        "D")))
    .hessian <- array(0, c(length(.value), 2, 2), list(NULL,
        c("S", "D"), c("S", "D")))
    .grad[, "S"] <- 0.5 * (0.5 * (.expr14 * .expr15) - .expr1)
    .hessian[, "S", "S"] <- 0.5 * (0.5 * ((.expr7 - 2 * (.expr12 -
        .expr1 * .expr1)) * .expr15 + .expr14 * (-0.5 * (.expr14 *
        .expr25))) - .expr1)
    .hessian[, "S", "D"] <- .hessian[, "D", "S"] <- 0.5 * (0.5 *
        (2 * (.expr1 * .expr3) * .expr15 - .expr14 * .expr39))
    .grad[, "D"] <- -0.5 * (0.5 * (.expr37 * .expr15) + .expr3)
    .hessian[, "D", "D"] <- -0.5 * (0.5 * (2 * (.expr36 - .expr3 *
        .expr3) * .expr15 - .expr37 * .expr39) + .expr3)
    attr(.value, "gradient") <- .grad
    attr(.value, "hessian") <- .hessian
    .value
}

### and this is the inverse function: maps internal dose to the
corresponding
### administered dose.
### CpkBi <- expression((idose^2 + idose*(exp(S) + exp(D)))/(idose +
exp(S)))
###
```

```
### Since I want to give log(BMD), use:
### CpkBi <- expression(log(idose^2 + idose*(exp(S) + exp(D))) - log(idose +
exp(S)))
### fixed: a vector of strings of parameters that will not be included in
### the gradient
CpkBi <- function (idose, S, D, fixed=NULL)
{
    .expr2 <- exp(S)
    .expr3 <- exp(D)
    .expr4 <- .expr2 + .expr3
    .expr6 <- idose^2 + idose * .expr4
    .expr8 <- idose + .expr2
    .value <- log(.expr6) - log(.expr8)
    .grad <- array(0, c(length(.value), 3), list(NULL, c("idose",
        "S", "D")))
    .grad[, "idose"] <- (2 * idose + .expr4)/.expr6 - 1/.expr8
    .grad[, "S"] <- idose * .expr2/.expr6 - .expr2/.expr8
    .grad[, "D"] <- idose * .expr3/.expr6
    if (!is.null(fixed))
      .grad <- .grad[,-match(fixed,colnames(.grad)),drop=FALSE]
    attr(.value, "gradient") <- .grad
    .value
}
"mypkPredict" <- function (object, newdata, level) {
  fparms <- object$coefficients$fixed
  rparms <- object$coefficients$random
  B.est <- !("fixed" %in% names(object$call$model[[3]])) ||
           !("B" %in% names(object$call$model[[3]][["fixed"]]))
  D.est <- !("fixed" %in% names(object$call$model[[3]])) ||
           !("D" %in% names(object$call$model[[3]][["fixed"]]))
  S.est <- !("fixed" %in% names(object$call$model[[3]])) ||
           !("S" %in% names(object$call$model[[3]][["fixed"]]))
  B.sex <- if (B.est) "B.sexF" %in% names(fparms) else NA
  indx <- paste("A.s.U", as.character(newdata$s.U), sep = "")
  A <- fparms[indx]
  B <- if (B.est) {
    indx <- if (B.sex) {
      paste("B.", "sex", as.character(newdata$sex), sep = "")
    } else {
      rep("B",nrow(newdata))
    }
    fparms[indx]
  } else {
    eval(object$call$model[[3]][["fixed"]])$B[as.character(newdata$sex)]
  }
  indx <- paste("BMD.", "sex", as.character(newdata$sex),
                sep = "")
  BMD <- fparms[indx]
### S is always fixed in my application
  S <- if (S.est) {
    rep(fparms["S"],nrow(newdata))
  } else {

eval(object$call$model[[3]][["fixed"]][["S"]])[as.character(newdata$sex)]
  }

### for simplicity, I'm assuming what we actually did, that is, model D ~ 1
### but fixed=list(D=c(F=,M=))
  D <- if (D.est) {
    rep(fparms["D"],nrow(newdata))
  } else {

eval(object$call$model[[3]][["fixed"]][["D"]])[as.character(newdata$sex)]
  }
  if (level >= 1) {
```

```
    grpname <- names(rparms)[1]
    if ("A.(Intercept)" %in% colnames(rparms[[1]]))
      A <- A + rparms[[1]][as.character(newdata[,grpname]),
                           "A.(Intercept)"]
    if (B.est) {
      B <- B + if (B.sex) {
        rparms[[1]][as.character(newdata[,grpname]), "B.(Intercept)"]
      } else {
        rparms[[1]][as.character(newdata[,grpname]), "B"]
      }
    }

    BMD <- BMD + rparms[[1]][as.character(newdata[,grpname]),
"BMD.(Intercept)"]
  }
  if (level == 2) {
    indx <- paste(as.character(newdata$mrid), as.character(newdata$set),
                  sep = "/")
    if ("A.(Intercept)" %in% colnames(rparms[[2]]))
      A <- A + rparms[[2]][indx, "A.(Intercept)"]
    if (B.est)
      B <- B + if (B.sex) {
        rparms[[2]][indx, "B.(Intercept)"]
      } else {
        rparms[[2]][indx, "B"]
      }
    BMD <- BMD + rparms[[2]][indx, "BMD.(Intercept)"]
  }
  CpkexpBwD(newdata$Dose.scaled, A, B, BMD, S, D)
}
### The following function is not general, but applies only to the model
### as used in the analysis of the OP data

mypkPredict2 <- function (object, newdata, level) {
  fparms <- object$coefficients$fixed
  rparms <- object$coefficients$random

  ## Fixed Effects
  indx <- paste("A.s.M.t", as.character(newdata$s.M.t), sep = "")
  A <- fparms[indx]

  PB <-
eval(object$call$model[[3]][["fixed"]])$PB[as.character(newdata$sex)]

  indx <- paste("BMD.", "sex", as.character(newdata$sex),
                sep = "")
  BMD <- fparms[indx]
### S is always fixed in my application
  S <-
eval(object$call$model[[3]][["fixed"]][["S"]])[as.character(newdata$sex)]

### for simplicity, I'm assuming what we actually did, that is, model D ~ 1
### but fixed=list(D=c(F=,M=))
  D <-
eval(object$call$model[[3]][["fixed"]][["D"]])[as.character(newdata$sex)]
  if (level >= 1) {
    grpname <- names(rparms)[1]
    BMD <- BMD + rparms[[1]][as.character(newdata[,grpname]),
"BMD.(Intercept)"]
  }
  if (level == 2) {
    indx <- paste(as.character(newdata$mrid), as.character(newdata$set),
                  sep = "/")
    BMD <- BMD + rparms[[2]][indx, "BMD.(Intercept)"]
  }
```

```
  CpkexpB2wD(newdata$Dose.scaled, A, PB, BMD, S, D)
}
myPredict <- function(object, newdata,level) {
  fparms <- object$coefficients$fixed
  rparms <- object$coefficients$random
  ## Did we estimate Bs?
  B.est <- !("fixed" %in% names(object$call$model[[3]]))
  ## Did the Bs differ between sexes"
  B.sex <- if (B.est) "B.sexM" %in% names(fparms) else NA
  B.REname <- if (B.est) {
    if (B.sex) "B.(Intercept)" else "B"
  } else NA
  ## is  this 'm' or 'BMD'?
  is.m <- "m.sexF" %in% names(fparms)

  ## Take care of the fixed effects
  indx <- paste("A.","s.U",as.character(newdata$s.U),sep="")
  A <- fparms[indx]
  if (B.est) {
    B <- if (B.sex) {
      indx <- paste("B.","sex",as.character(newdata$sex),sep="")
      fparms[indx]
    } else {
      rep(fparms["B"],nrow(newdata))
    }
  } else {
    B <- object$call$model[[3]]["fixed"][[1]]$B[as.character(newdata$sex)]
  }
  if (is.m) {
    indx <- paste("m.","sex",as.character(newdata$sex),sep="")
    thirdcol <- "m.(Intercept)"
  } else {
    indx <- paste("BMD.sex",as.character(newdata$sex),sep="")
    thirdcol <- "BMD.(Intercept)"
  }
  m <- fparms[indx]
  if (level >= 1) {
    grpname <- names(rparms)[1]
    if ("A.(Intercept)" %in% colnames(rparms[[1]]))
      A <- A + rparms[[1]][as.character(newdata[,grpname]),"A.(Intercept)"]
    if (B.est)
      B <- B + rparms[[1]][as.character(newdata[,grpname]),B.REname]
    m <- m + rparms[[1]][as.character(newdata[,grpname]),thirdcol]
  }
  if (level == 2) {
    indx <- paste(as.character(newdata$mrid),
                  as.character(newdata$set),sep="/")
    if ("A.(Intercept)" %in% colnames(rparms[[2]]))
      A <- A + rparms[[2]][indx,"A.(Intercept)"]
    if (B.est)
      B <- B + rparms[[2]][indx,B.REname]
    m <- m + rparms[[2]][indx,thirdcol]
  }
  if (is.m) {
    CexpB(newdata$Dose.scaled,A,B,m)
  } else {
    CexpBwD(newdata$Dose.scaled,A,B,m)
  }

}
myPredict2 <- function(object, newdata,level) {
  fparms <- object$coefficients$fixed
  rparms <- object$coefficients$random

  ## Take care of the fixed effects
```

```
  indx <- paste("A.s.M.t",as.character(newdata$s.M.t),sep="")
  A <- fparms[indx]

  B <- object$call$model[[3]]["fixed"][[1]][[1]][as.character(newdata$sex)]

  indx <- paste("BMD.sex",as.character(newdata$sex),sep="")
  thirdcol <- "BMD.(Intercept)"
  m <- fparms[indx]
  if (level >= 1) {
    grpname <- names(rparms)[1]
    if ("A.(Intercept)" %in% colnames(rparms[[1]]))
      A <- A + rparms[[1]][as.character(newdata[,grpname]),"A.(Intercept)"]
    m <- m + rparms[[1]][as.character(newdata[,grpname]),thirdcol]
  }
  if (level == 2) {
    indx <-
paste(as.character(newdata$mrid),as.character(newdata$set),sep="/")
    if ("A.(Intercept)" %in% colnames(rparms[[2]]))
      A <- A + rparms[[2]][indx,"A.(Intercept)"]
    m <- m + rparms[[2]][indx,thirdcol]
  }
  if ("B" %in% names(object$call$model[[3]]["fixed"][[1]]))
    CexpBwD(newdata$Dose.scaled,A,B,m)
  else
    CexpB2wD(newdata$Dose.scaled, A, B, m)
}
Expr <- expression(exp(A)*(1/(1 + exp(-B)) + (exp(-B)/(1 +
exp(-B)))*exp(log((1 - BMR - B)/(1-B))*Dose/BMD)))


### Function to produce a phony dataset for input into gls and gnls
### Dose, N, M, and SD can be vectors of the same length
### DoseName and RespName are strings that give the names for the
### corresponding
### variables

PhonyDF <- function(Dose,N,M,SD,DoseName,RespName,chem=NULL,ChemName=NULL){
  tmp <- data.frame(rep(Dose,N),
                    qlnorm01(N,M,SD))
  names(tmp) <- c(DoseName,RespName)
  if (!is.null(ChemName)) tmp[,ChemName] <- factor(rep(chem,N))
  tmp
}
### Possible values are "Best","Biggest","Both","None"
options(BMDSplot="Best")

assign("%inint%",function(x,interval) (min(interval) <= x && max(interval)
>= x))
plot.BMDS <- function(X, which=getOption("BMDSplot"),
                      LogX=c("auto","log","linear"),
                      ...) {
  dots <- list(...)
  reduceddots <- dots
  if ((indx <- match("ylim",names(reduceddots),nomatch=0)) > 0)
    reduceddots <- reduceddots[-indx]
  if ((indx <- match("xlim",names(reduceddots),nomatch=0)) > 0)
    reduceddots <- reduceddots[-indx]
  if ((indx <- match("Title",names(reduceddots),nomatch=0)) > 0)
    reduceddots <- reduceddots[-indx]
  switch(which,
         Best={
           Agg <- !is.null(X$Data[,X$varNames["SS"]])
           if (!Agg) {
             ## if there is replication, make an aggregated dataset
             ## and set MyAgg to be true
```

```
            tmp <- rle(sort(X$Data[,X$varNames["Dose"]]))$lengths
            MyAgg <- sum(tmp > 3) >= length(tmp) - 2
            if (MyAgg) {
              indx <- order(X$Data[,X$varNames["Dose"]])
              dose <- unique(X$Data[indx,X$varNames["Dose"]])
              resp <- tapply(X$Data[indx,X$varNames["Resp"]],
                             factor(X$Data[indx,X$varNames["Dose"]]),
                             mean)
              sd <- tapply(X$Data[indx,X$varNames["Resp"]],
                             factor(X$Data[indx,X$varNames["Dose"]]),
                             function(x) sqrt(var(x)))
              MyData <- data.frame(dose,resp,sd,tmp)
              names(MyData) <- X$varNames[c("Dose","Resp","SD","SS")]
            } else MyData <- X$Data
          } else MyData <- X$Data
          LogX <- match.arg(LogX)
          if (LogX == "auto") {
            if (Agg || MyAgg) {
              dose <- sort(MyData[,X$varNames["Dose"]])
              nd <- length(dose)
              LogX <- if ((dose[nd] - dose[1])/(dose[2] - dose[1]) > 20)
                "log" else "linear"
            } else {
              ## This is a copout for epi data, but works for now
              LogX <- "linear"
            }
          }
          drange <- range(c(MyData[,X$varNames["Dose"]],X$BMD))
          doses <- seq(min(drange),max(drange),length=101)
          if (LogX == "log" && 0 %inint% doses) doses <- doses[doses > 0]
          if (!is.null(X$Fit)) {
            newdata <- data.frame(Dose=doses/X$DoseScale)
            names(newdata) <- X$varNames["Dose"]
            predcrv <-predict(X$Fit,newdata=newdata)*X$RespScale
            if (!is.null(X$RR)) {
              newdata <- data.frame(c(0,X$BMD/X$DoseScale))
              names(newdata) <- X$varNames["Dose"]
              critresp <- predict(X$Fit,newdata=newdata)*X$RespScale
              BMR <- critresp[1]*(1 - X$RR)
            }
            else {
              critresp <- BMR <- NULL
              }
            } else predcrv <- critresp <- BMR <- NULL
            if (Agg || MyAgg) {
              mn <- MyData[,X$varNames["Resp"]]
              ss <- MyData[,X$varNames["SS"]]
              sd <- MyData[,X$varNames["SD"]]
              mnlcl <- mn - qt(0.975,ss - 1)*sd/sqrt(ss)
              mnucl <- mn + qt(0.975,ss - 1)*sd/sqrt(ss)
            } else {
              mnlcl <- mnucl <- MyData[,X$varNames["Resp"]]
            }
            ylim <- if (is.null(dots$ylim))
range(c(predcrv,mnlcl,mnucl,BMR))
            else dots$ylim
            xlim <- if (is.null(dots$xlim)) drange else dots$xlim
            Title <- if (is.null(dots$Title)) X$RunName else dots$Title
            if (Agg || MyAgg)
              do.call("plotCI",c(list(MyData[,X$varNames["Dose"]],
                                      MyData[,X$varNames["Resp"]],
                                      aui=mnucl,ali=mnlcl,err="y",
                                      ylim=ylim,xlim=xlim,
                                      xlab=X$varNames["Dose"],
                                      ylab=X$varNames["Resp"]),
```

```
                                    reduceddots))
            else
              do.call("plot",c(list(MyData[,X$varNames["Dose"]],
                                    MyData[,X$varNames["Resp"]],
                                    ylim=ylim,xlim=xlim,
                                    xlab=X$varNames["Dose"],
                                    ylab=X$varNames["Resp"]),
                               reduceddots))
            if (MyAgg) points(X$Data[,X$varNames["Dose"]],
                              X$Data[,X$varNames["Resp"]])
            if (!is.null(X$Fit)) lines(doses,predcrv,col="green")
###          } else {
###          }
          if (!is.null(X$RR) && !is.null(X$Fit)) {
            segments(par("usr")[1],BMR,X$BMD,BMR,col="red")
            segments(X$BMD,par("usr")[3],X$BMD,BMR,col="red")
            segments(X$BMDL,par("usr")[3],X$BMDL,BMR,col="red")
          }
          title(main=Title,sub=X$ModelName)
        },
        Biggest={
          if (is.null(X$FitAllDoses)) plot(X,which="Best",...)
          else plot(X$FitAllDoses,which="Best",...)
        },
        Both={
          plot(X,which="Best",...)
          if (!is.null(X$FitAllDoses)) plot(X$FitAllDoses,which="Best",...)
        },
        None={
          invisible(NULL)
        })
}
### Function to generate N approximately lognormal quantiles with exactly
### mean M and sd SD.  Does reasonable things if N,M,and SD are vectors
### that have the same length, and exits otherwise.  If N[i] == 1,
### then just returns the value of M[i].  If SD[i] == 0, returns
### rep(M[i],N[i]).

qlnorm01 <- function(N,M,SD) {
  if((length(N) != length(M) || (length(N) != length(SD))))
    stop("N, M, and SD must have the same length")
  out <- NULL
  for (i in 1:length(N)) {
    if (N[i] > 1) {
      if (SD[i] > 0) {
        sdlog <- sqrt(log(SD[i]^2/M[i]^2 + 1))
        mnlog <- log(M[i]) - sdlog^2/2
        y <- qlnorm(ppoints(N[i]),mnlog,sdlog)
        sd <- sqrt(var(y))
        mn <- mean(y)
        out <- c(out,SD[i]*(y - mn)/sd + M[i])
      } else {
        out <- c(out,rep(M[i],N[i]))
      }
    } else out <- c(out, M)
  }
  out
}
```

b. Part 2: Modifications to the package nlme.

Two small modifications to the package nlme (version 3.1-24 was used here) were required to facilitate convergence of the models.

```
R/reStruct.R
300c300
<       PACKAGE = "nlme")$loglik
---
>       PACKAGE = "nlme",NAOK=TRUE)$loglik

src/nlme.c
292c292
<             -1. /*dlt*/, pow(epsm, 1.0/3.0) /*gradtl*/, 0. /*stepmx*/,
---
>             -1. /*dlt*/, pow(epsm, 1.0/3.0) /*gradtl*/, 1.0 /*stepmx*/,
```

c. Part 3: Estimating parameters.

Parameters for both the basic and expanded models were estimated by maximizing a profile likelihood. Since this required repeated optimizations, each of which could take a significant amount of time, the optimizations were carried out on a cluster of computers. A single "master" script passed out tasks to be completed to "slave" scripts running on the different nodes of the cluster. The R package *rpvm*, which is an interface to the software package *pvm*, was used to facilitate communication between the master program and the slaves. After the *virtual machine* was initiated using the pvm console program, the simulations were started using the command line:

```
R --slave < master.R > master.Rout 2>&1 &
```

d. Part 3a: Basic model.

Before running 'master.R', another script, 'getStartVals.R' was used to set up directory structures and some basic data structures that were then used by the simulation programs. After running this sequence of scripts once, the grid is refined, 'master.R' and 'slave.R' are updated, and the whole thing run again.

```
### GetStartVals.R
### Construct skeletons of the Fit files that includes everything we can
### determine in advance: data set, pseudodata, etc, including skeletons
### of the LL and Fits vectors for holding the results.  Reorder the Bgrid
### data frame so that we start at the lower left corner (Pf = Pm = 0.001)
### and end up in the upper right corner.  When we do the fits, the start
### value for the fit will be the previous set of estimates
### In particular, set up a structure that holds the initial estimates
###
### From the pre-SAP-review data, extract the best parameter estimates from
### the basic model, and build a list.
###
### This is a list instead of a dataframe because the number of parameters
### will vary, because of the number of background parameters, and whether
### we have one or two estimates of B.  The entry for each chemical is a
### list of two elements, named "A", and "BMD".  Each element is
### a vector of parameter estimates, named appropriately.

### Make sure parameters are expressed on the original data
### scale, not the values based on the rescaled data.  New datasets may
### change the scale!  New datasets may also add new background values.

### The following function takes as its argument one of the Basic Model
### objects and returns a list with the proper format.
###
### Finally, we execute this code only once for any chemical.  Another
script
### will be used in case we need to updata data for some chemicals and rerun

require(nlme)
require(RBMDS)

getparms <- function(x,dta) {
  ## Extract the parameters
  parms <- if(!is.null(x)) {
    if (inherits(x$FitBMD,"nlme")) {
      x$FitBMD$coefficients$fixed
    } else {
      x$FitBMD$coefficients
    } } else {
      NULL
    }

  ## Break them into A, and BMD
  out <- vector("list",2)
  names(out) <- c("A","BMD")
  ## Get the values of s.M.t
  out[["A"]] <- log(unlist(unclass(by(dta,dta$s.M.t,function(x) {
    mean(x$chei[x$dose == min(x$dose)],na.rm=TRUE)
  }))))
  nm <- names(out[["A"]])
  out[["A"]] <- as.vector(out[["A"]])
  names(out[["A"]]) <- nm


  ## Get BMD from x if possible, else figure it is about 1/4 way between
  ## the maximum and minimum dose
  if (!is.null(parms)) {
    tmp <- parms[grep("^BMD",names(parms))]
    ## Rescale using x$Dosescale
    out[["BMD"]] <- tmp + log(x$Dosescale)
  } else {
    out[["BMD"]] <- log(rep(max(dta$dose)/4,2))
    names(out[["BMD"]]) <- c("BMD.sexF","BMD.sexM")
  }
```

```
  ## There is no point is getting "B", since that will be fixed.
  out
}


attach("../Data/opdata.rda")

Nchems <- length(Chemicals)
BasicStarts <- vector("list",Nchems)
names(BasicStarts) <- Chemicals
Nsteps <- 16
Bgrid <- expand.grid(B.F=seq(0.001,0.8,length=Nsteps),
                     B.M=seq(0.001, 0.8, length=Nsteps))
Bgrid$LL <- numeric(nrow(Bgrid))
Bgrid$LL[] <- NA

for (chem in Chemicals) {
  if (file.exists(file.path("Skeletons",paste(chem,"rda",sep=".")))) next
  f <- file.path("../../pre-Feb-02/BasicModelFits/Basic-Model",
                 paste(chem,"rda",sep="."))
  if (file.exists(f)) load(f) else xx <- NULL
  xxsave <- xx
  seldata <- BRAINdata[sel <- (BRAINdata$chemical == chem),]
  seldata$s.M.t <- factor(seldata$s.M.t)
  seldata$sex <- factor(seldata$sex)
  seldata$block <- factor(seldata$block)

### loop through seldata (by block), using PhonyDF to expand it to
### synthetic individual data.


  tmp <- by(seldata,list(seldata$block),function(x) {
    PhonyDF(x$dose,x$n,x$chei,x$sd,"Dose","AChE",
            chem=rep(as.character(x$chunit[1]),nrow(x)),
            ChemName="Unit")
  })

  for (i in seq(along=tmp)) {
    flds <- unlist(strsplit(names(tmp[i]),"-"))
    N <- nrow(tmp[[i]])
    tmp[[i]]$mrid <- rep(flds[2],N)
    tmp[[i]]$sex <- rep(flds[4],N)
    tmp[[i]]$set <- rep(paste(flds[2],flds[5],flds[6],sep=":"),N)
    tmp[[i]]$s.M.t <- paste(flds[4],flds[2],flds[5],sep="-")
  }

  Pseldata <- do.call("rbind",unclass(tmp))
  row.names(Pseldata) <- seq(along=Pseldata[[1]])
  Pseldata$set <- factor(Pseldata$set)
  Pseldata$s.M.t <- factor(Pseldata$s.M.t)
  Respscale <- max(seldata$chei)
  Dosescale <- max(seldata$dose)

  Pseldata$AChE.scaled <- Pseldata$AChE/Respscale
  Pseldata$Dose.scaled <- Pseldata$Dose/Dosescale
### Random Effects
  RnDoM1 <- if (length(levels(Pseldata$mrid)) == 1) {
    if (length(levels(Pseldata$set)) == 1) {
      NULL
    } else {
      list(set=pdDiag(form=BMD ~ 1)) ## BMD~1 | set
    }
  } else {
    if (length(levels(Pseldata$set)) > length(levels(Pseldata$mrid))) {
      list(mrid=pdDiag(form=BMD ~ 1),
```

```
            set=pdDiag(form=BMD~1)) ## BMD~1|mrid/set
    } else {
      list(mrid=pdDiag(form=BMD ~ 1)) ## BMD~1|mrid
    }
  }



  xx <- list(Chemical=chem,Start=getparms(xxsave,seldata),Data=seldata,
             Pdata=Pseldata,Respscale=Respscale,
             Dosescale=Dosescale,Random=RnDoM1)
  save(xx,file=file.path("Skeletons",paste(chem,"rda",sep=".")))
}
### master.R
### Assume that pvm has already been started, and assign work to each node
### in the cluster.  This is a general purpose master; be sure to change
### the value of slavename.  Also, the value of Chemicals can be reassigned
### if you want to do just a subset.  All the real work is done in the slave
### program.


require(rpvm)

attach("/home/setzer/tasks/CumRisk/post-Feb-SAP/Data/opdata.rda")
griddb <- read.csv("griddb.csv",row.names=1)
Chemicals <- row.names(griddb[griddb$Rerun == 1,])
cat("Running for Chemicals:\n")
print(Chemicals)

MOREWORK <- 22
TASKFAIL <- 99
HERESWORK <- 33
workingdir <- outputdir <- getwd()
Hosts <- "/home/setzer/.xpvm_hosts"
print(.PVM.start.pvmd(hosts=Hosts,block=TRUE))
### Here is where changes are most likely

slavename <- "slave" # name of program to do the work for a particular
                     # chemical


### To here


cfg <- .PVM.config()

nodenames <- as.character(cfg$name)

#ntasks <- max(min(nrow(cfg),length(Chemicals)-1),1)
ntasks <- min(nrow(cfg),length(Chemicals))

mytid <- .PVM.mytid()

children <- NULL
# for (i in seq(along=ntasks)) {
#   tmp <- .PVM.spawn(task="slaveR.sh",ntask=ntasks[i],
#                     flag="Host", where=nodenames[i],
#                     arglist=c(slavename,workingdir,
#                     outputdir))
#   cat(tmp, fill=TRUE)
#   if (length(tmp) > 0) children <- c(children,tmp)
# }
children <- .PVM.spawn(task="slaveR.sh",ntask=ntasks,
                     flag="Default",
```

```
                        arglist=c(slavename,workingdir,
                          outputdir))

Sys.sleep(10)

if (length(Chemicals) > 1) {
  tmp <- .PVM.spawn(task="slaveR.sh",ntask=1,flag="Host",
                    where="gandalf.localdomain",
                    arglist=c(slavename,workingdir,outputdir))
  children <- c(children,tmp)
}
### Check for and delete any -1's (system errors)

if (any(children == -1)) {
  warning(paste(sum(children == -1),"failed starts out of",length(children),
                "potential"))
  children <- children[children != -1]
}

if ((Nchild <- length(children)) == 0) stop("No children started\n")

### Get the nodenames of the tasks

NodeNames <- character(length(children))
for (i in seq(along=NodeNames))
  NodeNames[i] <-
as.character(cfg$name[match(.PVM.tasks(where=children[i])$host,
                                  cfg$host.id)])

### Request notification of task exiting

.PVM.notify(msgtag=TASKFAIL,what="ExitTask",children)

### Start looping

i <- 0
j <- 1
Nrunning <- Nchild
repeat {
  i <- (i %% Nchild) + 1
  while (buf <- .PVM.nrecv(-1, TASKFAIL) > 0) {
    tmp <- .PVM.upkint()
    kk <- match(tmp, children)
    if (!is.na(kk)) {
      cat(paste("\n>>> Task",tmp,"on",NodeNames[kk],"has exited\n"))
      Sys.sleep(5)
      children[kk] <- .PVM.spawn(task="slaveR.sh",ntask=1,
                                  flag="Host", where=NodeNames[kk],
                                  arglist=c(slavename,workingdir,
                                    outputdir))
      cat(paste("Replaced by task",children[kk],"\n\n"))
      Sys.sleep(3)
      .PVM.notify(msgtag=TASKFAIL,what="ExitTask",children[kk])
      next
    }
  }
  buf <- .PVM.nrecv(children[i], MOREWORK)
  if ( buf > 0) {
#    tmp <- .PVM.upkstrvec()
#    cat(paste("\n++",tmp,"completed at",date(),"\n"))
    if (j <= length(Chemicals)) {
      cat(paste("\n++ Sending",Chemicals[j],"to task",children[i],
                "on",NodeNames[i],"at",date(),"\n"))
      .PVM.initsend()
      .PVM.pkstrvec(Chemicals[j])
```

```
      .PVM.send(children[i],HERESWORK)
      j <- j + 1
    } else {
      Nrunning <- Nrunning - 1
    }
  }
  if (Nrunning <= 0) {
    break
  }
}
cat(paste("\n!!!!!! All Done!",date(),"\n"))
for (i in seq(along=children)) .PVM.kill(children[i])

rm(Chemicals) ## so we can use the one in opdata.rda
source("plotProfiles.R")

.PVM.exit()



### slave.R
### Profile likelihoods for exponential model and male,female values of B
###
### Get the command line arguments (to specify which chemicals to run):


invisible(options(echo=FALSE))
require(rpvm)
cat("\n================================\n")
mytid <- .PVM.mytid()
cat(paste("I am task",mytid,"on",system("uname -n",intern=TRUE),"\n"))
cat("\n================================\n\n")

attach("/home/setzer/tasks/CumRisk/post-Feb-SAP/Data/opdata.rda")
require(RBMDS)
require(nlme)
setwd("/home/setzer/tasks/CumRisk/post-Feb-SAP/ProfilesForB")
savepath <- "Fits"
griddb <- read.csv("griddb.csv",row.names=1)
MOREWORK <- 22
TASKFAIL <- 99
HERESWORK <- 33


myparent <- .PVM.parent()
chem <- "START"

### BMR is set here
BMR <- 0.1

if (myparent <= 0) stop("PVM error!")

repeat {
  .PVM.initsend()
#  .PVM.pkstrvec(chem)
  .PVM.send(myparent,MOREWORK)
  buf <- .PVM.recv(myparent,HERESWORK)
  chem <- .PVM.upkstrvec()
  cat(paste("-------\n",chem,"\n\n"))
  OldModel <- file.path("Skeletons", paste(chem,"rda",sep="."))
  load(OldModel)
  Firststart<- c(xx$Start[["A"]] - log(xx$Respscale),
                 xx$Start[["BMD"]] - log(xx$Dosescale))
  start <- Firststart
### Compute the grid of values.
```

```
  for (i in 1:nrow(xx$Bgrid)) {
    xx$Models[[i]] <- eval(substitute(AChE.scaled ~
CexpBwDS(Dose.scaled,A,BMD,sex,
                                                    fixed=x),

list(x=list(B=c(F=as.vector(xx$Bgrid$B.F[i]),

M=as.vector(xx$Bgrid$B.M[i])))))))
  }

  Mrids <- unique(xx$Data$mrid)
  if (length(Mrids) > 1) {
    Power <- rep(0.5, length(Mrids))
    names(Power) <- as.character(Mrids)
    Weights <- varComb(varIdent(form=~1|factor(mrid)),
                       varPower(form=~fitted(.)|factor(mrid),
                                fixed=Power))
  } else {
    Weights <- varPower(form=~fitted(.),fixed=1)
  }

  for (i in 1:nrow(xx$Bgrid)) {

cat(paste("\n------\n",i,":",xx$Bgrid$B.F[i],xx$Bgrid$B.M[i],date(),"\n\n"))
    Model <- xx$Models[[i]]
    RnDoM1 <- xx$Random


     xx$Fits[[i]] <- if (!is.null(RnDoM1)) {
         try(eval(substitute(nlme(Model,data=xx$Pdata,
                                  fixed=list(A ~ s.M.t - 1, BMD ~ sex-1),
                                  random=xxxx,
                                  start=start,
                                  weights=Weights,
                                  method="ML"),

list(Model=Model,xxxx=RnDoM1,Weights=Weights))))
     } else {
        try(eval(substitute(gnls(Model, data=xx$Pdata,
                                 params=list(A ~ s.M.t - 1, BMD ~ sex-1),
                                 start=start,
                                 weights=Weights),
                            list(Model=Model,Weights=Weights))))
    }
    xx$Bgrid$LL[i] <-
      if (!inherits(xx$Fits[[i]], "try-error")) logLik(xx$Fits[[i]]) else NA
    j <- if (i %% xx$Nsteps == 0) {
      i - xx$Nsteps + 1
    } else {
      i
    }
    start <- if (!inherits(xx$Fits[[j]],"try-error")) {
      if (inherits(xx$Fits[[j]],"nlme")) xx$Fits[[j]]$coefficients$fixed
      else xx$Fits[[j]]$coefficients
    } else Firststart

    save(xx,file=file.path(savepath,paste(chem,"rda",sep=".")))
  }
  cat(paste("-------",chem,"complete",date(),"\n\n"))
}


### makefinegriddb.R
loadpath <- "FineFits"
griddb <- read.csv("griddb.csv",row.names=1)
```

```
attach("../Data/opdata.rda")

griddb$Npoints[] <- 11
griddb$Rerun[] <- 1

for (chem in Chemicals) {
  load(file.path(loadpath,paste(chem,"rda",sep=".")))
  griddb[chem,"B.Fmin"] <- min(xx$Bgrid$B.F)
  griddb[chem,"B.Fmax"] <- max(xx$Bgrid$B.F)
  griddb[chem,"B.Mmin"] <- min(xx$Bgrid$B.M)
  griddb[chem,"B.Mmax"] <- max(xx$Bgrid$B.M)
}

write.table(griddb,file="finegriddb.csv",sep=",",col.names=NA)
```

e. Part 3b: Expanded model.

For profile likelihood estimation of S and D in the expanded model, first 'master.R' and 'slave.R' are run, which use the best values found in the estimation of P_B for the basic model as initial values. Then successive versions of 'master' and 'slave' are produced (see below, 'finemaster.R', 'fineslave.R') that result from successive refinements of the grid.

```
### master.R
### Assume that pvm has already been started, and assign work to each node
### in the cluster.  This is a general purpose master; be sure to change
### the value of slavename.  Also, the value of Chemicals can be reassigned
### if you want to do just a subset.  All the real work is done in the slave
### program.


require(rpvm)

attach("/home/setzer/tasks/CumRisk/post-Feb-SAP/Data/opdata.rda")
griddb <- read.csv("griddb.csv",row.names=1)
Chemicals <- row.names(griddb[griddb$DoIt == 1,])
cat("Running for Chemicals:\n")
print(Chemicals)

MOREWORK <- 22
TASKFAIL <- 99
HERESWORK <- 33
GETSTARTED <- 1

workingdir <- outputdir <- getwd()
Hosts <- "/home/setzer/.xpvm_hosts"
#print(.PVM.start.pvmd(hosts=Hosts,block=TRUE))
#Sys.sleep(10)
### Here is where changes are most likely

slavename <- "slave" # name of program to do the work for a particular
                     # chemical


### To here
```

```
cfg <- .PVM.config()

nodenames <- as.character(cfg$name)

#ntasks <- max(min(nrow(cfg),length(Chemicals)-1),1)
ntasks <- min(nrow(cfg),length(Chemicals))

mytid <- .PVM.mytid()
#children <- NULL
# for (i in seq(along=ntasks)) {
#   tmp <- .PVM.spawn(task="slaveR.sh",ntask=ntasks[i],
#                     flag="Host", where=nodenames[i],
#                     arglist=c(slavename,workingdir,
#                     outputdir))
#   cat(tmp, fill=TRUE)
#   if (length(tmp) > 0) children <- c(children,tmp)
# }
children <- .PVM.spawn(task="slaveR.sh",ntask=ntasks,
                     flag="Default",
                     arglist=c(slavename,workingdir,
                       outputdir))
Sys.sleep(10)
print(children)

if (length(Chemicals) > 1) {
  tmp <- .PVM.spawn(task="slaveR.sh",ntask=1,flag="Host",
                    where="gandalf.localdomain",
                    arglist=c(slavename,workingdir,outputdir))
  children <- c(children,tmp)
}
Sys.sleep(10)
print(children)
### Check for and delete any -1's (system errors)

if (any(children == -1)) {
  warning(paste(sum(children == -1),"failed starts out of",length(children),
                "potential"))
  children <- children[children != -1]
}

if ((Nchild <- length(children)) == 0) stop("No children started\n")

### Get the nodenames of the tasks

NodeNames <- character(length(children))
for (i in seq(along=NodeNames)) {
  NodeNames[i] <-
as.character(cfg$name[match(.PVM.tasks(where=children[i])$host,
                                  cfg$host.id)])
}
### Request notification of task exiting

.PVM.notify(msgtag=TASKFAIL,what="ExitTask",children)

### Start them

.PVM.initsend()
.PVM.pkintvec(1:3)
.PVM.mcast(children,GETSTARTED)

### Start looping

i <- 0
j <- 1
```

```
Nrunning <- Nchild
repeat {
  i <- (i %% Nchild) + 1
  while (buf <- .PVM.nrecv(-1, TASKFAIL) > 0) {
    tmp <- .PVM.upkint()
    kk <- match(tmp, children)
    if (!is.na(kk)) {
      cat(paste("\n>>> Task",tmp,"on",NodeNames[kk],"has exited\n"))
      Sys.sleep(5)
      children[kk] <- .PVM.spawn(task="slaveR.sh",ntask=1,
                                 flag="Host", where=NodeNames[kk],
                                 arglist=c(slavename,workingdir,
                                   outputdir))
      cat(paste("Replaced by task",children[kk],"\n\n"))
      Sys.sleep(3)
      .PVM.notify(msgtag=TASKFAIL,what="ExitTask",children[kk])
      next
    }
  }
  buf <- .PVM.nrecv(children[i], MOREWORK)
  if ( buf > 0) {
    tmp <- .PVM.upkstrvec()
    cat(paste("\n++",tmp,"completed at",date(),"\n"))
    if (j <= length(Chemicals)) {
      cat(paste("\n++ Sending",Chemicals[j],"to task",children[i],
                "on",NodeNames[i],"at",date(),"\n"))
      .PVM.initsend()
      .PVM.pkstrvec(Chemicals[j])
      .PVM.send(children[i],HERESWORK)
      j <- j + 1
    } else {
      Nrunning <- Nrunning - 1
    }
  }
  if (Nrunning <= 0) {
    break
  }
}
cat(paste("\n!!!!!! All Done!",date(),"\n"))
for (i in seq(along=children)) .PVM.kill(children[i])

rm(Chemicals) ## so we can use the one in opdata.rda
source("plotProfiles.R")

.PVM.exit()
### slave.R

invisible(options(echo=FALSE))
require(rpvm)
cat("\n================================\n")
mytid <- .PVM.mytid()
cat(paste("I am task",mytid,"on",system("uname -n",intern=TRUE),"\n"))
cat("\n================================\n\n")



attach("/home/setzer/tasks/CumRisk/post-Feb-SAP/Data/opdata.rda")
require(RBMDS)
require(nlme)
setwd("/home/setzer/tasks/CumRisk/post-Feb-SAP/ProfilesForSD")
savepath <- "Fits"

griddb <- read.csv("griddb.csv",row.names=1)
### Walks diagonals of a grid, starting at the upper right hand corner.
walkgrid <- function(Nsteps) {
```

```
  mx <- matrix(1:(Nsteps*Nsteps),nrow=Nsteps,byrow=TRUE)
  GridIndex <- numeric(Nsteps*Nsteps)
  indx <- 1
  for (i in 1:Nsteps) {
    for (k in 1:i) {
      if (i %% 2 == 1) {
        ii <- i - k + 1
        jj <- Nsteps - k + 1
      } else {
        ii <- k
        jj <- Nsteps - i + k
      }
      GridIndex[indx] <- mx[ii,jj]
      indx <- indx + 1
    }
  }
  for (i in 2:Nsteps) {
    for (k in i:Nsteps) {
      if ((Nsteps - i + 1) %% 2 == 1) {
        ii <- Nsteps + i - k
        jj <- Nsteps - k + 1

      } else {
        ii <- k
        jj <- k - i + 1
      }
      GridIndex[indx] <- mx[ii,jj]
      indx <- indx + 1
    }
  }

  GridIndex
}

MOREWORK <- 22
TASKFAIL <- 99
HERESWORK <- 33
GETSTARTED <- 1
myparent <- .PVM.parent()

chem <- "START"
.PVM.recv(myparent, msgtag=GETSTARTED)
tmp <- .PVM.upkintvec()

if (myparent <=  0) stop("PVM error")
repeat {
  .PVM.initsend()
  .PVM.pkstrvec(chem)
  .PVM.send(myparent,MOREWORK)
  buf <- .PVM.recv(myparent,HERESWORK)
  chem <- .PVM.upkstrvec()
  fname <- paste(chem,"rda",sep=".")
  cat(paste("\n-----------------",chem,"--------------------\n\n"))

  load(file.path("../ProfilesForB/FinerFits",fname))

### Get the best fitting model
  indx <- which.max(xx$Bgrid$LL)
  StartLL <- xx$Bgrid$LL[indx]

### Get initial start value, the coefficients from xx$Fit[[indx]]
  start <- if (inherits(xx$Fit[[indx]], "nlme")) {
    xx$Fit[[indx]]$coefficients$fixed
  } else {
    xx$Fit[[indx]]$coefficients
```

```
  }
Bests <- as.vector(c(xx$Bgrid[indx,"B.F"],xx$Bgrid[indx,"B.M"]))

### Get the model for the random parameters used in the basic model
  RandomParms <- xx$Random


  Pseldata <- xx$Pdata

### Compute the grid of values.

  SDgrid <- expand.grid(S = seq(griddb[chem,"Smin"],
                          griddb[chem,"Smax"],
                          length=griddb[chem,"Npoints"]),
                        D = seq(griddb[chem,"Dmin"],
                          griddb[chem, "Dmax"],
                          length=griddb[chem,"Npoints"]))
  SDgrid$LL <- numeric(nrow(SDgrid))
  SDgrid$LL[] <- as.numeric(NA)
  Fits <- vector("list",nrow(SDgrid))
  GridIndex <- walkgrid(griddb[chem,"Npoints"])

  Mrids <- unique(xx$Data$mrid)
  if (length(Mrids) > 1) {
    Power <- rep(0.5, length(Mrids))
    names(Power) <- as.character(Mrids)
    Weights <- varComb(varIdent(form=~1|factor(mrid)),
                       varPower(form=~fitted(.)|factor(mrid),
                                fixed=Power))
  } else {
    Weights <- varPower(form=~fitted(.),fixed=1)
  }

  xx$Fit <- Fits
  xx$Bgrid <- NULL
  xx$SDgrid <- SDgrid

  ## If our Dmin exceeds 0.001, walk up from 0.001 to Dmin in steps of 0.05

  if (griddb[chem,"Dmin"] > 0.001) {
    cat("\n------- Walking up to the parameter grid -----\n")
    Dseq <- seq(0.001, griddb[chem,"Dmin"], by=0.05)
    Smax <- griddb[chem,"Smax"]

    for (i in 1:length(Dseq)) {
      cat("\n-------------\n")
      cat(paste("i:",i,"D[i]:",Dseq[i],", S:",Smax,"\n"))
      cat("\n-------------\n")
      ## set up the model
       Model <- eval(substitute(AChE.scaled ~
CpkexpB2wDS(Dose.scaled,A,BMD,sex,

fixed=list(PB=c(F=xxxx,

M=yyyy),
                                                          S=c(F=zzzz,

M=zzzz),
                                                          D=c(F=wwww,

M=wwww))),
                                  list(xxxx=Bests[1],yyyy=Bests[2],
                                       zzzz=log(Smax),
                                       wwww=log(Dseq[i]))))
     ## estimate it
      if (!is.null(RandomParms)) {
```

```
        try(fitpk <-
            eval(substitute(nlme(Model,data=Pseldata,
                                 fixed=list(A ~ s.M.t - 1, BMD ~ sex - 1),
                                 random=xxxx,
                                 start=zzzz,
                                 weights=Weights,
                                 method="ML"),
                            list(Model=Model,xxxx=RandomParms,
                                 zzzz=start,
                                 Weights=Weights))))
      } else {
        fitpk <-
          try(eval(substitute(gnls(Model, data=Pseldata,
                                   params=list(A ~ s.M.t - 1, BMD ~ sex -
1),
                                   start=zzzz,
                                   weights=Weights),
                              list(Model=Model,
                                   zzzz=start,
                                   Weights=Weights))))
      }
      if (!inherits(fitpk, "try-error")) {
        ## Use the current fit to give the start value for the next
        start <- if (inherits(fitpk, "nlme")) {
          fitpk$coefficients$fixed
        } else {
          fitpk$coefficients
        }
      }
    }
  }

  for (iii in 1:nrow(SDgrid)) {
    i <- GridIndex[iii]
    cat("\n-------------\n")
    cat(paste("i:",i,"D[i]:",SDgrid$D[i],", S[i]:",SDgrid$S[i],date(),"\n"))
    cat("\n-------------\n")

    ## set up the model
    Model <- eval(substitute(AChE.scaled ~
CpkexpB2wDS(Dose.scaled,A,BMD,sex,

fixed=list(PB=c(F=xxxx,
                                                                  M=yyyy),
                                                        S=c(F=zzzz, M=zzzz),
                                                        D=c(F=wwww,
M=wwww))),
                              list(xxxx=Bests[1],yyyy=Bests[2],
                                   zzzz=log(SDgrid$S[i]),
                                   wwww=log(SDgrid$D[i]))))
    ## estimate it
    if (!is.null(RandomParms)) {
      try(fitpk <-
        eval(substitute(nlme(Model,data=Pseldata,
                                 fixed=list(A ~ s.M.t - 1, BMD ~ sex - 1),
                                 random=xxxx,
                                 start=zzzz,
                                 weights=Weights,
                                 method="ML"),
                            list(Model=Model,xxxx=RandomParms,
                                 zzzz=start,
                                 Weights=Weights))))
     } else {
       fitpk <-
        try(eval(substitute(gnls(Model, data=Pseldata,
```

```
                                    params=list(A ~ s.M.t - 1, BMD ~ sex - 1),
                                    start=zzzz,
                                    weights=Weights),
                                list(Model=Model,
                                    zzzz=start,
                                    Weights=Weights))))
  }

    if (!inherits(fitpk, "try-error")) {
      xx$SDgrid$LL[i] <- logLik(fitpk)
      cat(paste("\nLL:",xx$SDgrid$LL[i],"\n"))
      ## Use the current fit to give the start value for the next
      start <- if (inherits(fitpk, "nlme")) {
        fitpk$coefficients$fixed
      } else {
        fitpk$coefficients
      }
    }
    xx$Fit[[i]] <- fitpk
    save(xx,file=file.path("Fits",paste(chem,"rda",sep=".")))
  }
  cat(paste(chem,"finished",date(),"\n\n"))

}
### plotProfiles.R
require(akima)
pdf(file="Profiles-4-SD2.pdf")
par(xpd=NA)
dirname <- "Fits"
attach("../Data/opdata.rda")
for (chem in Chemicals){
  fname <- paste(chem,"rda",sep=".")
  load(file.path("../ProfilesForB/FinerFits",fname))
  B.BestLL <- max(xx$Bgrid$LL,na.rm=TRUE)
  xx <- list()
  ## Get the best-fitting Basic model

  if (!file.exists(file.path(dirname,fname))) next
  tmp <- try(load(file.path(dirname,fname)))
  if (inherits(tmp,"try-error")) {
    cat(paste("Problem reading",chem,"\n"))
    next
  }
  LLgrid <- xx$SDgrid
  LLgrid <- na.omit(LLgrid)
  if (length(LLgrid$LL) == 0) {
    plot(c(0,1),c(0,1),type="n",axes=FALSE, xlab="", ylab="")
    text(0.5,0.5,paste(chem,"no fits"),adj=0.5)
  } else {
    indx <- which.max(LLgrid$LL)
    BestLL <- max(LLgrid$LL[indx],B.BestLL)
    ScaledLL <- 2*(BestLL - LLgrid$LL)
    out <- try(interp(LLgrid$S,LLgrid$D,ScaledLL,
                  xo=seq(min(LLgrid$S), max(LLgrid$S),length=100),
                  yo=seq(min(LLgrid$D), max(LLgrid$D),length=100)))
    if (!inherits(out, "try-error")) {
      res <- try(image(out,xlab="S",ylab="D",main=chem,
                       col = rev(heat.colors(9)),
                       breaks =
c(0,qchisq(c(0.05,.10,.25,.50,.75,.90,.95,.99),2),1e26)))
      if (inherits(res, "try-error")) {
        plot(c(0,1),c(0,1),type="n",axes=FALSE, xlab="", ylab="")
        text(0.5,0.5,paste(chem,"not enough fits"),adj=0.5)
      } else {
        critX2 <- qchisq(0.95,2)
```

```
        points(LLgrid$S,LLgrid$D,
               pch=ifelse(2*(BestLL - LLgrid$LL) < critX2, 19, 3))
        points(LLgrid$S[indx],LLgrid$D[indx],pch=4,cex=1.5)
      }
    } else {
    plot(c(0,1),c(0,1),type="n",axes=FALSE, xlab="", ylab="")
    text(0.5,0.5,paste(chem,"not enough fits"),adj=0.5)
    }
  }
}
dev.off()

### makeFineGrid.R --
### Uses the information in the files in ./Fits to create a new set of
templates
### in FineFits.  Each new template already contains SDgrid and Fits
###
### The best expanded fit for the following chemicals is essentially the
basic model,
### so they are excluded from further action.  The criteria were:
### 1) Pvalue for the difference in log likelihoods was greater than 0.05
AND
### 2) BOTH BMDs were no more than 10% different between the expanded and
basic models
###    with the expanded model being the point of comparison (in the
denominator).

K2IJ <- function(k, Npoints) {
  c(Sindx = (i <- ((k-1)%%Npoints) + 1),
    Dindx = (k - i)/Npoints + 1)
}

IJ2K <- function(i, j, Npoints) {
  i + Npoints * (j - 1)
}

indxmin <- function(i, N) {
  if (i == 1) i else i - 1
}
indxmax <- function(i, N) {
  if (i == N) i else i + 1
}
Neighbors <- function(k, Npoints) {
  ij <- K2IJ(k, Npoints)
  i <- ij[1]
  j <- ij[2]
  iseq <- indxmin(i,Npoints):indxmax(i,Npoints)
  jseq <- indxmin(j, Npoints):indxmax(j, Npoints)

as.vector(apply(data.matrix(expand.grid(i=iseq,j=jseq)),1,function(x)IJ2K(x[
1],x[2],Npoints)))
}


require(RBMDS)
require(nlme)

DropChems <-
c("ACEPHATE","CHLORPYRIPHOSMETHYL","DICROTOPHOS","DIMETHOATE","ETHOPROP",

"FENTHION","METHAMIDOPHOS","METHIDATHION","NALED","OXYDEMETONMETHYL",
                "PIRIMIPHOSMETHYL","PROFENOFOS")

attach("../Data/opdata.rda")
savedir <- "Skel2"
```

```
finegriddb <- read.csv("griddb.csv",row.names=1)

finegriddb[DropChems,"DoIt"] <- 0
finegriddb$Npoints[] <- 5

Dochem <- which(finegriddb$DoIt == 1)


for (chem in Chemicals) {
  fname <- paste(chem,"rda",sep=".")
  load(file.path("Fits",fname))
  ##  'xx' is the new version
  if (chem %in% DropChems) {
    save(xx, file=file.path("FineFits",fname))
    next
  } else {
    oldxx <- xx
    xx$Nsteps <- 5
    xx$Fit <- list()
    Slist <- sort(unique(oldxx$SDgrid$S))
    Dlist <- sort(unique(oldxx$SDgrid$D))

    ## Build the new SDgrid centered around the maximum LL on the old one
    indx <- which.max(oldxx$SDgrid$LL)
    SDindx <- K2IJ(indx,oldxx$Nsteps)
    Smin <- Slist[indxmin(SDindx[1],oldxx$Nsteps)]
    Smax <- Slist[indxmax(SDindx[1],oldxx$Nsteps)]
    Dmin <- Dlist[indxmin(SDindx[2],oldxx$Nsteps)]
    if (SDindx[2] < oldxx$Nsteps) {
      Dmax <- Dlist[indxmax(SDindx[2], oldxx$Nsteps)]
    } else {
      ## If the old D was on the upper border of the grid, expand it by one
step
      Delta <- (max(Dlist) - min(Dlist))/oldxx$Nsteps
      Dmax <- max(Dlist) + Delta
    }
    finegriddb[chem,"Smin"] <- Smin
    finegriddb[chem,"Smax"] <- Smax
    finegriddb[chem,"Dmin"] <- Dmin
    finegriddb[chem,"Dmax"] <- Dmax
    xx$SDgrid <- expand.grid(S=seq(Smin,Smax,length=xx$Nsteps),
                             D=seq(Dmin,Dmax,length=xx$Nsteps))
    xx$SDgrid$LL <- numeric(nrow(xx$SDgrid))
    xx$SDgrid$LL[] <- NA
    xx$Start <- if (inherits(oldxx$Fit[[indx]],"nlme")) {
      oldxx$Fit[[indx]]$coefficients$fixed
    } else {
      oldxx$Fit[[indx]]$coefficients
    }

    xx$Fit <- vector("list",nrow(xx$SDgrid))

    ## Finally, fill in the LLs and Fits we already know.
    ## What are all the indexes?
    nearby <- Neighbors(indx,oldxx$Nsteps)

    ## There must be a better way to do this, but I'm tired ... .
    for (i in seq(along=nearby)) {
      K <- nearby[i]
      a1 <- sapply(xx$SDgrid$S,
                   function(x)identical(all.equal(x,
oldxx$SDgrid$S[K]),TRUE))
      a2 <- sapply(xx$SDgrid$D,
                   function(x)identical(all.equal(x,
oldxx$SDgrid$D[K]),TRUE))
```

```
      j <- which(a1 & a2)
      if (length(j) == 1) {
        xx$SDgrid$LL[j] <- oldxx$SDgrid$LL[K]
        xx$Fit[[j]] <- oldxx$Fit[[K]]
      }
    }

    save(xx,file=file.path(savedir,fname))
  }
}

write.table(finegriddb,file="finegriddb.csv",sep=",",col.names=NA)

### master.R
### Assume that pvm has already been started, and assign work to each node
### in the cluster.  This is a general purpose master; be sure to change
### the value of slavename.  Also, the value of Chemicals can be reassigned
### if you want to do just a subset.  All the real work is done in the slave
### program.


require(rpvm)

attach("/home/setzer/tasks/CumRisk/post-Feb-SAP/Data/opdata.rda")
griddb <- read.csv("finegriddb.csv",row.names=1)
Chemicals <- row.names(griddb[griddb$DoIt == 1,])
cat("Running for Chemicals:\n")
print(Chemicals)

MOREWORK <- 22
TASKFAIL <- 99
HERESWORK <- 33
GETSTARTED <- 1

workingdir <- outputdir <- getwd()
Hosts <- "/home/setzer/.xpvm_hosts"
#print(.PVM.start.pvmd(hosts=Hosts,block=TRUE))
#Sys.sleep(10)
### Here is where changes are most likely

slavename <- "fineslave" # name of program to do the work for a particular
                         # chemical


### To here


cfg <- .PVM.config()

nodenames <- as.character(cfg$name)

#ntasks <- max(min(nrow(cfg),length(Chemicals)-1),1)
ntasks <- min(nrow(cfg),length(Chemicals))

mytid <- .PVM.mytid()
#children <- NULL
# for (i in seq(along=ntasks)) {
#   tmp <- .PVM.spawn(task="slaveR.sh",ntask=ntasks[i],
#                     flag="Host", where=nodenames[i],
#                     arglist=c(slavename,workingdir,
#                     outputdir))
#   cat(tmp, fill=TRUE)
#   if (length(tmp) > 0) children <- c(children,tmp)
# }
children <- .PVM.spawn(task="slaveR.sh",ntask=ntasks,
```

```
                        flag="Default",
                        arglist=c(slavename,workingdir,
                          outputdir))
Sys.sleep(10)

if (length(Chemicals) > 1) {
  tmp <- .PVM.spawn(task="slaveR.sh",ntask=1,flag="Host",
                    where="gandalf.localdomain",
                    arglist=c(slavename,workingdir,outputdir))
  children <- c(children,tmp)
}
Sys.sleep(10)

### Check for and delete any -1's (system errors)

if (any(children == -1)) {
  warning(paste(sum(children == -1),"failed starts out of",length(children),
                "potential"))
  children <- children[children != -1]
}

if ((Nchild <- length(children)) == 0) stop("No children started\n")

### Get the nodenames of the tasks

NodeNames <- character(length(children))
for (i in seq(along=NodeNames)) {
  NodeNames[i] <-
as.character(cfg$name[match(.PVM.tasks(where=children[i])$host,
                                  cfg$host.id)])
}
### Request notification of task exiting

.PVM.notify(msgtag=TASKFAIL,what="ExitTask",children)

### Start them

.PVM.initsend()
.PVM.pkintvec(1:3)
.PVM.mcast(children,GETSTARTED)

### Start looping

i <- 0
j <- 1
Nrunning <- Nchild
repeat {
  i <- (i %% Nchild) + 1
  while (buf <- .PVM.nrecv(-1, TASKFAIL) > 0) {
    tmp <- .PVM.upkint()
    kk <- match(tmp, children)
    if (!is.na(kk)) {
      cat(paste("\n>>> Task",tmp,"on",NodeNames[kk],"has exited\n"))
      Sys.sleep(5)
      children[kk] <- .PVM.spawn(task="slaveR.sh",ntask=1,
                                 flag="Host", where=NodeNames[kk],
                                 arglist=c(slavename,workingdir,
                                   outputdir))
      cat(paste("Replaced by task",children[kk],"\n\n"))
      Sys.sleep(3)
      .PVM.notify(msgtag=TASKFAIL,what="ExitTask",children[kk])
      next
    }
  }
  buf <- .PVM.nrecv(children[i], MOREWORK)
```

```
  if ( buf > 0) {
    tmp <- .PVM.upkstrvec()
    cat(paste("\n++",tmp,"completed at",date(),"\n"))
    if (j <= length(Chemicals)) {
      cat(paste("\n++ Sending",Chemicals[j],"to task",children[i],
                "on",NodeNames[i],"at",date(),"\n"))
      .PVM.initsend()
      .PVM.pkstrvec(Chemicals[j])
      .PVM.send(children[i],HERESWORK)
      j <- j + 1
    } else {
      Nrunning <- Nrunning - 1
    }
  }
  if (Nrunning <= 0) {
    break
  }
}
cat(paste("\n!!!!!! All Done!",date(),"\n"))
for (i in seq(along=children)) .PVM.kill(children[i])

rm(Chemicals) ## so we can use the one in opdata.rda
source("plotProfiles.R")

.PVM.exit()
### slave.R

invisible(options(echo=FALSE))
require(rpvm)
cat("\n=================================\n")
mytid <- .PVM.mytid()
cat(paste("I am task",mytid,"on",system("uname -n",intern=TRUE),"\n"))
cat("\n=================================\n\n")



attach("/home/setzer/tasks/CumRisk/post-Feb-SAP/Data/opdata.rda")
require(RBMDS)
require(nlme)
setwd("/home/setzer/tasks/CumRisk/post-Feb-SAP/ProfilesForSD")
savepath <- "FineFits"

### Walks diagonals of a grid, starting at the upper right hand corner.

MOREWORK <- 22
TASKFAIL <- 99
HERESWORK <- 33
GETSTARTED <- 1
myparent <- .PVM.parent()

chem <- "START"
.PVM.recv(myparent, msgtag=GETSTARTED)
tmp <- .PVM.upkintvec()

if (myparent <=  0) stop("PVM error")
repeat {
  .PVM.initsend()
  .PVM.pkstrvec(chem)
  .PVM.send(myparent,MOREWORK)
  buf <- .PVM.recv(myparent,HERESWORK)
  chem <- .PVM.upkstrvec()
  fname <- paste(chem,"rda",sep=".")
  cat(paste("\n-----------------",chem,"--------------------\n\n"))

  load(file.path("Skel2",fname))
```

```
### Get initial start value, the coefficients from xx$Fit[[indx]]
  start <- xx$Start

### and the estimates of PB to use
  indx <- which.max(xx$SDgrid$LL)
  Bests <- as.vector(eval(xx$Fit[[indx]]$call$model[[3]]$fixed$PB))

### Get the model for the random parameters used in the basic model
  RandomParms <- xx$Random


  Pseldata <- xx$Pdata


  Mrids <- unique(xx$Data$mrid)
  if (length(Mrids) > 1) {
    Power <- rep(0.5, length(Mrids))
    names(Power) <- as.character(Mrids)
    Weights <- varComb(varIdent(form=~1|factor(mrid)),
                       varPower(form=~fitted(.)|factor(mrid),
                                fixed=Power))
  } else {
    Weights <- varPower(form=~fitted(.),fixed=1)
  }

  for (i in 1:nrow(xx$SDgrid)) {
    if (!is.na(xx$SDgrid$LL[i])) next
    cat("\n------------\n")
    cat(paste("i:",i,"D[i]:",xx$SDgrid$D[i],",
S[i]:",xx$SDgrid$S[i],date(),"\n"))
    cat("\n-------------\n")

    ## set up the model
    Model <- eval(substitute(AChE.scaled ~
CpkexpB2wDS(Dose.scaled,A,BMD,sex,

fixed=list(PB=c(F=xxxx,
                                                          M=yyyy),
                                                S=c(F=zzzz, M=zzzz),
                                                D=c(F=wwww,

M=wwww))),
                           list(xxxx=Bests[1],yyyy=Bests[2],
                                zzzz=log(xx$SDgrid$S[i]),
                                wwww=log(xx$SDgrid$D[i]))))
    ## estimate it
      fitpk <- if (!is.null(RandomParms)) {
        try(eval(substitute(nlme(Model,data=Pseldata,
                                 fixed=list(A ~ s.M.t - 1, BMD ~ sex - 1),
                                 random=xxxx,
                                 start=zzzz,
                                 weights=Weights,
                                 method="ML"),
                            list(Model=Model,xxxx=RandomParms,
                                 zzzz=start,
                                 Weights=Weights))))
      } else {
        try(eval(substitute(gnls(Model, data=Pseldata,
                                 params=list(A ~ s.M.t - 1, BMD ~ sex - 1),
                                 start=zzzz,
                                 weights=Weights),
                            list(Model=Model,
                                 zzzz=start,
                                 Weights=Weights))))
      }
```

```
    if (!inherits(fitpk, "try-error")) {
      xx$SDgrid$LL[i] <- logLik(fitpk)
      cat(paste("\nLL:",xx$SDgrid$LL[i],"\n"))
    }
    xx$Fit[[i]] <- fitpk
    save(xx,file=file.path("FineFits",paste(chem,"rda",sep=".")))
  }
  cat(paste(chem,"finished",date(),"\n\n"))

}
### makeFine2Grid.R --
### Uses the information in the files in ./Fits to create a new set of
templates
### in Fine2Fits.  Each new template already contains SDgrid and Fits
###
### The best expanded fit for the following chemicals is essentially the
basic model,
### so they are excluded from further action.  The criteria were:
### 1) Pvalue for the difference in log likelihoods was greater than 0.05
AND
### 2) BOTH BMDs were no more than 10% different between the expanded and
basic models
###    with the expanded model being the point of comparison (in the
denominator).

DropChems <-
c("ACEPHATE","CHLORPYRIPHOSMETHYL","DICROTOPHOS","DIMETHOATE","ETHOPROP",

"FENTHION","METHAMIDOPHOS","METHIDATHION","NALED","OXYDEMETONMETHYL",
                "PIRIMIPHOSMETHYL","PROFENOFOS")

K2IJ <- function(k, Npoints) {
  c(Sindx = (i <- ((k-1)%%Npoints) + 1),
    Dindx = (k - i)/Npoints + 1)
}

IJ2K <- function(i, j, Npoints) {
  i + Npoints * (j - 1)
}

indxmin <- function(i, N) {
  if (i == 1) i else i - 1
}
indxmax <- function(i, N) {
  if (i == N) i else i + 1
}
Neighbors <- function(k, Npoints) {
  ij <- K2IJ(k, Npoints)
  i <- ij[1]
  j <- ij[2]
  iseq <- indxmin(i,Npoints):indxmax(i,Npoints)
  jseq <- indxmin(j, Npoints):indxmax(j, Npoints)

as.vector(apply(data.matrix(expand.grid(i=iseq,j=jseq)),1,function(x)IJ2K(x[
1],x[2],Npoints)))
}


require(RBMDS)
require(nlme)


attach("../Data/opdata.rda")
savedir <- "Skel2"
```

```
finegriddb <- read.csv("griddb.csv",row.names=1)

finegriddb[DropChems,"DoIt"] <- 0
finegriddb$Npoints[] <- 5

Dochem <- which(finegriddb$DoIt == 1)


for (chem in Chemicals) {
  fname <- paste(chem,"rda",sep=".")
  load(file.path("FineFits",fname))
  ##  'xx' is the new version
  if (chem %in% DropChems) {
    save(xx, file=file.path("Fine2Fits",fname))
    next
  } else {
    oldxx <- xx
    xx$Nsteps <- 5
    xx$Fit <- list()
    Slist <- sort(unique(oldxx$SDgrid$S))
    Dlist <- sort(unique(oldxx$SDgrid$D))

    ## Build the new SDgrid centered around the maximum LL on the old one
    indx <- which.max(oldxx$SDgrid$LL)
    SDindx <- K2IJ(indx,oldxx$Nsteps)
    Smin <- Slist[indxmin(SDindx[1],oldxx$Nsteps)]
    Smax <- Slist[indxmax(SDindx[1],oldxx$Nsteps)]
    Dmin <- Dlist[indxmin(SDindx[2],oldxx$Nsteps)]
    if (SDindx[2] < oldxx$Nsteps) {
      Dmax <- Dlist[indxmax(SDindx[2], oldxx$Nsteps)]
    } else {
      ## If the old D was on the upper border of the grid, expand it by one
step
      Delta <- (max(Dlist) - min(Dlist))/oldxx$Nsteps
      Dmax <- max(Dlist) + Delta
    }
    finegriddb[chem,"Smin"] <- Smin
    finegriddb[chem,"Smax"] <- Smax
    finegriddb[chem,"Dmin"] <- Dmin
    finegriddb[chem,"Dmax"] <- Dmax
    xx$SDgrid <- expand.grid(S=seq(Smin,Smax,length=xx$Nsteps),
                             D=seq(Dmin,Dmax,length=xx$Nsteps))
    xx$SDgrid$LL <- numeric(nrow(xx$SDgrid))
    xx$SDgrid$LL[] <- NA
    xx$Start <- if (inherits(oldxx$Fit[[indx]],"nlme")) {
      oldxx$Fit[[indx]]$coefficients$fixed
    } else {
      oldxx$Fit[[indx]]$coefficients
    }

    xx$Fit <- vector("list",nrow(xx$SDgrid))

    ## Finally, fill in the LLs and Fits we already know.
    ## What are all the indexes?
    nearby <- Neighbors(indx,oldxx$Nsteps)

    ## There must be a better way to do this, but I'm tired ... .
    for (i in seq(along=nearby)) {
      K <- nearby[i]
      a1 <- sapply(xx$SDgrid$S,
                   function(x)identical(all.equal(x,
oldxx$SDgrid$S[K]),TRUE))
      a2 <- sapply(xx$SDgrid$D,
                   function(x)identical(all.equal(x,
oldxx$SDgrid$D[K]),TRUE))
```

```
      j <- which(a1 & a2)
      if (length(j) == 1) {
        xx$SDgrid$LL[j] <- oldxx$SDgrid$LL[K]
        xx$Fit[[j]] <- oldxx$Fit[[K]]
      }
    }

    save(xx,file=file.path(savedir,fname))
  }
}

write.table(finegriddb,file="fine2griddb.csv",sep=",",col.names=NA)
```