



**ČESKÁ ZEMĚDĚLSKÁ UNIVERZITA V PRAZE**

FAKULTA AGROBIOLOGIE, POTRAVINOVÝCH  
A PŘÍRODNÍCH ZDROJŮ

**Statistical modelling of dose response curves**  
with a special focus on herbicide resistance  
Ing. Kateřina Hamouzová, Ph.D.

e-book

2024

ISBN 978-80-213-3371-0

© Ing. Kateřina Hamouzová, Ph.D.

**Lektoroval: prof. Husrev Mennan**

**Jazyková korektura: Publikace neprošla jazykovou korekturou.**

**Vydala Česká zemědělská univerzita ve svém nakladatelství**

**ISBN 978-80-213-3371-0**

# Contents

<b>Introduction .....</b>	<b>1</b>
<b>Approach and models .....</b>	<b>2</b>
Data .....	2
Graphs .....	4
Choice of mean function .....	4
Violation of model assumptions .....	5
Detection and remedy.....	6
Lack of fit test .....	6
Box-Cox transformation .....	7
Choose candidate models .....	8
Dose response model.....	9
Built-in-models in R.....	10
Advantages of non-linear models .....	10
Disadvantages of non-linear models.....	11
Model parameters .....	12
Comparison of parameters.....	14
Fit of models.....	15
Group I – exponential.....	16
Group II – sigmoid functions.....	17
Example of a non-linear model modification .....	19
Fitting nonlinear models.....	21
Starting values .....	22
Convergence .....	22
Goodness of fit.....	23
<b>Nonsimilar dose-response curves model fit and .....</b>	<b>25</b>
Differences in upper and lower limit .....	25
Hormesis.....	28
<b>Package drc .....</b>	<b>29</b>
Fitting a single dose response curve .....	30
Fitting a multiple dose response.....	31
Simultaneous fitting – model reduction.....	32
Simultaneous fitting – model reduction: an example.....	34
<b>Practical training .....</b>	<b>34</b>
About R.....	36
Download R project .....	37

R Studio download .....	37
R basics .....	38
Statistical modelling of non-linear curves in R or R-Studio .....	41
<b>References .....</b>	<b>73</b>
<b>Supplementary data.....</b>	<b>74</b>
Data for analysis - florasulam .....	74
Data for analysis - tribenuron .....	75

Acknowledgement: We acknowledge partial utilization for the data analysis section from A. Linn and J. Čepl.

## Introduction

Statistical modelling in agricultural sciences often involves analysing complex relationships between various factors such as soil characteristics, weather conditions, crop types, and agricultural practices. When analysing herbicide resistance data in R, researchers may employ various statistical techniques and models to characterize monotone functions, assess symmetry or asymmetry in resistance patterns, and evaluate the efficacy of different management approaches. Functions for fitting monotone models, such as generalized additive models (GAMs) or nonlinear regression models, can be utilized to explore the relationship between herbicide exposure and resistance development. Additionally, statistical tests and visualization tools in R facilitate the comparison of resistance levels across herbicides and the identification of symmetric or asymmetric resistance patterns within weed populations.

Nonlinear models are commonly used in agricultural sciences to capture these intricate relationships. Here's a guide on how to perform statistical modelling using R with a focus on nonlinear models in agricultural sciences. Nonlinear modelling approaches are widely used in herbicide resistance studies to characterize dose-response relationships, understand the mechanisms underlying resistance, and optimize herbicide management strategies. Their flexibility, interpretability, and ability to capture complex dose-response relationships make them valuable tools for researchers and practitioners in the field of herbicide resistance management.

Herbicide resistance poses significant challenges in agriculture, and modelling approaches play a crucial role in understanding its dynamics, predicting its spread, and designing effective management strategies. Modelling provides insights into the underlying biological mechanisms driving herbicide resistance, such as target-site mutations, metabolic detoxification, or reduced herbicide uptake. By elucidating these mechanisms, models can help identify genetic and physiological factors contributing to resistance and guide research efforts to develop novel control methods. Models can simulate the dynamics of herbicide resistance evolution over time, considering factors such as selection pressure, mutation rates, gene flow, and population genetics. Predictive models allow researchers to anticipate the emergence and spread of resistant weed biotypes and assess the efficacy of different management strategies in mitigating resistance. Modelling enables the evaluation of various herbicide resistance management strategies, such as herbicide rotation, mixtures, and spatial diversification. By simulating different scenarios and management interventions, models help optimize decision-making to delay or prevent the onset of herbicide resistance while minimizing economic and environmental costs. Herbicide resistance management is a long-term endeavour, and models provide a framework for assessing the long-term impacts of management decisions on resistance dynamics, weed populations, crop yields, and agroecosystem sustainability. Long-term projections help stakeholders prioritize investments and adopt proactive management approaches. Models contribute to evidence-based policymaking by quantifying the economic, social, and environmental impacts of herbicide resistance and management interventions. Policy-relevant models inform regulatory decisions, such as herbicide registration, usage restrictions, and incentive programs aimed at promoting sustainable weed management practices. Models integrate diverse sources of data, including field experiments, genetic studies, agronomic observations, and expert knowledge. By synthesizing multidisciplinary information, models provide a holistic understanding of herbicide resistance dynamics and facilitate interdisciplinary collaboration among researchers, practitioners, and policymakers. Herbicide resistance is a dynamic and evolving problem influenced by factors such as climate change, agricultural practices, and technological innovations. Modelling frameworks can be adapted and updated to incorporate new data, emerging technologies, and evolving research findings, ensuring their relevance and applicability in a changing agricultural landscape.

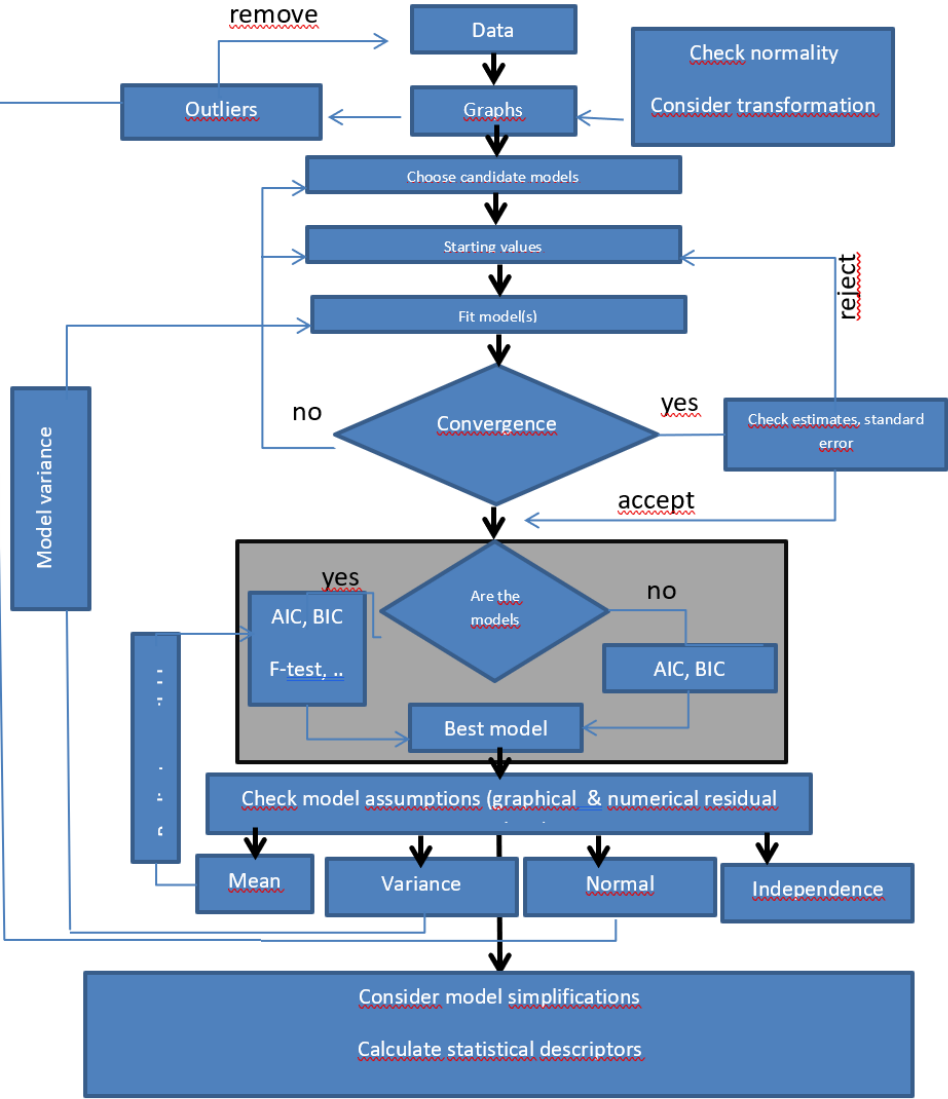
In summary, modelling approaches are indispensable tools for addressing the complex challenges posed by herbicide resistance in agriculture. By combining biological insights with mathematical and

computational techniques, models contribute to our understanding of resistance dynamics, inform decision-making, and support sustainable weed management practices.

## Approach and models

### Data

In agricultural sciences, as in many other fields, data can be classified into different types based on their characteristics. In the realm of herbicide resistance research within agricultural sciences, data often fall into distinct categories: binomial, nominal, and ordinal (gradual). Figure 1 shows a decision that has to be made when analysing data.



Archontoulis S.V., Miguez F.E., 2014

Fig. 1: Decision making for non-linear modelling of biological data

### Three types of data:

- **Binominal:**

Binomial data are characterized by two possible outcomes, often represented as success or failure, yes or no, 1 or 0. Examples in agricultural sciences could include the presence or absence of a

disease in plants, the survival or death of livestock after treatment, or the success or failure of a crop yield. Statistical techniques for analysing binomial data include logistic regression, binomial tests, and contingency tables.

- Examples of binomial data: dead/alive, abnormal/normal...
  - Size of response/effect depends on  $n$
  - No true replication
  - Fixed upper and lower limit

- **Nominal:**

Nominal data consist of categories without any inherent order or ranking. Examples in agricultural sciences could include different varieties of crops, types of soil, or breeds of livestock. Statistical techniques for analysing nominal data include chi-square tests, multinomial logistic regression, and correspondence analysis.

Examples of nominal data: number of seeds per plant...

- Only whole numbers possible
- True replications
- Fixed lower limit
- Sometimes recalculated to % inhibition or % of total response

- **Ordinal (Gradual) data:**

Ordinal data represent categories with a natural order or ranking but with unequal intervals between them. Examples in agricultural sciences might include subjective ratings such as crop quality ratings (e.g., poor, fair, good, excellent), severity rankings of pest infestations, or Likert scale responses measuring attitudes of farmers. Statistical techniques for analysing ordinal data include ordinal logistic regression (fig. 2), Wilcoxon rank-sum tests, and Kendall's tau correlation coefficient.

Examples of gradual data:

- biomass, growth rates, concentrations...

For gradual data statistical model is:

$$\underbrace{y_i}_{\text{response}} = \underbrace{f(\text{dose}_i, \alpha)}_{\text{mean}} + \underbrace{\varepsilon_i}_{\text{error}},$$

- where dose  $i$  is the dose value for observation  $i$
- the assumptions are:
  - 1) variance homogeneity  
(constant variance of the errors)
  - 2) correct mean (the right choice of function  $f$ )
  - 3) normally distributed errors
  - 4) independent measurements



Fig. 2: Dose-response of *Stellaria media* treated with increasing doses of florasulam. Picture was taken 30 days after the treatment (30 DAT)

You have to decide:

- Which is the best model to describe our data?
- How do you choose among competing models?
- Which is the best statistical index to judge the goodness of fit?

When conducting statistical modelling in R for agricultural sciences with these types of data, you would typically choose appropriate statistical techniques and models that are suitable for the specific data type. R provides a wide range of packages and functions for conducting analyses on all these types of data. For instance, `glm()` function in base R can be used for fitting logistic regression models for binomial data, while packages like MASS, nnet, or brms may provide more advanced functionalities for fitting nonlinear models to such data. Similarly, packages like car, ordinal, or rms offer tools for modelling ordinal data using regression techniques. Additionally, visualization tools in R, such as ggplot2, can be utilized to explore and communicate patterns in the data effectively.+

## Graphs

### Choice of mean function

Monotone functions:

In herbicide resistance studies, monotone functions may represent the relationship between the dosage of a herbicide and the probability of resistance development within a weed population.

Analysing monotone functions can help researchers identify the threshold levels of herbicide exposure that lead to resistance emergence and assess the effectiveness of different management strategies. Symmetric herbicide resistance refers to situations where weed populations exhibit similar levels of resistance across different herbicide types or modes of action. This symmetry suggests that the mechanisms conferring resistance are equally effective against multiple herbicides, potentially indicating widespread resistance development within the population. Understanding symmetric herbicide resistance patterns is essential for devising integrated weed management strategies that target multiple resistance mechanisms simultaneously. Asymmetric herbicide resistance occurs when weed populations demonstrate varying degrees of resistance to different herbicides or herbicide groups. In asymmetric resistance scenarios, certain herbicides may be more effective at controlling resistant weed biotypes than others. Identifying asymmetric resistance



patterns is crucial for optimizing herbicide selection and rotation strategies to manage resistant weed populations effectively while minimizing the risk of further resistance evolution (fig. 3).

- Symmetric (logistic)
- Asymmetric (Gompertz/Weibull)

Non-monotone function:

- Initial inhibition/stimulation (Brain-Cousens' model, hormesis)

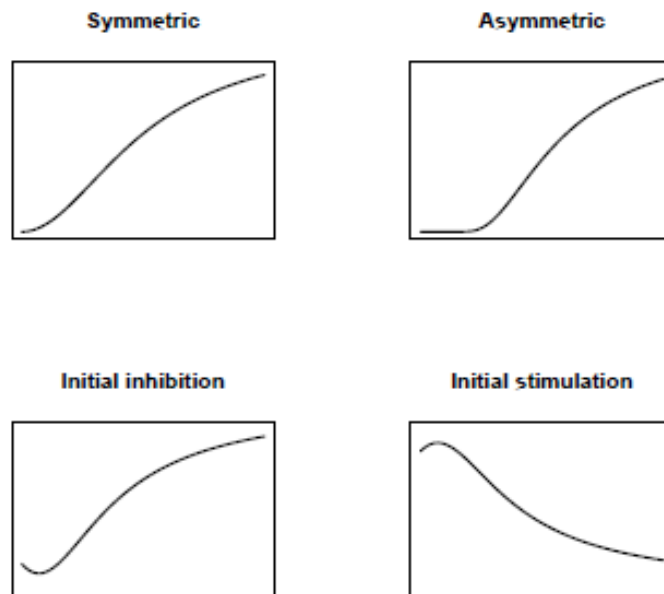


Fig. 3: Examples of mean function

### Violation of model assumptions

When analysing data, especially in the context of statistical modelling, it's important to consider potential violations of model assumptions, such as heteroscedasticity (variance heterogeneity), systematic deviations from the mean, and non-normal errors. These violations can impact the validity of statistical inferences and the reliability of model predictions.

#### a) Variance inhomogeneity

Heteroscedasticity refers to the situation where the variability of the response variable (dependent variable) differs across levels of the independent variables. In agricultural sciences, this might occur when the variability of crop yields or herbicide efficacy varies across different conditions or treatments. Heteroscedasticity violates the assumption of homogeneity of variance, which is typically assumed in many statistical models, including linear regression. Detection and correction of heteroscedasticity might involve transforming the response variable or using robust regression techniques that are less sensitive to variance heterogeneity.

- influence on standard errors (eg. may be too small)

#### b) Isolated or systematic deviation from the mean

Systematic deviations from the mean refer to patterns in the residuals (the differences between observed and predicted values) that are not random but exhibit a systematic trend. In agricultural research, this could manifest as consistent overestimation or underestimation of crop yields or other measured outcomes across different experimental conditions. Systematic deviations violate the assumption of independent and identically distributed (i.i.d.) residuals, which is often assumed in

statistical modelling. Identifying the source of systematic deviations and adjusting the model or experimental design accordingly is essential to ensure the validity of statistical inferences.

- imprecise estimates and/or large standard errors

c) Non-normal errors

Non-normal errors indicate that the distribution of residuals deviates from a normal (Gaussian) distribution. In agricultural studies, this might arise due to non-normality in the distribution of crop yields, pesticide concentrations, or other measured variables. Non-normal errors violate the assumption of normally distributed residuals, which is often assumed for conducting hypothesis tests and constructing confidence intervals. Techniques such as transformation of the response variable or using generalized linear models (GLMs) can be employed to accommodate non-normal errors and maintain the validity of statistical analyses.

- influence on standard errors

d) Independence is usually implied by design of experiment

e) DEPARTURES FROM THE ASSUMPTIONS NEED TO BE REMEDIED

## Detection and remedy

In R, various diagnostic tools and techniques are available to detect and address violations of model assumptions. These include residual plots, tests for normality (e.g., Shapiro-Wilk test), and methods for transforming variables. Additionally, robust statistical methods and alternative models (e.g., generalized linear models) can be employed to handle violations of traditional assumptions and ensure robust statistical inference in agricultural research.

- Deviations detected by
  - lack-of fit-test (comparing to a more general model)
  - plotting data together with fitted dose-response curves
  - inspecting the residual plot
- Non-normal errors and/or variance inhomogeneity
  - data transformation (only response, or both response and mean)
  - extension of the model (extrapolation methods)

## Lack of fit test

A lack of fit test is a statistical test used to assess whether a statistical model adequately fits the observed data. It helps to determine whether the model captures the underlying relationships between the independent and dependent variables effectively, or if there is a significant discrepancy between the model predictions and the observed data. In the context of agricultural sciences, a lack of fit test could be applied to various types of models, such as regression models predicting crop yields based on different factors like fertilizer application, soil nutrients, and environmental conditions.

In R, lack of fit tests can be conducted using appropriate functions depending on the type of model being evaluated. For linear regression models, the `anova()` function can be used to perform an F-test for lack of fit. For categorical data models, packages such as `car` or `MASS` offer functions to perform lack of fit chi-square tests. Additionally, graphical tools such as residual plots can help visually assess lack of fit in regression models.

- Comparing a dose-response model to a more general ANOVA model (fig. 4)
  - F-test
    - Non-significant: there is no evidence against the dose-response model
    - Significant: our model may not be appropriate

ANOVA table						
	Model	Df	RSS	F value	p value	
One-way ANOVA		17	5.1799			
DRC model		20	5.4002	3	0.2411	0.8665

Fig. 4: Comparing a dose-response model using anova function (R – output)

- The test is weak
- Supply the test by using a graphical assessment

**Box-Cox transformation**

The Box-Cox transformation is a statistical technique used to stabilize the variance and make data more normally distributed (fig. 5). It's particularly useful when dealing with data that violates the assumption of homoscedasticity (constant variance) or normality, which are common issues in regression analysis and other statistical modelling techniques.

- to remedy non-normal errors and/or variance heterogeneity
- both-side transformation (transforming response and mean)

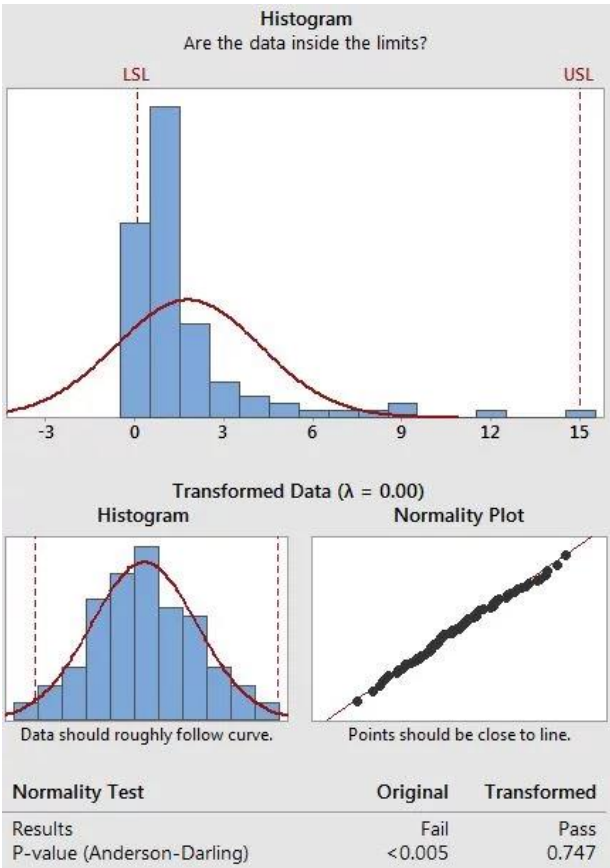


Fig. 5: Transforming a data for a proper analysis

- both the response and the nonlinear model function are shifted by a constant  $c$  and then transformed by a suitable function
- exponent lambda  $\lambda = -5$  to  $5$

- $\lambda \in \mathbb{R}$ ,  $\lambda=1$  – no transformation,  $\lambda=0$  – logarithm transformation
  - may alleviate variance heterogeneity and some skewness and thus recover a normal distribution
  - optimal  $\lambda$  is estimated from a more general linear model and not from the specified dose-response model
  -

In R, the Box-Cox transformation can be implemented using the `boxcox()` function available in various packages. This function automatically selects the optimal value of  $\lambda$  and applies the transformation to the data. Alternatively, you can manually specify different values of  $\lambda$  and apply the transformation using the formula described above. Additionally, graphical methods, such as the Box-Cox plot, can be used to visually assess the effectiveness of the transformation in stabilizing variance and normalizing the data distribution.

- steps to be followed
  - selecting lambda
  - transforming the data
  - model fitting
  - back-transformation

### Choose candidate models

Choosing candidate models for dose-response relationships is crucial in agricultural sciences, particularly in areas such as herbicide efficacy, fertilizer response, and pesticide toxicity. Here's a systematic approach to selecting candidate models:

- Understanding the Biological Mechanism: Begin by understanding the underlying biological mechanisms governing the dose-response relationship. For example, if you're studying the effect of a herbicide on weed growth inhibition, consider whether a linear or nonlinear relationship is more biologically plausible based on how the herbicide interacts with the weed's physiological processes.
- Exploratory Data Analysis (EDA): Conduct exploratory data analysis to visualize the dose-response relationship. Plot the response variable (e.g., plant biomass, weed mortality) against the dose levels (e.g., herbicide concentration, fertilizer rate) to identify potential patterns. Scatter plots, dose-response curves, and smoothers (e.g., LOESS curves) can aid in this process.
- Consider Candidate Models:
  - Linear Model: Start with a simple linear model if there's no prior knowledge suggesting a nonlinear relationship. A linear model assumes a constant slope across all dose levels
  - Nonlinear Models: Consider various nonlinear models that may better capture complex dose-response patterns. Common choices include:
    - Logistic Model: Suitable for sigmoidal dose-response curves, where the response increases rapidly at low doses and levels off at high doses.
    - Exponential Model: Appropriate for situations where the response increases or decreases exponentially with dose.
    - And others...
- Model Comparison: Compare the candidate models using statistical criteria such as goodness-of-fit measures (e.g., R-squared, AIC, BIC) and diagnostic plots (e.g., residual plots). Assess the models' ability to capture the observed dose-response pattern while avoiding overfitting.
- Cross-Validation: If possible, perform cross-validation to evaluate the predictive performance of the candidate models on independent datasets. This helps assess how well the models generalize to new data and can aid in model selection.

- f) Biological Interpretation: Finally, consider the biological relevance of the selected model. A model that not only fits the data well but also aligns with existing biological knowledge and hypotheses is often preferred.

In R, you can fit various candidate models for dose-response relationships using functions from packages such as `drc` (dose-response analysis), `nlme` (nonlinear mixed-effects models), `nls` (nonlinear least squares), or `glm` (generalized linear models). Additionally, tools for model selection and validation, such as `AIC()` and `BIC()` functions, are available to aid in the decision-making process.

## Dose response model

$$y=f(x, \Theta)+\varepsilon$$

$y$ - response/dependent variable

$f$ - function or model

$x$ - inputs/independent variable/dose

$\Theta$  - parameters to be estimated

$\varepsilon$  – error

- Types of responses (tab. 1)
  - dose (metameter)
    - any pre-specified amount of biological, chemical, radiation stress eliciting a certain, well-define response
    - non-negative value
    - often (not always) assumed to be measured without error
  - response
    - quantification of a biologically relevant effect
    - subject of random variation
    - **continuous** response: biomass, enzyme activity... (normal distribution)
    - **binary** (aggregated binomial) response: dead/alive, present/absent (binomial distribution)
    - **discrete**: number of juveniles, offsprings, roots...

Tab. 1: Models and model functions

Model type	Model function (f)	Function in drc
Generalized log-logistic	$c + \frac{d-c}{(1 + \exp(b(\log(x) - \log(e))))^a}$	llogistic()
Brain-Couzens	$c + \frac{d-c \cdot k}{1 + \exp(b(\log(x) - \log(e)))}$	
Cedergreen-Ritz-Streibig	$c + \frac{d-c + \exp(-1/(x^a))}{1 + \exp(b(\log(x) - \log(e)))}$	cedergreen()
<i>(0 &lt; a &lt; 1 is usually fixed in advance)</i>		
Log-logistic fractional polynomial	$c + \frac{d-c}{1 + \exp(b(\log(x+1))^{p1} + e(\log(x+1))^{p2})}$	fplogistic()
Log-normal	$c + (d - c)\Phi(b(\log(x) - \log(e)))$	lnormal()
<i>(Φ: distribution function for a normal distribution)</i>		
Weibull I	$c + (d - c)\exp(-\exp(b(\log(x) - \log(e))))$	weibull1()
Weibull II	$c + (d - c)(1 - \exp(-\exp(b(\log(x) - \log(e))))$	weibull2()
Gamma	$c + (d - c)\tilde{\Gamma}(bx, e, 1)$	gammadr()
<i>(Γ: distribution function for a Γ distribution)</i>		
Multistage	$c + (d - c)\exp(-b_1 - b_2x - b_3x^2)$	multi2()
NEC	$c + (d - c)\exp(b(x - e))$ for $x > e$ and $d$ otherwise	NEC.4()

doi:10.1371/journal.pone.0146021.t001

## Built-in-models in R

Why non-linear model?

Each parameter can be evaluated for whether it is linear or not

- if the second derivative of the function with respect to a parameter is not equal to zero, then the parameter is nonlinear
- given function can have a mix of linear and nonlinear parameters

## Advantages of non-linear models

Non-linear models offer several advantages when modelling dose-response relationships in herbicide resistance studies:

- 1) Flexibility: Non-linear models can capture complex relationships between dose levels and response variables, allowing for more flexible modelling of dose-response curves. In herbicide resistance research, where responses may not necessarily follow linear patterns, non-linear models can better accommodate various biological mechanisms underlying resistance development.
- 2) Better Fit: Non-linear models often provide better fit to the data compared to linear models when the relationship between dose and response is non-linear. By allowing for curved or sigmoidal dose-response curves, non-linear models can more accurately describe the true underlying relationship, leading to improved predictive accuracy and more reliable inference.
- 3) Interpretability: Non-linear models can offer more interpretable parameter estimates compared to linear models, especially when modelling complex biological processes. Parameters in non-linear models often correspond to biologically meaningful parameters such as EC50 (half-maximal effective concentration) or Hill slope, making it easier to interpret the estimated effects of dose levels on the response variable.
- 4) Extrapolation: Non-linear models can be more suitable for extrapolating dose-response relationships beyond the observed dose range. This is particularly valuable in herbicide

resistance research, where predicting the response to novel herbicide doses or combinations is important for informing weed management strategies.

- 5) Accommodation of Plateau and Saturation Effects: Non-linear models can effectively capture plateau and saturation effects commonly observed in dose-response relationships. These models allow for asymptotic behaviour where the response variable reaches a maximum or minimum value as dose levels increase, which may be critical for understanding herbicide efficacy and resistance mechanisms.
- 6) Robustness: Non-linear models can be robust to violations of assumptions such as homoscedasticity and normality of errors, which are often challenging to meet in dose-response studies due to inherent variability in biological systems.

Overall, the advantages of non-linear models in dose-response herbicide resistance studies lie in their ability to capture the complex, non-linear relationships between herbicide dose and response variables, leading to more accurate predictions and improved understanding of resistance mechanisms.

- Parsimony
  - models should have as few parameters as possible
  - linear models should be preferred to non-linear models
  - experiments relying on few assumptions should be preferred to those relying on many
  - models should be pared down until they are minimal adequate
  - simple explanations should be preferred to complex explanations
- Interpretability
- Prediction – tend to be more robust than competing polynomials, especially outside the range of observed data (i.e. extrapolation)

### Disadvantages of non-linear models

While non-linear models offer various advantages in modelling dose-response relationships in herbicide resistance studies, they also come with some disadvantages:

- 1) Model Complexity: Non-linear models can be more complex than linear models, often involving more parameters and requiring more computational resources for estimation. This complexity may make the model more challenging to interpret and may increase the risk of overfitting, especially with limited data.
- 2) Parameter Estimation Challenges: Estimating parameters in non-linear models can be computationally intensive and may require specialized optimization techniques. Convergence issues or local minima can arise, particularly with complex model structures, leading to difficulties in obtaining reliable parameter estimates.
- 3) Model Selection Difficulty: Selecting an appropriate non-linear model structure can be challenging, as there are numerous candidate models with different functional forms. Identifying the most suitable model for a specific dose-response relationship may require extensive model comparison and evaluation, which can be time-consuming and subjective.
- 4) Extrapolation Uncertainty: While non-linear models may provide good fit within the observed dose range, extrapolating beyond this range can be uncertain and may lead to unreliable predictions. Extrapolation in non-linear models is particularly risky when the underlying biological mechanisms are poorly understood or when extrapolating to novel conditions.
- 5) Interpretability Issues: Non-linear models may lack the straightforward interpretability of linear models, especially for models with complex functional forms. Parameter estimates may not always have direct biological meaning, making it challenging to interpret the estimated effects of dose levels on the response variable.

- 6) Assumption Sensitivity: Non-linear models may still rely on assumptions such as independence of observations, homogeneity of variance, and normality of errors, which may not always be met in practice. Violations of these assumptions can lead to biased parameter estimates and unreliable inference, similar to linear models.
- 7) Data Requirement: Non-linear models may require larger sample sizes compared to linear models to achieve stable parameter estimates and reliable inference, especially when fitting complex model structures. Obtaining sufficient data points across the dose range of interest can be challenging in herbicide resistance studies, particularly for rare or newly emerging resistance mechanisms.

In summary, while non-linear models offer significant advantages in capturing complex dose-response relationships in herbicide resistance studies, they also present challenges related to model complexity, parameter estimation, model selection, extrapolation, interpretability, assumption sensitivity, and data requirement. Careful consideration of these disadvantages is essential when choosing and applying non-linear models in herbicide resistance research.

In a brief, the disadvantages are as following:

- less flexible than linear models
- generally there is no analytical solution for estimating the parameters
- choice of model is crucial
  - try a large library of functions and choose the model with the lowest error
  - choose a model based on whether it has been used successfully in similar applications
- numerical method needs to be used to find estimates for the parameters
  - convergence of the algorithm needs to be checked
    - a lack of convergence often results from the consideration that these numerical methods require starting values
- in some cases also linear or generalized linear models (*glm*) can be successfully used

### Model parameters

The log-logistic model is commonly used to describe dose-response relationships, particularly in toxicology and pharmacology, but it can also be applied in herbicide resistance studies. In the log-logistic model, the response variable is typically transformed using logarithms to achieve linearity in the parameters. The general form of the log-logistic model for a dose-response relationship can be expressed as:

$$Y = \frac{c}{1 + (d/X)^b}$$

where:

Y is the response variable (e.g., herbicide efficacy, resistance level).

X is the dose or concentration of the herbicide.

c is the upper asymptote of the dose-response curve, representing the maximum response achievable.

d is the dose at which the response is halfway between the minimum and maximum asymptotes (EC50 or ED50).

b is the slope parameter, which determines the steepness of the dose-response curve.

The parameters of the log-logistic model have specific interpretations in the context of dose-response relationships:



Upper Asymptote (*c*): This parameter represents the maximum achievable response level. In the context of herbicide resistance, it may indicate the maximum possible reduction in weed control or mortality achievable with the herbicide under study. EC50 (or ED50) (*d*): The EC50 (effective concentration 50%) or ED50 (effective dose 50%) is the dose at which the response is halfway between the minimum and maximum asymptotes. It serves as a measure of potency or effectiveness of the herbicide in inhibiting or controlling the target organism. In herbicide resistance studies, EC50 may indicate the concentration of herbicide required to achieve a certain level of inhibition or mortality in resistant weed populations. Slope (*b*): The slope parameter determines the steepness of the dose-response curve. A higher value of *bb* indicates a steeper curve, suggesting a more rapid change in the response with changes in dose. In herbicide resistance studies, the slope parameter may provide insights into the sensitivity of resistant weed populations to varying doses of herbicides.

Estimating these parameters involves fitting the log-logistic model (fig. 6) to experimental or observational data using statistical software. Once estimated, the parameters can be used to describe the shape of the dose-response curve, make predictions about response levels at different doses, and compare the efficacy of different herbicides or herbicide treatments.

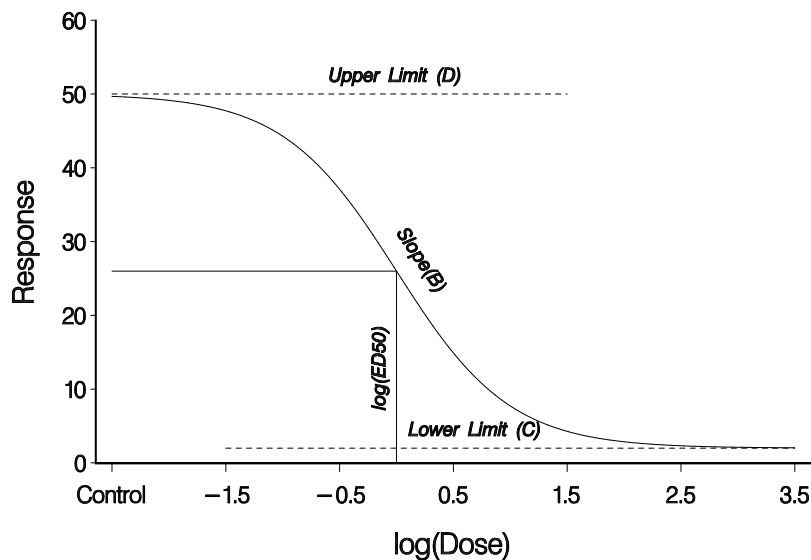


Fig. 6: Example of log-logistic model parameters

$$f(x, (b, c, d, e)) = c + \frac{d - c}{1 + \exp \{b(\log(x) - \log(e))\}}$$

Four parameters:

*b* - relative slope around *e* – decreasing function of the dose, corresponding to positive *b*, increasing with dose corresponds to negative *b*

*c* - lower limit

*d* - upper limit

*e* – also *ED*<sub>50</sub> dose producing a response half-way between the upper limit *d* and lower limit *c*

Function is symmetric around *e*

## Comparison of parameters

When comparing parameters in nonlinear models for herbicide resistance studies, it's essential to consider the specific model being used and the context of the study. Here are some key considerations for comparing parameters in different nonlinear models:

- 1) model Structure: Different nonlinear models may have distinct functional forms and parameterizations. For example, the log-logistic model and the logistic model are commonly used in dose-response studies, but they have different parameter interpretations and may fit the data differently.
- 2) interpretability of Parameters: Parameters in nonlinear models should have clear biological interpretations. For example, in dose-response models, parameters such as EC50 (effective concentration 50%) or LD50 (lethal dose 50%) represent biologically meaningful points on the dose-response curve and can be compared across models to assess differences in potency or efficacy.
- 3) goodness of Fit: When comparing parameters across models, it's important to consider the goodness of fit of each model to the observed data. Models with better fit to the data may provide more reliable parameter estimates and predictions.
- 4) statistical Inference: Statistical tests and confidence intervals can be used to compare parameter estimates across models and assess whether differences are statistically significant. Hypothesis tests can be conducted to determine whether parameters are significantly different between models.
- 5) practical Implications: Ultimately, the choice of model and interpretation of parameters should be guided by the practical implications for herbicide resistance management. For example, differences in EC50 values between herbicides or resistance mechanisms may indicate variations in potency or effectiveness, which could inform herbicide selection or resistance management strategies.
- 6) robustness: Consider the robustness of parameter estimates to variations in data and model assumptions. Models that produce stable parameter estimates across different datasets or model specifications may be more reliable for practical applications.
- 7) model Complexity: Take into account the complexity of each model and the trade-off between model complexity and interpretability. More complex models may provide better fit to the data but may also be more difficult to interpret and prone to overfitting.

Overall, comparing parameters in nonlinear models for herbicide resistance studies involves evaluating model fit (fig. 7), interpreting parameter estimates in a biological context, conducting statistical tests for differences, and considering the practical implications for herbicide resistance management. It's essential to carefully consider these factors when selecting and interpreting nonlinear models for dose-response analysis in herbicide resistance research.

- Common approach
  - re-parameterize and re-fit the model for each function of interest – difficult
- Single-model approach was implemented
  - single model is fit and the delta method (van der Vaart, 1998) is used to calculate approximate standard errors for function of the parameters

Way to compare parameters

- t-test
  - based on a single fitted dose-response model
  - based on particular components in the model fit
- confidence interval
  - parameter estimate lie in the confidence interval of another parameter
- F-test

- based on two fitted models, one being more general than the other
- utilise the entire model fit

```

A 'logistic' model was fit.

Parameter estimates:

      Estimate Std. Error   t-value   p-value
b:1  0.5195260  0.0763599   6.8036530 1.346e-09
b:2  0.8009531  0.2257107   3.5485829  0.0006
b:3  0.6818983  0.1285630   5.3040019 8.851e-07
b:4  1.8447607  0.1663488  11.0897146 1.698e-18
b:5  1.6507990  0.1758393   9.3881105 8.863e-15
c:1 -0.0165855  0.1078225  -0.1538223  0.8781
c:2  0.1326173  0.0471817   2.8107768  0.0061
c:3  0.1463932  0.0604363   2.4222722  0.0175
c:4  0.0795360  0.0394604   2.0155892  0.0470
c:5 -0.0090699  0.0443533  -0.2044913  0.8385
d:1  1.8795509  0.0423711  44.3592666 7.412e-61
d:2  0.9459940  0.0422671  22.3813367 1.084e-37
d:3  1.0903222  0.0405608  26.8811926 1.268e-43
d:4  2.1535763  0.0281853  76.4078098 2.017e-80
d:5  1.8062724  0.0292458  61.7617568 1.052e-72
e:1  1.7949368  0.4782218   3.7533561  0.0003
e:2  0.9455148  0.2494531   3.7903514  0.0003
e:3  1.3731181  0.4527616   3.0327617  0.0032

```

Fig. 7: Output of the function “summary” with estimated parameters

#### Comparison of parameters

- Comparison of different herbicides
- Effective dose (ED) and selectivity index (SI)/resistance factor (R)
  - both are functions of the parameters
  - ED: dose that yields a response which is (100-y)% of the maximal response  $d$  (a reduction of  $y$ %)
 
$$ED_y = e / (100 - y)^{1/b}$$
  - SI: ratio between  $ED_x$  for one curve and  $ED_y$  for another curves
 
$$SI(x, y) = ED_x / ED_y$$

#### Fit of models

In agriculture, various nonlinear models are used to describe relationships between input factors (e.g., dose, time, environmental conditions) and agricultural outcomes (e.g., crop yield, pest population growth, herbicide efficacy). Here are some typical nonlinear models along with application examples:

- Group I – exponential

- Group II – sigmoid functions
- Group III – photosynthesis
- Group IV – temperature dependencies
- Group V – peak or bell-shaped curves
- Group VI – other nonlinear equations

These are just a few examples of nonlinear models commonly used in agriculture. Each model has its own set of assumptions, parameter interpretations, and applications, and the choice of model depends on the specific research question, available data, and biological context of the agricultural system being studied.

## Group I – exponential

In herbicide resistance studies, the exponential model is not typically used as a primary modeling approach for dose-response relationships. However, it can be relevant in certain contexts, particularly when describing the dynamics of resistance evolution within weed populations over time. The exponential model assumes constant proportional growth over time, which may not directly apply to dose-response relationships in herbicide resistance studies. However, it can be employed to describe the exponential increase in the frequency of herbicide-resistant individuals within weed populations under certain conditions. Here's how the exponential model could be applied in the context of herbicide resistance:

- 1) Frequency of Resistant Individuals: The exponential model could be used to describe the increase in the frequency of herbicide-resistant individuals within weed populations over successive generations of herbicide exposure. This could be relevant for studying the long-term dynamics of resistance evolution in response to herbicide selection pressure.
- 2) Population Dynamics: The exponential model may also be relevant for modeling population growth dynamics of herbicide-resistant weed biotypes within agricultural fields or landscapes. This could involve describing the exponential increase in the abundance of resistant weeds over time in response to repeated herbicide applications.
- 3) Modeling Resistance Spread: In some cases, the exponential model could be used to model the spatial spread of herbicide resistance within agricultural landscapes. This might involve describing the exponential increase in the area infested with resistant weed populations over time due to dispersal and expansion of resistant biotypes.

It's important to note that while the exponential model can provide insights into the rapid increase of herbicide resistance within weed populations, it may oversimplify the complex dynamics of resistance evolution. Other models, such as logistic models or more mechanistic models based on evolutionary principles, may be more suitable for capturing the nuanced interactions between herbicide selection pressure, genetic mechanisms of resistance, and ecological factors influencing resistance evolution. Overall, while the exponential model may not be the primary choice for dose-response modeling in herbicide resistance studies, it can still be relevant for understanding certain aspects of resistance dynamics, particularly in the context of population growth and spread of herbicide-resistant weed biotypes over time.

Exponential model:

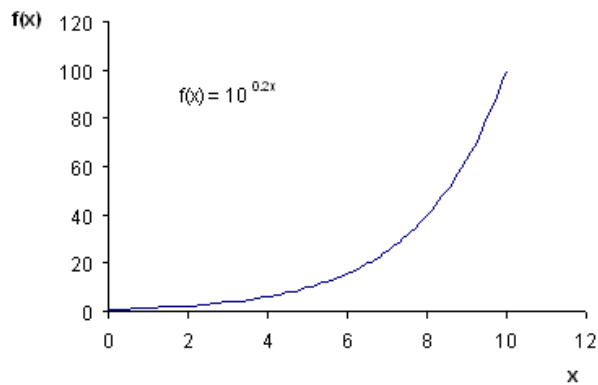


Fig. 8: Exponential model – an example

Equation of exponential model (fig. 8)

$$y = y_0 \exp(-kt)$$

$$y = y_0 [1 - \exp(-kt)]$$

where  $y$  is the response variable (e.g. soil organic matter)

$t$  is the explanatory variable (e.g. time)

$y_0$  is the initial or the maximum  $y$  value

$k$  is a rate constant that determines the steepness of the curve

- Commonly used to describe light and N vertical distribution within plant canopies
- N<sub>2</sub>O emission response to N fertilizer
- Cumulative soil respiration
- Photoperiodic sensitivity
- Temperature or moisture responses to nitrification
- Water infiltration rate

## Group II – sigmoid functions

In herbicide resistance studies, sigmoid functions are commonly used to model dose-response relationships (fig. 9), particularly in the context of assessing herbicide efficacy and characterizing the response of weed populations to varying herbicide concentrations. Sigmoid functions are preferred because they can capture the typical S-shaped or sigmoidal curves often observed in dose-response relationships, where the response increases gradually at low doses, reaches a maximum level of response, and then levels off at higher doses. These sigmoid functions can be fitted to experimental or observational data using nonlinear regression techniques, and the parameters of the models can be estimated to describe the shape of the dose-response curve and assess herbicide efficacy. By fitting sigmoid functions to dose-response data, researchers can quantify important parameters such as EC<sub>50</sub> (or ED<sub>50</sub>), slope, and maximum efficacy, providing valuable insights into herbicide performance and resistance levels in weed populations.

- Logistic
- Richards
- Gompertz
- Weibull

...

$$y = y_{\text{asym}} / \{1 + \exp[-k(t - t_m)]\}$$

$$y = y_{\text{asym}} / \{1 + v \exp[-k(t - t_m)]\}^{1/v}$$

$$y = y_{\text{asym}} \exp\{-\exp[-k(t - t_m)]\}$$

$$y = y_{\text{asym}} [1 - \exp(-at^b)]$$

where  $y$  is the response variable

$t$  is the explanatory variable

$y_{\text{asym}}$  is the asymptotic or the maximum  $y$  value

$t_m$  is the inflection point at which the growth rate is maximized

$k$  controls the steepness of the curve

$v$  deals with the asymmetric growth

$a$  and  $b$  are parameters that determine the shape of the curve

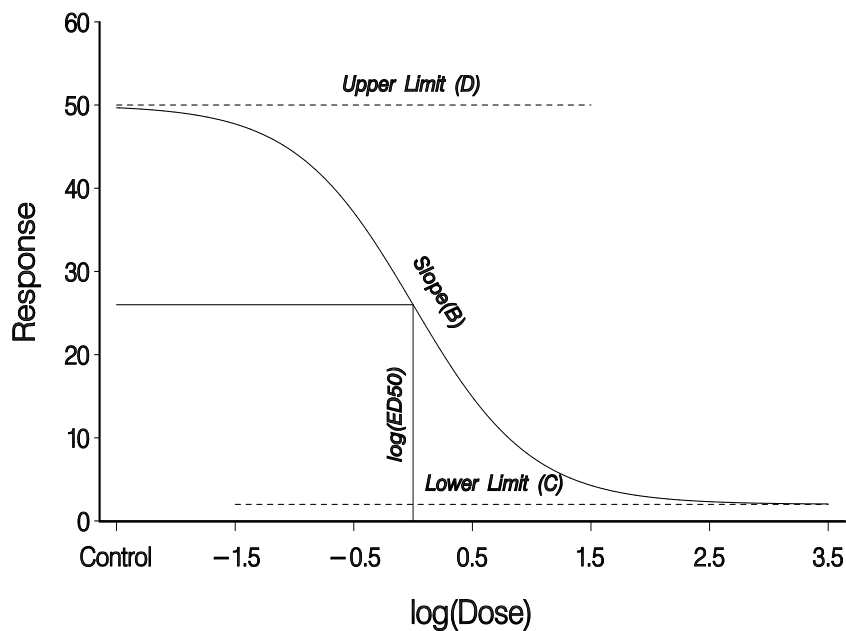


Fig. 9: Log-logistic model – an example

- mathematical functions having S shape
- described plant height, weight, leaf area index or seed germination as a function of time
- N application rate
- herbicide dose
- used as 0-1 modifiers in process-based models to incorporate moisture availability or soil pH

Important: all sigmoid equations assume an initial  $y$  value close to zero at time zero (planting, biomass weight...)

- *logistic equation*
  - describes symmetric growth
  - having inflection point at half the final size/maximum/upper limit
- *Gompertz equation*
  - inflection point that is controlled by its asymptotic value and is at about one-third
- *Richards, Weibull or beta*
  - have more flexibility in dealing with asymmetric growth
  - the inflection point can be at any  $x$  value

- having a flexible inflection point is another important feature of sigmoid curve (in cases if you did not reach 100% efficacy)
  - Birch (1999) modified the logistic equation to deal with asymmetric growth by adding an extra shape parameter
  - when growth is known to decrease after a certain period of time, then the beta function is a better option
  - in cases when the initial phase is very important, different versions of beta function should be used

### Example of a non-linear model modification

In herbicide resistance studies, modifying nonlinear models by changing parameters can be a useful approach to improve the model fit, address specific research questions, or account for additional factors influencing the dose-response relationship. Here are several ways parameters can be modified in nonlinear models:

- 1) **Parameter Constraints:** Adjusting parameter constraints can ensure that parameter estimates remain within biologically plausible ranges. For example, constraining the EC50 parameter in a dose-response model to be greater than zero may prevent unrealistic estimates.
- 2) **Parameter Interactions:** Introducing interactions between parameters can capture complex relationships between variables. For instance, introducing interactions between herbicide type and weed species in a dose-response model can account for differential responses among weed species to different herbicides.
- 3) **Parameter Linkages:** Linking parameters across multiple dose-response curves or experimental conditions can improve parameter estimation efficiency and reduce model complexity. This approach is particularly useful when fitting dose-response models to data from multiple experiments or treatments.
- 4) **Parameter Transformation:** Transforming model parameters using mathematical functions can improve model convergence or interpretability. For example, exponentiating or taking the logarithm of a parameter may ensure that it remains positive or interpretable on a different scale.
- 5) **Model Selection:** Selecting a subset of parameters or simplifying the model structure can reduce overfitting and improve model generalization. Conducting model selection procedures such as Akaike Information Criterion (AIC) or cross-validation can help identify the most parsimonious model while maintaining good fit to the data.
- 6) **Model Ensemble Approaches:** Combining multiple nonlinear models or model variants into an ensemble can improve prediction accuracy and robustness. Ensemble approaches, such as model averaging or stacking, integrate information from different model parameterizations to provide more reliable predictions.
- 7) **Bayesian Methods:** Using Bayesian estimation techniques allows for incorporating prior knowledge or beliefs about parameter values into the modeling process. Bayesian methods provide a framework for quantifying uncertainty in parameter estimates and making probabilistic predictions.
- 8) **Model Calibration:** Calibrating model parameters to observed data through optimization or Bayesian inference can improve model accuracy and reliability. Calibration involves adjusting model parameters iteratively until the model predictions closely match the observed data.

Overall, modifying parameters in nonlinear models offers flexibility in capturing complex dose-response relationships in herbicide resistance studies. By adjusting model parameters appropriately, researchers can develop more accurate and robust models that better reflect the underlying biological processes and improve the understanding of herbicide resistance mechanisms.

Example of a nonlinear model modification is given in a table 2 and figure 10. Starting with Eq 7.1 the parameters a, b, c and d were added step by step to Eq. 7.1 resulting in four new equations. Horizontal and vertical arrows in the figure panel indicate how the additional parameters affected the model.

Tab. 2: Examples of a non-linear model modification

#	Equation	Parameters
7.1	$y=(x/(1+x))$	-
7.2	$y=(bx/(1+bx))$	b=0.5
7.3	$y=a*(bx/(1+bx))$	b=0.5 a=0.8
7.4	$y=a*(b(x-c)/(1+b(x-c)))$	b=0.5 a=0.8 c=0.5
7.5	$y=(a*(b(x-c)/(1+b(x-c))))+d$	b=0.5 a=0.8 c=0.5 d=0.5

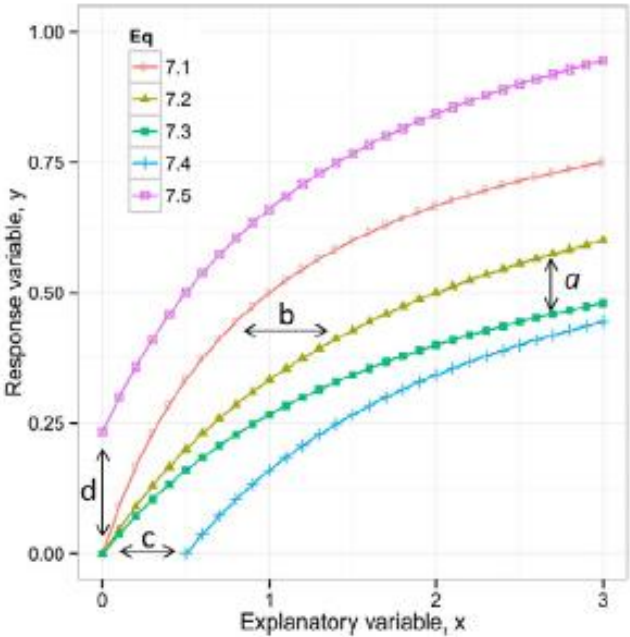


Fig. 10: Example of a nonlinear model modification



## Fitting nonlinear models

Several statistical software packages offer tools for nonlinear modeling in herbicide resistance studies. These software packages provide a range of tools and capabilities for fitting nonlinear models to dose-response data in herbicide resistance studies. The choice of software depends on factors such as familiarity with the software, specific modeling requirements, and availability of desired features for data analysis and visualization.

- Many statistical software packages available:
  - SAS - SAS (Statistical Analysis System) is a widely used commercial statistical software suite that includes procedures for nonlinear modeling. The `PROC NLIN` procedure in SAS can be used for fitting nonlinear models to dose-response data.
  - R - R is a free and open-source statistical computing and graphics software that offers numerous packages for nonlinear modeling. Packages like `nlme`, `nls`, `minpack.lm`, and `drc` are commonly used for fitting nonlinear models to dose-response data in herbicide resistance studies.
  - JMP - JMP is a graphical statistical software package that offers interactive tools for data exploration and modeling. It provides functions for fitting nonlinear models to data, including dose-response curves in herbicide resistance studies.
  - GEnStat
  - MatLab - MATLAB provides a comprehensive environment for nonlinear modeling and optimization. The Curve Fitting Toolbox in MATLAB offers functions for fitting nonlinear models to data, including custom models for dose-response relationships in herbicide resistance studies.
  - Origin - Origin is a data analysis and graphing software package that offers extensive capabilities for fitting nonlinear models. It includes built-in functions for fitting custom models to data and visualizing the results.
  - SPSS - (Statistical Package for the Social Sciences) is another popular commercial statistical software package that offers capabilities for nonlinear modeling. Although SPSS may not have dedicated procedures for nonlinear modeling, custom scripts or plugins can be used to fit nonlinear models to data.
- Nonlinear parameter estimates obtained using different methods, most common:
  - ordinary least squares (LSQ):
    - minimizes the sum of squared error between observation and prediction
    - $x_1, \dots, x_n$  – dose
    - $y_1, \dots, y_n$  – observed response
    - $w_i$  – user-specified weights
      - often left unspecified, equal 1
      - argument *weights*, eg. expressed as standard deviations of response
      - transform-both-sides approach should be preferred over using weights
  - maximum likelihood method (ML):
    - seeks the probability distribution that makes the observed data most likely
- Choice of estimation methods affect the parameter estimates
- Most problems encountered during the use of standard nonlinear regression software functions are due to:
  - poor choice of competing modes
  - incorrect equation
  - incorrect starting values

## Starting values

Choosing appropriate starting values for parameters in nonlinear modeling, particularly in herbicide resistance studies, is crucial for ensuring the convergence of optimization algorithms and obtaining reliable parameter estimates. Here are some strategies for selecting starting values:

all the procedures for NL parameter estimation require initial value

- all the procedures for NL parameter estimation require initial value
- choice of values influence the convergence of the estimation algorithm
  - Convergence = sequence of essentially random or unpredictable events can sometimes be expected to settle into a pattern
  - worst case yielding in no convergence

there is no standard procedure for getting initial estimates

- Choosing starting values - 5 practical methods:
  - 1) parameters with biological meaning
    - similar experiments or in a data-driven way using the dose-response data themselves to elicit relevant information
  - 2) use graphical exploration
  - 3) transform the NL model into a linear one
    - eg. logarithmic transformation in which rough estimates can be obtained by linear regression
  - 4) in case of no clear guidance use grid search
    - generate an extensive coverage of possible parameter values and then evaluate the model at each one of these parameter combinations
  - 5) use pre-specified algorithms

These algorithms are specific to a given equation and given data set:

- choice of starting values crucially affects whether convergence is achieved
- availability of good starting values facilitate parameter estimation in nonlinear models
- obtained by using parameter estimates previously reported for self starter functions return data-driven starting values for the model parameters
- R self starter functions available for the function *nls*
- supplying starting values manually, use argument *start*
- function *getInitial ()* – obtain starting values that were used for obtaining a particular model fit

## Convergence

Convergence issues in nonlinear regression, particularly in the context of herbicide resistance studies, can arise when optimization algorithms fail to find a satisfactory solution within a reasonable number of iterations. Here are some common strategies for addressing convergence problems in nonlinear regression:

- Checking algorithm convergence
  - convergence is achieved when a measure is below a certain threshold value (measure = such as the relative offset or maximum change among parameter estimates)
  - if convergence is not achieved
    - poor choice of starting values
    - the selected model is not well suited to describe the data

- if convergence is achieved
- next step – evaluate whether the parameter estimates are within a reasonable range
- Evaluation whether the parameter estimates are within a reasonable range
    - requires evaluating the point estimates and also their standard errors
    - usually large standard errors are a sign of convergence problems (even if convergence was apparently achieved in previous step)
    - if no problems were encountered – continue by assessing model assumptions and simplifying model
  - Evaluating model assumptions
    - One model
      - normally distributed errors
      - independent errors
      - homogeneous variance for the errors
      - can be both - linear and non-linear models
      - Substantial deviations from the assumptions result in inaccurate estimates, distorted standard errors or both
      - Violation of these assumptions can be detected from an analysis of the residuals by means of graphical procedures and formal statistical tests
  - Evaluating model assumptions
    - One model
      - whether the distribution of measurement errors follows normality – use standardized residual plot
      - heterogeneity of variance can be detected by looking at the plot of the fitted values over residuals
        - residual errors show a trend, this can be addressed by modelling the variance as a function of the independent variable or the fitted values
        - if variance heterogeneity is ignored, the parameter estimates might not be influenced much, but confidence and prediction intervals are
        - residuals are assumed to be independent – this assumption is violated – it is visually evident in a plot of correlations of residuals against lag
        - typically, variables measure with time on the same subject tend to result in autocorrelated residuals that need to be accounted for by modelling the variance – covariance matrix
  - Multiple models – model selection criteria
    - How to find the best model among competing models?
      - depends on the structure of the models
      - different statistical criteria can be used:
        - F test, Akaike information criterion (AIC), Bayesian information criterion (BIC), likelihood ratio test
      - when the models are nested (=one model is a special case of another), any of these criteria can be used
      - when the models are non-nested (=models having different structure), use AIC and BIC

### Goodness of fit

Assessing the goodness of fit for nonlinear models in herbicide resistance studies is crucial for evaluating the adequacy of the model in capturing the underlying dose-response relationship. Here are several approaches:

- no single method or index to best assess the goodness of fit
- many different methods – graphical and numerical
- graphical comparison – quick visual assessment
- numerical statistical indices:  $R^2$ , adjusted  $R^2$ , bias, mean squared error, RMSE, modelling efficiency (ME), concordance correlation ...
- some indices measure the absolute error (includes units), some the relative error (excludes units)
- depending on the data – combination of the indices is possible
- some simple and very common indices like  $r^2$  and bias do not account for the number of parameters
- $R^2$  does not represent a good metric of model performance for NL modes
- does not account for the number of parameters
- full model does not necessarily include the simpler model with one single parameter as is the case with linear models
- numerical statistical descriptors
- indicate the average performance of the model across the sample
- when the variability is not constant throughout the sample, then statistical indices do not capture the fact that the uncertainty is not the same at different magnitudes of the response variables
- predictive ability of the model
- cross-validation techniques can be used
- mean squared error of prediction is more appropriate

## I. Nonsimilar dose-response curves model fit and hormesis

Dose-response curves are almost never perfectly similar due to various factors, including biological variability, environmental conditions, genetic diversity, and experimental variability. Given these sources of variability, it's important to recognize that dose-response curves are rarely identical and may exhibit differences even under controlled experimental conditions. Understanding the factors contributing to variability in dose-response curves is essential for interpreting experimental results and designing effective herbicide resistance management strategies.

Reasons why dose-response may exhibit differences:

- differences in formulation
- differences in uptake and translocation
- environmental factors
- differences at the molecular level affecting the affinity of the target site
- Models should always be tested statistically for similarity (lack of fit F-test, likelihood-ratio test)

### Differences in upper and lower limit

Ignoring differences in upper and lower limits in nonlinear models, particularly in the context of dose-response relationships in herbicide resistance studies, can lead to incorrect conclusions and misinterpretation of results. To address these issues, it's essential to select appropriate model structures that explicitly account for differences in upper and lower limits in dose-response relationships. For example, using sigmoidal models like the logistic or log-logistic models allows for flexible parameterization of upper and lower limits, ensuring that the model accurately captures the entire dose-response curve. Additionally, thorough model validation procedures, such as cross-validation or independent dataset validation, can help assess the predictive performance of the model and identify potential limitations or biases. Overall, by considering differences in upper and lower limits in nonlinear modelling, researchers can obtain more accurate and reliable insights into herbicide efficacy and resistance mechanisms. Models with different upper and lower limits are shown in a picture below (fig. 11):

a) similar upper and lower limits

$$(b = 1.5, D = 1, C = 0)$$

b) similar slopes but different lower limits

$$(b = 1.5, D = 1, C = 0 \text{ and } 0.3)$$

c) different slopes and similar upper and lower limits ( $b = 1$  and  $2, D = 1, C = 0$ ),

d) different slopes and different upper and lower limits

( $b = 1$  and  $2$ ,  $D = 0.8$  and  $1$ ,  $C = 0$  and  $0.3$ )

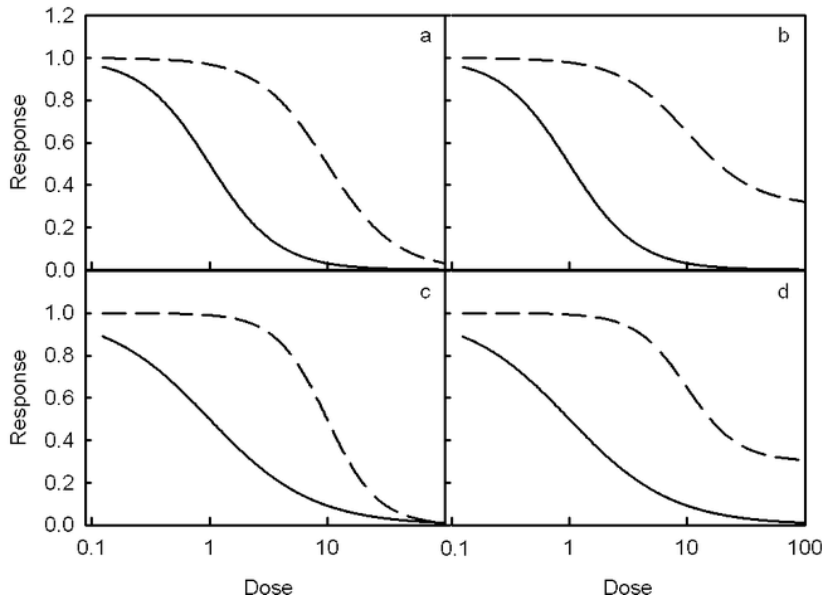


Fig. 11: Different upper and lower limits and similar and/or different slopes

Relative potencies as a function of response between the curves  $a$  and  $c$ ,  $b$  and  $d$

If  $B$  is reference herbicide,  $A$  is an unknown one, dose  $x_A$  of herbicide  $A$  gives the same response as dose  $x_B$  of herbicide  $B$ , then the relative potency  $r_A$  of herbicide  $A$  is

$$r_A = x_B / x_A \quad (\text{fig. 12}).$$

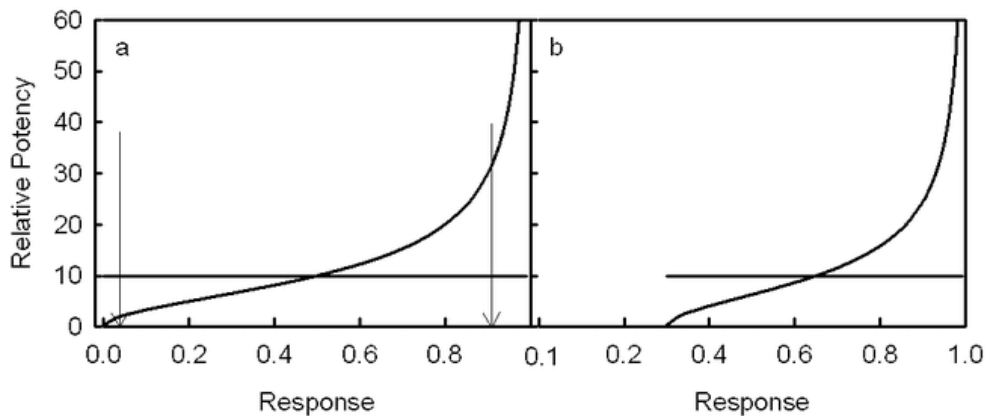


Fig. 12: Relative potencies as a function of response between the curves  $a$  and  $c$ ,  $b$  and  $d$

- transform data to calculate relative potencies with different upper and lower limits
- estimation of  $D$  and  $C$  for two herbicides to the same scale

$$w_A = \frac{y_A - C_A}{D_A - C_A}, \quad w_B = \frac{y_B - C_B}{D_B - C_B}$$

- artificial conclusions – transformed responses directly comparable at all response levels, while original responses only at the intersection
- Procedure (data output of the transformation is shown in the fig. 13)
  - fit the curves simultaneously
  - apply Box-Cox transformation if necessary
  - check the quality of each set of dose-response models
    - analysis of variance by lack-of-fit F test
  - test for similarity of parameters
    - compare the fit of multiple dose-response models with and without similar parameters by F-test based on residual sum of squares of the two models

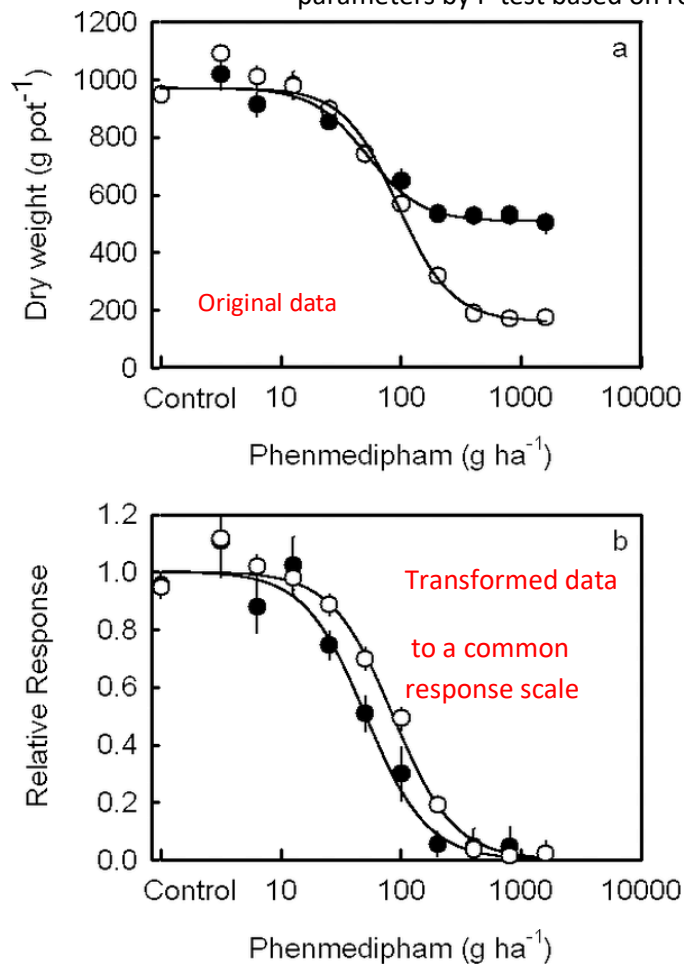


Fig. 13: Transformation of the data to a common scale

- slight differences in slope of the dose-response curves cause large deviations at agronomically relevant control levels
- it is necessary to show the whole story

- actual response is recommended
- display the entire range of relative potency from low to high levels
- scaling removes some of the biological information – can lead to wrong conclusions

## Hormesis

Hormesis is a phenomenon in which exposure to a low dose of a stressor (such as a herbicide) results in a beneficial response or stimulation of growth or other physiological processes, while higher doses of the stressor have detrimental effects. In the context of herbicide resistance studies, hormesis can manifest as increased plant growth or tolerance to herbicides at low doses, followed by inhibition or toxicity at higher doses. Modelling hormesis in herbicide resistance studies using nonlinear models involves capturing non-monotonic dose-response relationship. By incorporating hormesis into nonlinear modelling approaches for herbicide resistance studies, researchers can better understand the complex dynamics of herbicide-plant interactions and their implications for weed management and herbicide efficacy (fig. 14).

- Stimulatory effect of low dosage of a toxic compound

### F glyphosate (Roundup Speed)

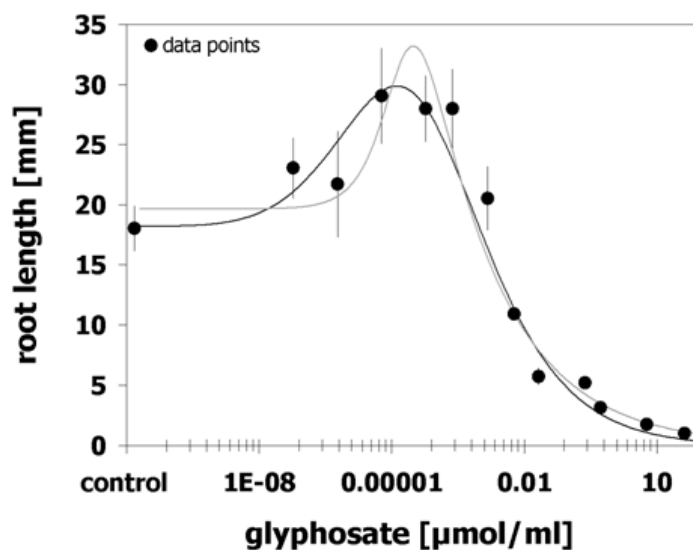


Fig. 14: Root growth of *Lactuca sativa* after the glyphosate treatment, black curve – Cedergreen et al., grey curve – Brain- Cousens model, error bars – standard deviation (Belz et al. 2012)

- Brain-Cousens model
  - a specific type of nonlinear model used in herbicide resistance studies, particularly for describing hormetic dose-response relationships
  - developed by Brain and Cousens in 1989
  - based on the assumption that the effects of a herbicide on plant growth can vary nonlinearly with dose
  - exhibiting both stimulatory and inhibitory effects across different concentration ranges
  - problems when fitting data displaying an early increase in responses at low doses, a broad hormetic dose range and/or gently sloping curves



- Cedergreen model
  - problems in case of pre-hormetic toxicity
  - developed by Line Cedergreen in 2008
  - based on the assumption that the effect of a herbicide on plant growth or response varies nonlinearly with dose
  - exhibiting both stimulatory and inhibitory effects across different concentration ranges
- Both models have limitations in case of extremely steep sloping data sets

“Which model best fits observed responses must be statistically and graphically reassessed for every set of empirical data. An uncritical application of a particular model can cause serious misinterpretation .... (Belz et al., 2012).“

## Package drc

The "drc" package in R is a tool for dose-response modeling, which is commonly used in various fields including toxicology, pharmacology, and agriculture, including herbicide resistance studies. The "drc" package provides functions for fitting a wide range of dose-response models to experimental data and conducting model selection and evaluation. It provides a comprehensive set of tools for fitting, comparing, and evaluating dose-response models, allowing for robust analysis and interpretation of experimental data.

The package drc:

- bioassays – experiments with biologically active compounds
- dose response studies for different plant species and/or different herbicides
- potencies of compounds compared at some *a priori* response levels – 50% reduction of biomass or other response
- add-on package for the language and environment R
- open source
- freely available <http://www.R-project.org>
- consists entirely of interpreted R lines
- current version at <http://www.bioassay.dk>
- number of other R packages related to dose-response analysis: *DoseFinding*, *drfit*, *grofit*, *MCPMod*, *nlstools*
- originally developed to provide nonlinear model fitting for specialized analysis routinely carried out in weed science
- main function **multdrc** – carries out the estimation of parameters and returns a model fit
- default function – four-parameter logistic model with built-in self starter functions

Drc package

- Once a model fit is obtained, following methods for **extracting information** are available:
  - **anova**: lack-of fit test or test for reduction between two models
  - **coef**: parameter estimates
  - **fitted**: fitted values
  - **logLik**: log likelihood value
  - **plot**: plot of the fitted curves
  - **residuals**: raw residuals
  - **summary**: summary of the model fit

- **Vcov**: estimated variance-covariance matrix
- **compParm, ED and SI**: comparisons of parameters, ED values and SI values

Limitation of using drc package

- assumed that responses are independent
- certain correlation structures may readily be implemented within the framework of *drc*, eg. correlation that is a function of the dose-response model function
- for continuous responses the package *meddrc* extends many of the capabilities of *drc* to correlated dose-response data fitted through nonlinear mixed-effects regression models
- *meddrc* utilizes the capabilities of the package *nlme*, but allows estimation of conditional and population-based ED levels

### Fitting a single dose response curve

- main function *multdrc/drm* – can be used to fit data from one or more dose response curves
- default – four parameter logistic model is fitted to the data  
`>model <-multdrc(NAME)`
- argument to function *multdrc* is a data frame which is a collection of columns of the same length
- call *multdrc* produces no output – all relevant information is stored in the object model
- to see the outputs, use extractors

Extractors

1) *anova (model)* – can be used to obtain a lack-of fit test, comparing the four-parameter logistic model to a one-way ANOVA model (fig. 15)

### ANOVA table

	Model	Df	RSS	Df	F value	p value
One-way ANOVA		17	5.1799			
DRC model		20	5.4002	3	0.2411	0.8665

Fig. 15: Lack-of-fit test using *anova (model)*

Not significant = four-parameter logistic model provides as good a fit as the one-way ANOVA

- *t*-statistics and corresponding *p*-values are for testing the null hypotheses that the parameters are equal to 0 (not necessarily relevant hypotheses to consider)
- Estimate of common variance parameter  $\sigma^2$  is 0.27
  - the variance of any residual; in particular, the variance  $\sigma^2 (y - Y)$  of the difference between any variate *y* and its regression function *Y*
  - the higher the value, the worse the fit

Hierarchical model structure

- provides a convenient way to specify special cases obtained by fixing one or more parameters at certain given value
- these parameters will not be estimated from the data, will be kept fixed at the specified value – quite frequent case in practice (*fixed*)
  - examples:
    - fixing *f*=1: `LL.4(fixed=c(NA, NA, NA, 1))`,
    - fixing *c*=0, *d*=1: `LL.4(fixed=c(NA, 0, 1, NA))`

- NA=parameters have to be estimated from the data, the order of model parameters is b, c, d, e, f

### Fitting a multiple dose response

- We assume:

- 1) each curve follows a four-parameter logistic curve
- 2) parameters differ among curves

```
>model2<-multdrc(response to dose~independent variable, grouping variable, data=NAME)
>anova(model2)
```

We have to check an assumption that regression provides an acceptable description of the data (fig. 16).

#### ANOVA table

	ModelDf	RSS	Df	F value	p value
Two-way ANOVA	70	0.38635			
DRC model	85	0.45955	15	0.8841	0.5842

Fig. 16: Not significant nonlinear regression provides an acceptable description of the data

```
>summary(model2) – output is shown in fig. 17
```

c limit in sample 1 and 5 – negative, not meaningful but not significantly different from 0 – use the three parameters model

```
A 'logistic' model was fit.
```

```
Parameter estimates:
```

	Estimate	Std. Error	t-value	p-value
b:1	0.5195260	0.0763599	6.8036530	1.346e-09
b:2	0.8009531	0.2257107	3.5485829	0.0006
b:3	0.6818983	0.1285630	5.3040019	8.851e-07
b:4	1.8447607	0.1663488	11.0897146	1.698e-18
b:5	1.6507990	0.1758393	9.3881105	8.863e-15
c:1	-0.0165855	0.1078225	-0.1538223	0.8781
c:2	0.1326173	0.0471817	2.8107768	0.0061
c:3	0.1463932	0.0604363	2.4222722	0.0175
c:4	0.0795360	0.0394604	2.0155892	0.0470
c:5	-0.0090699	0.0443533	-0.2044913	0.8385
d:1	1.8795509	0.0423711	44.3592666	7.412e-61
d:2	0.9459940	0.0422671	22.3813367	1.084e-37
d:3	1.0903222	0.0405608	26.8811926	1.268e-43
d:4	2.1535763	0.0281853	76.4078098	2.017e-80
d:5	1.8062724	0.0292458	61.7617568	1.052e-72
e:1	1.7949368	0.4782218	3.7533561	0.0003
e:2	0.9455148	0.2494531	3.7903514	0.0003
e:3	1.3731181	0.4527616	3.0327617	0.0032

Fig. 17: Fitting of the logistic model with function “summary”

>ED50(model2, c(10, 50,90)) – output shown in fig. 18

	Estimate	Std. Error
1:10	0.02613925	1.266207e-02
1:50	1.79493682	4.782218e-01
1:90	123.25517064	1.016492e+02
2:10	0.06085337	4.795887e-02
2:50	0.94551481	2.494531e-01
2:90	14.69102314	1.240795e+01
3:10	0.05474148	2.946568e-02
3:50	1.37311812	4.527616e-01
3:90	34.44286207	2.810508e+01
4:10	0.05996769	6.824301e-03
4:50	0.19732814	1.018932e-02
4:90	0.64932290	8.071697e-02
5:10	0.05569450	8.054704e-03
5:50	0.21079544	1.382424e-02
5:90	0.79782946	1.332322e-01

Fig. 18: Estimated of ED10, ED50 a ED90

Output provides standard errors and *p-values* for testing the null hypothesis that the indices are equal to 1.

### Simultaneous fitting – model reduction

In herbicide resistance studies, simultaneous fitting and model reduction techniques are often employed to efficiently analyze dose-response data and identify the most parsimonious model that adequately describes the underlying biological processes.

- Reduce a model using significance tests
  - Data: TM (Cedergreen, 2004)
  - Variables: dose, ptc = curve number, rgr = response (growth rates of duckweed and the treatments are mixtures of two herbicides with different modes of action), 180 observations
  - Simultaneous model using four-parameter Gompertz model with different parameters for different assays – fig. 19 and 20

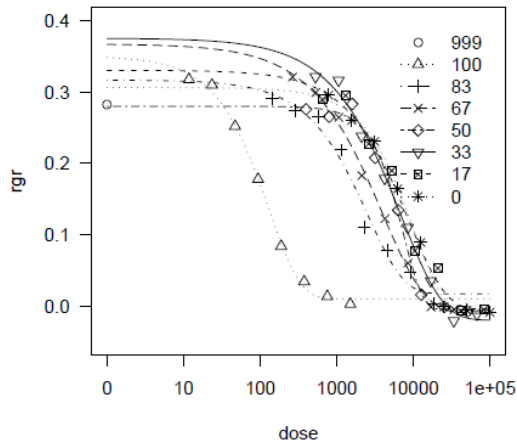


Fig. 19: Simultaneous fitting of several responses (Cedergreen, 2004)

	Estimate	Std. Error	t-value	p-value
b:100	1.0655191	0.2170291	4.9095672	2.344e-06
b:83	0.9288037	0.2498459	3.7175067	0.0003
b:67	0.7840124	0.1500974	5.2233574	5.737e-07
b:50	1.7389206	0.2908936	5.9778581	1.566e-08
b:33	0.8050234	0.1412846	5.6978859	6.163e-08
b:17	0.9136431	0.1806474	5.0576035	1.215e-06
b:0	1.1056319	0.2084593	5.3038276	3.963e-07
c:100	0.0109434	0.0099491	1.0999415	0.2731
c:83	0.0179145	0.0167275	1.0709619	0.2859
c:67	-0.0089369	0.0141037	-0.6336573	0.5273
c:50	-0.0040568	0.0094046	-0.4313600	0.6668
c:33	-0.0189114	0.0122196	-1.5476233	0.1238
c:17	-0.0039410	0.0118115	-0.3336561	0.7391
c:0	-0.0099106	0.0095277	-1.0401952	0.2999
d:999	0.2814899	0.0068124	41.3199574	1.670e-84
d:100	0.3497876	0.0245492	14.2484165	4.906e-30
d:83	0.3166707	0.0237328	13.3431502	1.265e-27
d:67	0.3668345	0.0310219	11.8250075	1.495e-23
d:50	0.2798449	0.0103144	27.1314743	3.096e-60
d:33	0.3749350	0.0316003	11.8649171	1.168e-23
d:17	0.3301507	0.0254194	12.9881540	1.129e-26
d:0	0.3059784	0.0211905	14.4394434	1.530e-30
e:100	4.8721705	0.1126764	43.2403908	2.980e-87
e:83	7.8423011	0.1611212	48.6733107	1.697e-94
e:67	8.2910662	0.1547290	53.5844382	1.858e-100
e:50	8.9175008	0.0811260	109.9216722	2.602e-146
e:33	8.8503914	0.1470690	60.1785064	9.934e-108
e:17	9.0866222	0.1369517	66.3491119	6.726e-114
e:0	9.2302599	0.1146251	80.5256594	3.003e-126

Estimate of residual variance: 0.0005569125

Fig. 20: Simultaneous model using four-parameter Gompertz model with different parameters for different assays

- By default – parameter  $d$  is estimated for the control group,  $d:999$
- Individual curves does not produce a fit that agree with the control group – single parameter  $d$  to all curves
- Use „collapse“ argument – may be specified using a *data frame* or a list as argument
- Difference between data frame and collapse as an argument:
  - Overlapping in their functionality
  - Data frame:

- Better for collapsing parameters for arbitrary curves
- No requirement for the corresponding grouping variable to be defined
- List specification
  - More general structures involving more than one variable per parameter
  - Follows the same syntax as is used for *lm* and *glm*

### Simultaneous fitting – model reduction: an example

- General model, based on the assumption and dataset plot: `modele3.1 <- multdrc(rgr ~ dose, pct, data = TM, fct = g4())`
- Model with common upper limit: `modele3.2 <- multdrc(TM[,c(3,1)], TM[,2], collapse=data.frame(TM[,2], TM[,2], colFct(TM[,2],1:8), TM[,2]), fct=g4())`
- Model with common lower limit: `modele3.3 <- multdrc(rgr ~ dose, pct, data.frame(pct, 1, pct, pct), data = TM, fct = g4())`
  - Comparing two models:
    - `anova(modelex3.1, modelex3.2)`  $p=2.869e-06$  (NS-model with a common upper limit rejected)
    - `anova(modelex3.3, modelex3.1)`  $p=0.5006$  (S- modelex3.1 is as good as modelex3.3 -> fit a model with the common lower limit equal to 0)
- Model with the lower limit equal to 0: `modele3.4 <- multdrc(rgr ~ dose, pct, data.frame(pct, pct, pct), data = TM, fct = g3())`
  - `anova(modelex3.4, modelex3.3)`,  $p=0.256$  (S – we can reduce the initial model to a model with common lower limit equal to 0)

### Practical training

- Experimental design:
  - 2 populations of *Stellaria media*
    - S – sensitive standard (fig. 21)
    - Putatively R – survival of application of sulfonylureas in field
  - Small pot experiment (fig. 22)
    - 9 doses of florasulam + untreated control
    - 4 replications per dose
  - Expected output – calculation of resistance factor (fig. 23)



Fig. 21: Examples of dose-response of *Stellaria media* treated with florasulam. On the picture, there are two populations, one susceptible to that active ingredient, the other is resistant to florasulam.

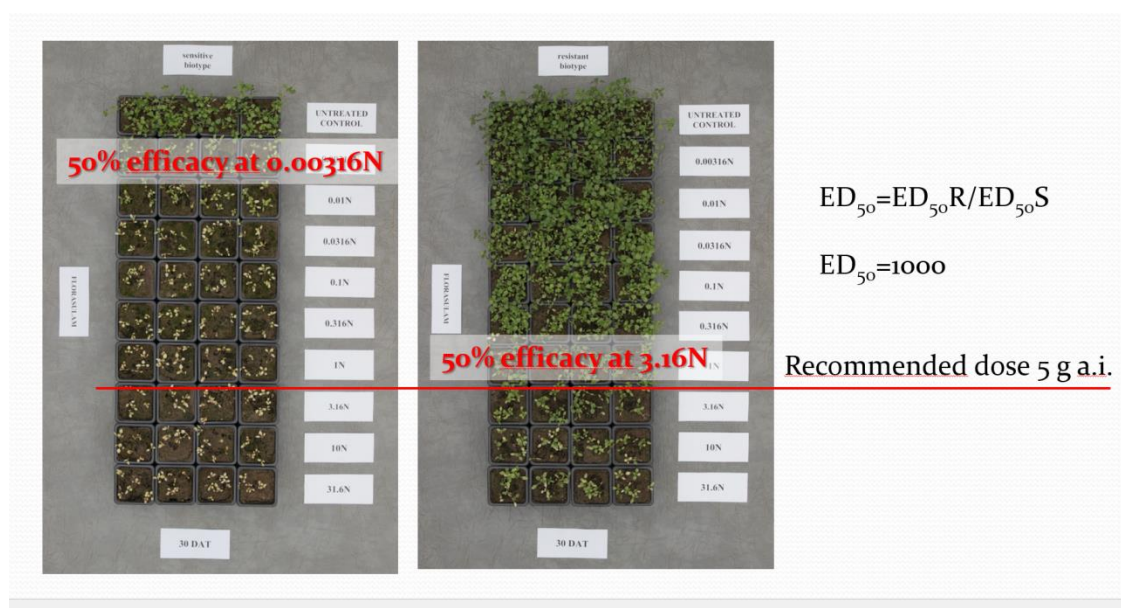


Fig. 22: *Stellaria media* resistant to ALS inhibitors – recently presented in Rothamsted, Resistance '19

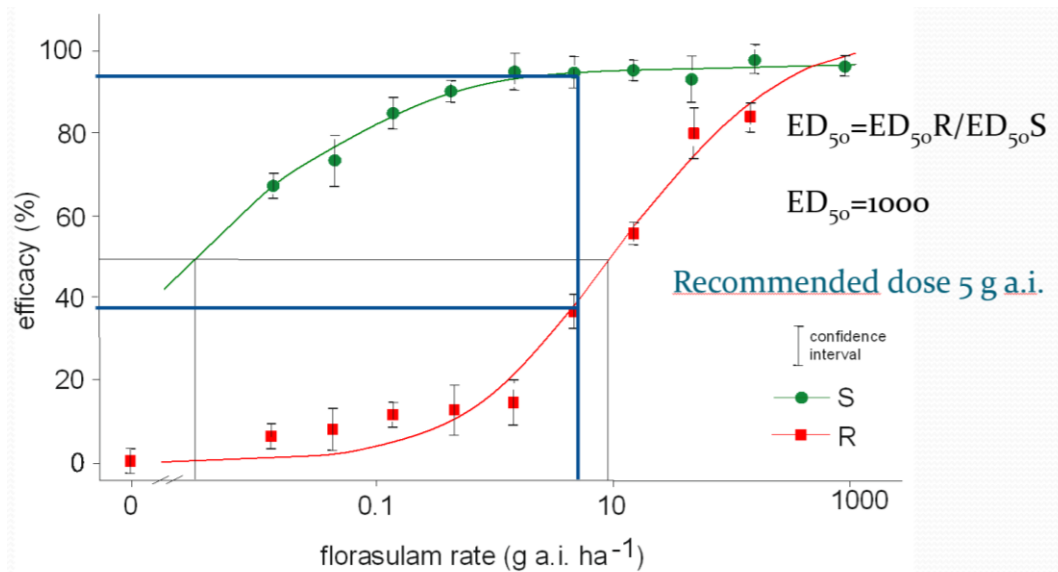


Fig. 23: Statistical modelling and calculations of resistance factor

## About R

R is a powerful programming language and environment specifically designed for statistical computing and graphics. It offers a wide variety of statistical and graphical techniques, making it a popular choice among statisticians, data analysts, and researchers. Here are some basics of statistical analysis in R:

- 1) **Installing R and RStudio:** R can be downloaded from the Comprehensive R Archive Network (CRAN) website (<https://cran.r-project.org/>). RStudio is a popular integrated development environment (IDE) for R, providing a more user-friendly interface for coding in R.
- 2) **Loading Data:** Before conducting any statistical analysis, you'll need to load your data into R. Commonly used functions for loading data include `read.csv()`, `read.table()`, `read_excel()` from the `readxl` package, and `readr::read_csv()` from the `readr` package.
- 3) **Exploratory Data Analysis (EDA):** EDA is the process of exploring your data to understand its characteristics. Functions such as `summary()`, `str()`, `head()`, and `tail()` are useful for getting an overview of the data structure, summary statistics, and first few or last few observations.
- 4) **Data Manipulation:** R offers powerful tools for data manipulation. The `dplyr` package provides functions like `filter()`, `select()`, `mutate()`, `summarize()`, and `arrange()` for data manipulation tasks such as subsetting, filtering, creating new variables, and summarizing data.
- 5) **Statistical Tests and Models:** R has a vast array of packages for conducting statistical tests and fitting models. Some commonly used packages include `stats`, `MASS`, `lme4`, `nlme`, `survival`, `caret`, `glmnet`, and many others. Depending on the analysis you need to perform, you can find specific functions for hypothesis testing, regression analysis, time series analysis, survival analysis, machine learning, and more.
- 6) **Visualization:** Visualization is a crucial aspect of data analysis. R provides various packages for creating plots and graphs, including `ggplot2`, `lattice`, and `base` graphics. These packages allow you to create static and interactive visualizations for exploring data patterns and communicating results effectively.
- 7) **Reporting:** RMarkdown is a powerful tool for integrating R code, output, and narrative text into a single document. It allows you to create reports, presentations,



and even websites directly from R. By combining R code with Markdown syntax, you can generate dynamic and reproducible documents.

- 8) Packages and Libraries: R's functionality can be extended through packages, which are collections of R functions, data, and documentation. You can install packages using the `install.packages()` function and load them into your R session using the `library()` function.

## Download R project

<https://cran.r-project.org/bin/windows/base/>

Download R (latest version)

Follow installation instructions

Once the R is installed: go to Packages – Set CRAN mirror, choose DENMARK

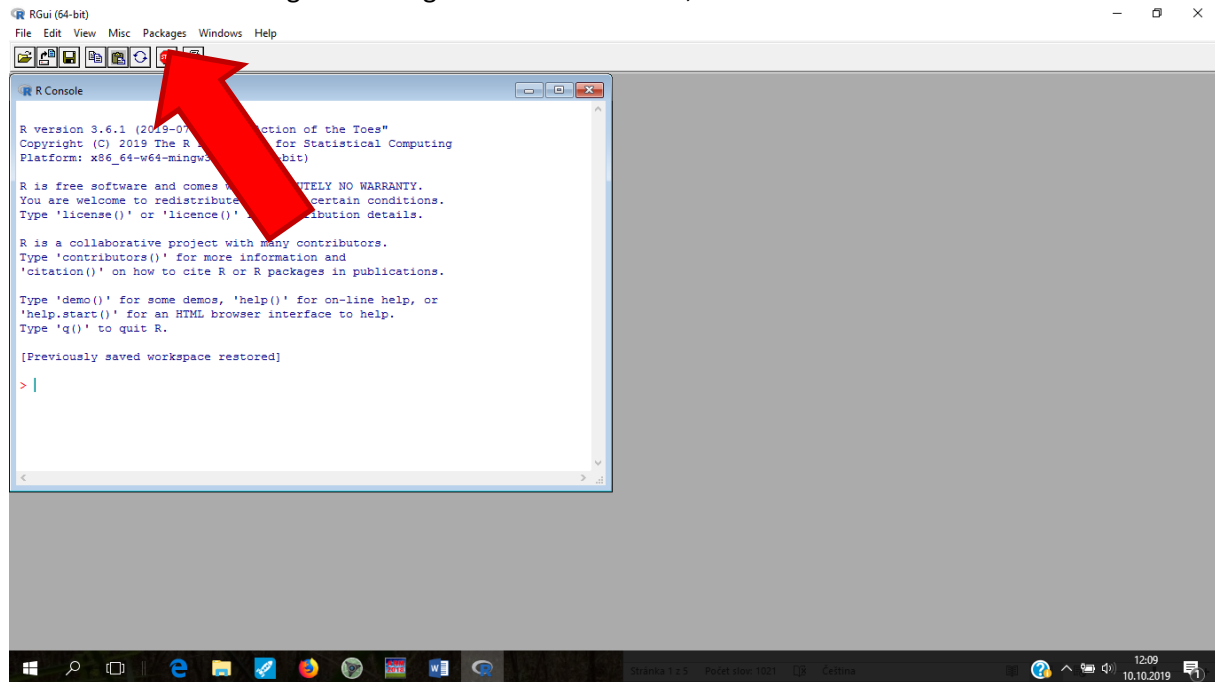


Fig. 24: Installing packages for statistical analysis

Again go to Packages – Install packages choose „drc“ (fig. 24)

During the installation process you might be asked: Would you like to use a personal library instead? Confirm Yes

Would you like to create a personal library C:// to install packages into? Confirm Yes

Caution: Extraction process takes some time!

## R Studio download

Installation of the RStudio gives a user an access to a much-improved environment to work in your R scripts. It includes a console that supports direct code execution and tools for plotting and keeping track of your variables in the workspace, among other features (fig. 25).

<https://cran.rstudio.com/>

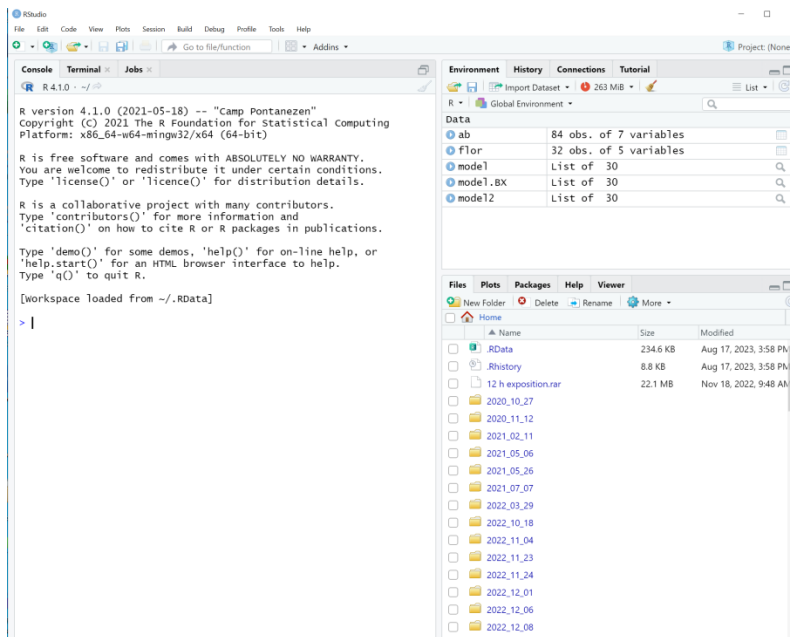


Fig. 25: Installation of R-Studio

## R basics

If you're a beginner interested in learning R, here's a step-by-step guide to get you started with basic functions:

1+2

2^2

sqrt(9)

4/(3\*4 - 2^3)

# logical operations

1 == 2

1 != 2

1 < 2

1 > 2

1 >= 2

4/(3\*4 - 2^3) == 1

# values storing - object creation

x <- 1

y <- c(1, 2, 3)

is(x)

is(y)

# vectors and their types

# logical

a <- c(TRUE, TRUE, FALSE, TRUE, FALSE)

is(a)

1:5 == 1:5

1:10 > 5

```

# integer
b <- 1:5
b2 <- c(1L, 2L, 3L)
is(b)

# numeric
c <- c(1, 2, 3, 4, 5)
c2 <- rep(10,10)
c3 <- seq(from = 0, to = 1, by = 0.1)
is(c)

# character
d <- c("a", "b", "2")
is(d)

# vectors are converted to most general type (logical -> integer -> numeric -> character)
abc <- c(TRUE)
abc
is(abc)

abc <- c(TRUE, 1L)
abc
is(abc)

abc <- c(TRUE, 1L, 1/3)
abc
is(abc)

abc <- c(TRUE, 1L, 1/3, "a")
abc
is(abc)

# other types of vectors - complex
abc <- c(1+1i, 2+1i)
abc
is(abc)

# special type of vector - factors, good format for categorical data
e <- factor(d)
e
is(e)

# matrices and arrays
mat1 <- matrix(1:9, nrow = 3, ncol = 3)
mat1

mat2 <- matrix(1:9, nrow = 3, ncol = 3, byrow = T)
mat2
is(mat1)
dim(mat1)

```

```

arr1 <- array(1:27, c(3, 3, 3))
arr1
is(arr1)
dim(arr1)

mat3 <- matrix(c(1:8, "i"), 3, 3)
mat3

# operations with vectors and matrices
vec1 <- 1:10

# operations are element-wise
vec1 + 1
vec1^2
vec1 * vec1
vec2 <- vec1 * vec1

# vectors are "recycled"
vec1 + 1:5*100

# cross product (vektorov soutin)
vec1 %*% vec1

# matrix algebra
matrix(1, nrow = 3, ncol = 2) * matrix(1, nrow = 2, ncol = 3)
matrix(1, nrow = 3, ncol = 2) %*% matrix(1, nrow = 2, ncol = 3)

# lists
list1 <- list(first = TRUE, second = 1:100, third = "long description of something")
list1

# data frames combine features of matrices and lists
# ideal for storing data
df1 <- data.frame(a = c(T,T,F,F,T), b = 1:5, c = c("a", "b", "c", "d", "e"))
df1

df2 <- data.frame(a = c(T,T,F,F,T), b = 1:5, c = c("a", "b", "c", "d", "e"), stringsAsFactors = F)
df2

# and operator &
TRUE & TRUE
FALSE & TRUE
FALSE & FALSE

# or operator |
TRUE | TRUE
FALSE | TRUE
FALSE | FALSE

# xor operator
xor(TRUE, TRUE)
xor(FALSE, TRUE)

```

```
xor(FALSE, FALSE)
```

## Statistical modelling of non-linear curves in R or R-Studio

```
# loading libraries
```

```
> library(drc)
```

```
> library(sandwich)
```

```
> library(lmtest)
```

```
# loading florasulam data
```

```
> florasulam <-read.table (file.choose(), sep=";", dec=".", header=TRUE)
```

```
> florasulam
```

	herbicide	dose	biotype	response	fresh.matter	dry.matter
1	florasulam	0.0010	R	0	8.433	1.277
2	florasulam	0.0010	R	0	6.537	1.000
3	florasulam	0.0010	R	0	5.586	0.783
4	florasulam	0.0010	R	0	4.576	0.737
5	florasulam	0.0158	R	5	2.626	0.481
6	florasulam	0.0158	R	5	4.003	0.725
7	florasulam	0.0158	R	5	4.686	0.877
8	florasulam	0.0158	R	10	3.755	0.738
9	florasulam	0.0500	R	10	3.501	0.595
10	florasulam	0.0500	R	5	5.788	0.879
11	florasulam	0.0500	R	5	4.546	0.881
12	florasulam	0.0500	R	10	5.007	0.807
13	florasulam	0.1580	R	10	3.804	0.611
14	florasulam	0.1580	R	15	3.545	0.655
15	florasulam	0.1580	R	10	3.703	0.612
16	florasulam	0.1580	R	10	4.867	0.771
17	florasulam	0.5000	R	15	4.423	0.595
18	florasulam	0.5000	R	15	3.347	0.447
19	florasulam	0.5000	R	15	3.938	0.521
20	florasulam	0.5000	R	5	5.836	0.763
21	florasulam	1.5800	R	10	5.498	0.573
22	florasulam	1.5800	R	20	1.634	0.322
23	florasulam	1.5800	R	15	4.071	0.626
24	florasulam	1.5800	R	15	4.571	0.671
25	florasulam	5.0000	R	35	1.966	0.303
26	florasulam	5.0000	R	40	1.649	0.262
27	florasulam	5.0000	R	35	2.021	0.227
28	florasulam	5.0000	R	35	1.675	0.244
29	florasulam	15.8000	R	55	1.275	0.159
30	florasulam	15.8000	R	55	1.698	0.177
31	florasulam	15.8000	R	55	1.379	0.139
32	florasulam	15.8000	R	60	0.830	0.130
33	florasulam	50.0000	R	80	0.505	0.076
34	florasulam	50.0000	R	80	0.186	0.062
35	florasulam	50.0000	R	80	0.588	0.090
36	florasulam	50.0000	R	80	0.728	0.101
37	florasulam	158.0000	R	85	0.240	0.079
38	florasulam	158.0000	R	85	0.133	0.055

39	florasulam	158.0000	R	85	0.262	0.072
40	florasulam	158.0000	R	85	0.407	0.036
41	florasulam	0.0010	S	0	1.301	0.329
42	florasulam	0.0010	S	0	3.631	0.491
43	florasulam	0.0010	S	0	3.627	0.567
44	florasulam	0.0010	S	0	4.632	0.746
45	florasulam	0.0158	S	65	0.314	0.061
46	florasulam	0.0158	S	70	0.362	0.076
47	florasulam	0.0158	S	68	0.248	0.061
48	florasulam	0.0158	S	70	0.268	0.070
49	florasulam	0.0500	S	70	0.181	0.062
50	florasulam	0.0500	S	75	0.156	0.044
51	florasulam	0.0500	S	75	0.100	0.044
52	florasulam	0.0500	S	75	0.210	0.071
53	florasulam	0.1580	S	85	0.078	0.038
54	florasulam	0.1580	S	85	0.063	0.029
55	florasulam	0.1580	S	85	0.067	0.032
56	florasulam	0.1580	S	85	0.068	0.027
57	florasulam	0.5000	S	90	0.048	0.019
58	florasulam	0.5000	S	90	0.052	0.025
59	florasulam	0.5000	S	90	0.043	0.025
60	florasulam	0.5000	S	90	0.111	0.034
61	florasulam	1.5800	S	95	0.047	0.025
62	florasulam	1.5800	S	95	0.069	0.023
63	florasulam	1.5800	S	95	0.039	0.023
64	florasulam	1.5800	S	95	0.045	0.031
65	florasulam	5.0000	S	95	0.054	0.026
66	florasulam	5.0000	S	95	0.042	0.022
67	florasulam	5.0000	S	95	0.054	0.036
68	florasulam	5.0000	S	95	0.046	0.023
69	florasulam	15.8000	S	95	0.048	0.026
70	florasulam	15.8000	S	95	0.052	0.028
71	florasulam	15.8000	S	95	0.049	0.026
72	florasulam	15.8000	S	95	0.038	0.026
73	florasulam	50.0000	S	95	0.036	0.024
74	florasulam	50.0000	S	95	0.052	0.034
75	florasulam	50.0000	S	95	0.049	0.026
76	florasulam	50.0000	S	95	0.041	0.026
77	florasulam	158.0000	S	95	0.041	0.029
78	florasulam	158.0000	S	98	0.038	0.025
79	florasulam	158.0000	S	98	0.045	0.027
80	florasulam	158.0000	S	98	0.039	0.025

# display dose vs response data

```
> plot(florasulam$dose, florasulam$response)
```

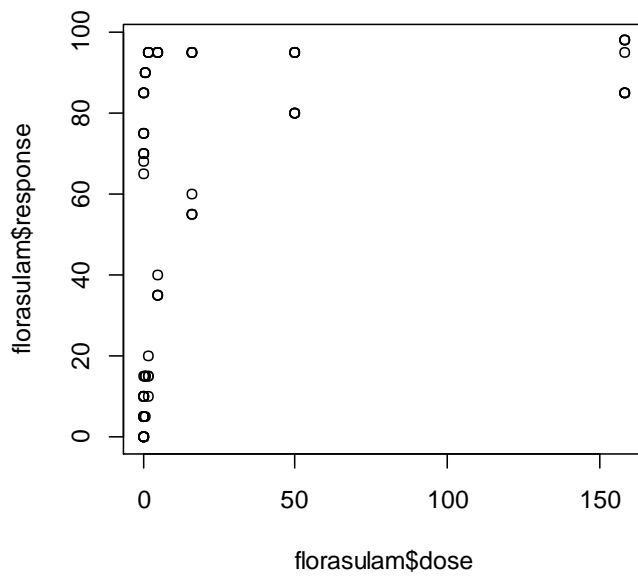


Fig. 26

# with differently colored biotypes

```
> plot(florasulam$dose, florasulam$response, col = ifelse(florasulam$biotype == "R", "darkred", "black"), pch = ifelse(florasulam$biotype == "R", 0, 5))
```

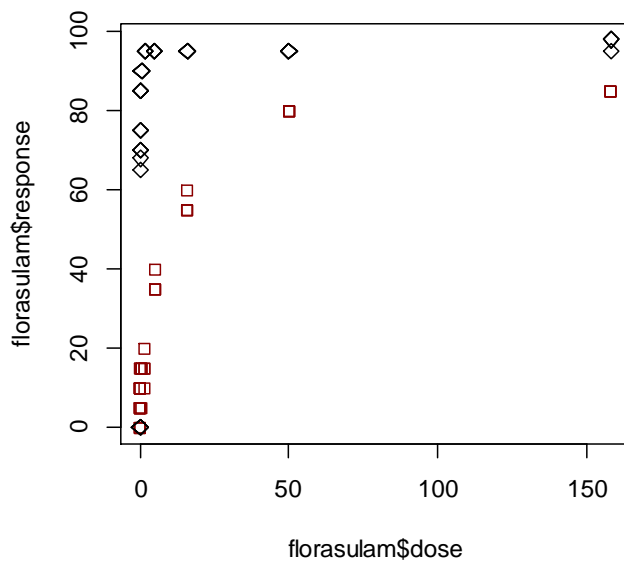


Fig. 27

# on logarithmic scale

```
> plot(florasulam$dose, florasulam$response, log = "x", col = ifelse(florasulam$biotype == "R", "darkred", "black"), pch = ifelse(florasulam$biotype == "R", 0, 5))
```

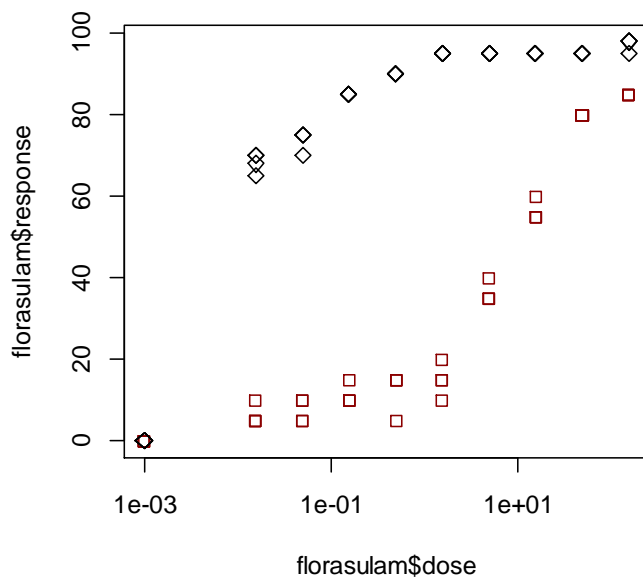


Fig. 28  
# fitting both biotypes simultaneously

```
> m<-drm(response~dose, biotype, data=florasulam, fct=LL.4(fixed = c(NA, NA, NA, NA)))
> plot(m)
```

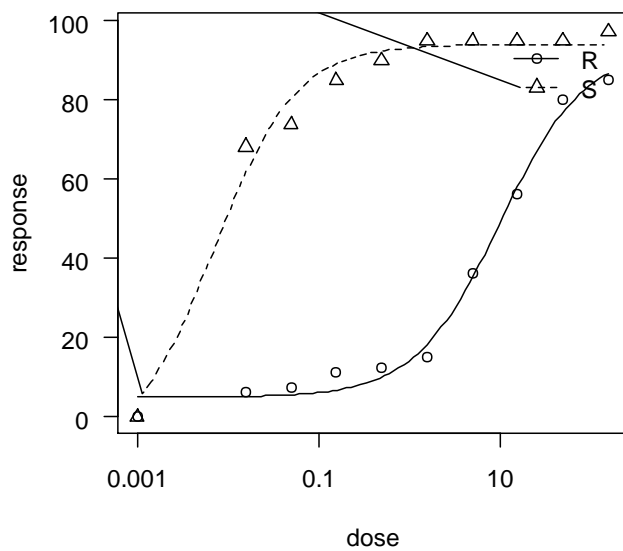


Fig. 29  
# summary of models with estimated parameters  
> summary(m)  
Model fitted: Log-logistic (ED50 as parameter) (4 parms)

Parameter estimates:

Estimate	Std. Error	t-value	p-value
----------	------------	---------	---------



```

b:R -0.9314994  0.1064109 -8.7537942  0.0000
b:S -0.9693657  0.0902818 -10.7371061  0.0000
c:R  5.0138499  1.1952688  4.1947469  0.0001
c:S -8.8159397  4.3744323 -2.0153335  0.0476
d:R 93.0604593  3.9158563 23.7650343  0.0000
d:S 94.0131606  0.9221261 101.9526106  0.0000
e:R 10.1112981  1.3621863  7.4228453  0.0000
e:S  0.0071609  0.0010014  7.1510506  0.0000

```

Residual standard error:

4.091408 (72 degrees of freedom)

# creating dataframes for resistant and susceptible biotypes for separate models fitting

```

> florasulam_R <- florasulam[florasulam$biotype=="R",]
> florasulam_S <- florasulam[florasulam$biotype=="S",]

```

#### LL4 model fitting for resistant biotype

```

> florasulam_R_LL4 <- drm(response ~ dose, data = florasulam_R, fct = LL.4(fixed = c(NA, NA, NA,
NA)))

```

```

> summary (florasulam_R_LL4)

```

Model fitted: Log-logistic (ED50 as parameter) (4 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-0.96735	0.10189	-9.49415	0
c:(Intercept)	5.70023	1.09852	5.18900	0
d:(Intercept)	92.23628	3.49624	26.38156	0
e:(Intercept)	10.09029	1.24072	8.13261	0

Residual standard error:

3.923603 (36 degrees of freedom)

# lack-of-fit test

```

> modelFit(florasulam_R_LL4)

```

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	30	225.00			
DRC model	36	554.21	6	7.3157	0.0001

# plot fitted curve

```

> plot(florasulam_R_LL4, broken = T, ylim = c(0, 100), xlim = c(0, 1000), col = "darkred", pch = 0,
col.main = "darkred", xlab = expression("Florasulam rate [g Al ha^-1*]"), ylab = "efficacy [%]", type
="all", main = "florasulam_R_LL4")

```

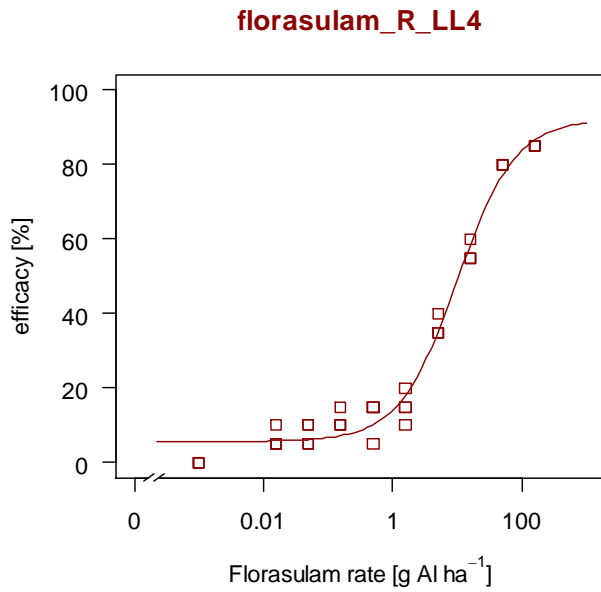


Fig. 30

# add 95 % confidence interval

```
> pm <- predict(florasulam_R_LL4, newdata = data.frame(dose=exp(seq(log(0.0005), log(200),
length=100))), interval="confidence")
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 2], lty = 2)
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 3], lty = 2)
```

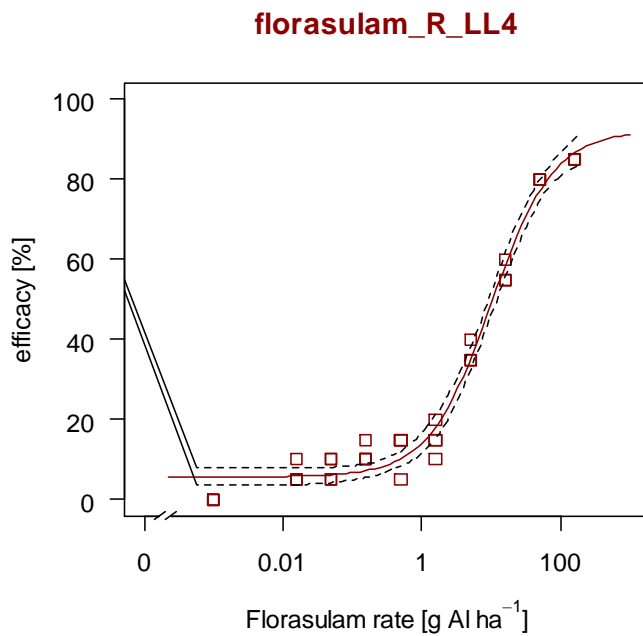


Fig. 31

# add ED10, ED50 and ED90

```
> abline(v = ED(florasulam_R_LL4, c(10,50,90))[1], col = "darkred")
```

### florasulam\_R\_LL4

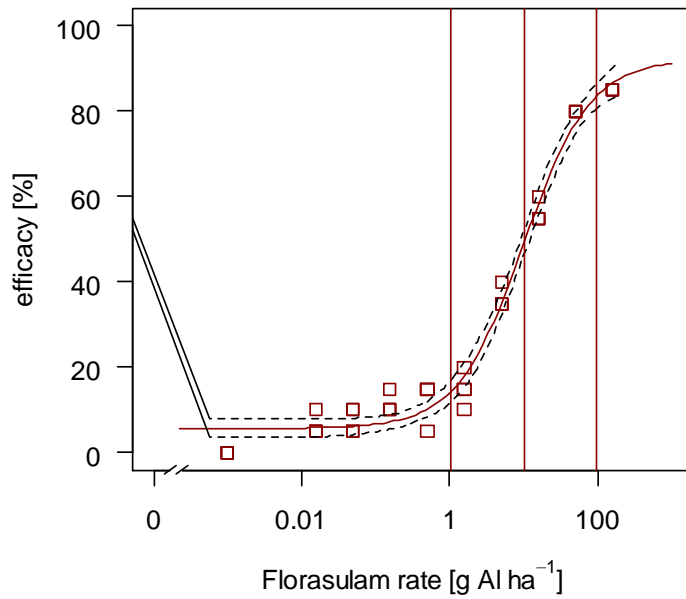


Fig. 32

# see four estimated parameters

```
> summary(florasulam_R_LL4)
```

Model fitted: Log-logistic (ED50 as parameter) (4 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-0.96735	0.10189	-9.49415	0
c:(Intercept)	5.70023	1.09852	5.18900	0
d:(Intercept)	92.23628	3.49624	26.38156	0
e:(Intercept)	10.09029	1.24072	8.13261	0

Residual standard error:

3.923603 (36 degrees of freedom)

```
coefest(florasulam_R_LL4, vcov = sandwich)
```

```
> abline(v = 10.0903 + 1.96*0.813)
```

```
> abline(v = 10.0903 - 1.96*0.813)
```

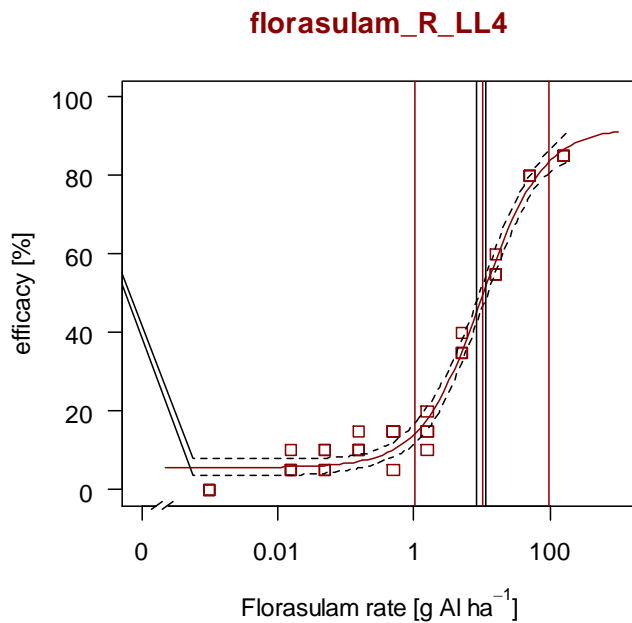


Fig. 33

```
#### LL4 model with fixed boundaries at 0 and 100
```

```
> florasulam_R_LL4.0_100fixed <- drm(response ~ dose, data = florasulam_R, fct = LL.4(fixed = c(NA, 0, 100, NA)))
```

```
> summary (florasulam_R_LL4.0_100fixed)
```

Model fitted: Log-logistic (ED50 as parameter) (2 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-0.702917	0.039974	-17.584333	0
e:(Intercept)	10.543756	0.797897	13.214427	0

Residual standard error:

4.81359 (38 degrees of freedom)

```
# plot fitted curve, add it to previous one by setting parameter new = T
```

```
> par(new=T)
```

```
> plot(florasulam_R_LL4.0_100fixed, broken = T, ylim = c(0, 100), xlim = c(0, 1000), col = "darkblue", pch = 0, cex = 0.4, xlab = expression("Florasulam rate [g AI ha^-1*]"), ylab = "efficacy [%]", type = "all", main = ""), title("florasulam_R_LL4 with fixed boudaries", line = 0.5, col.main = "darkblue")
```

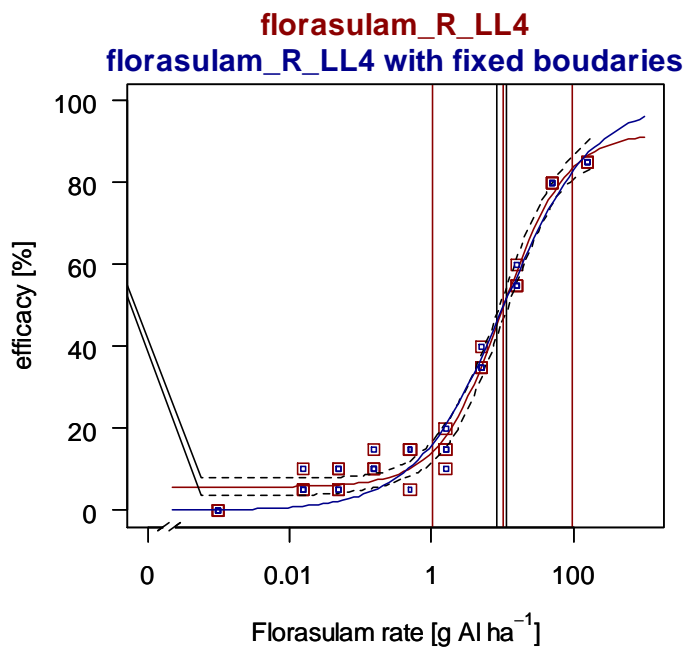


Fig. 34

# add 95 % confidence interval

```
> pm <- predict(florasulam_R_LL4.0_100fixed , newdata = data.frame(dose=exp(seq(log(0.0005),
log(200), length=100))), interval="confidence")
```

```
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 2], lty = 2)
```

```
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 3], lty = 2)
```

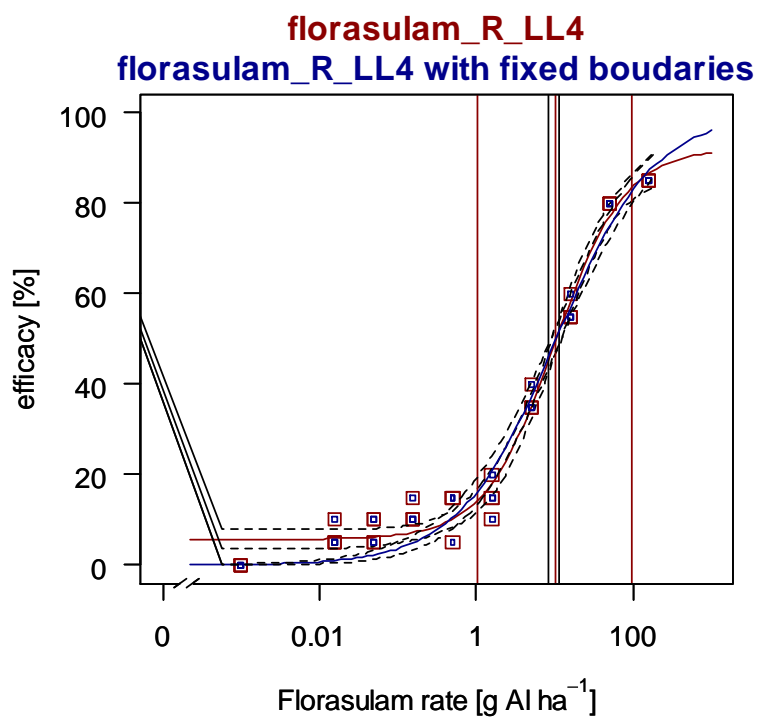


Fig. 35

# add ED10, ED50 and ED90

```
> abline(v = ED(florasulam_R_LL4.0_100fixed, c(10,50,90))[,1], col = "darkblue")
```

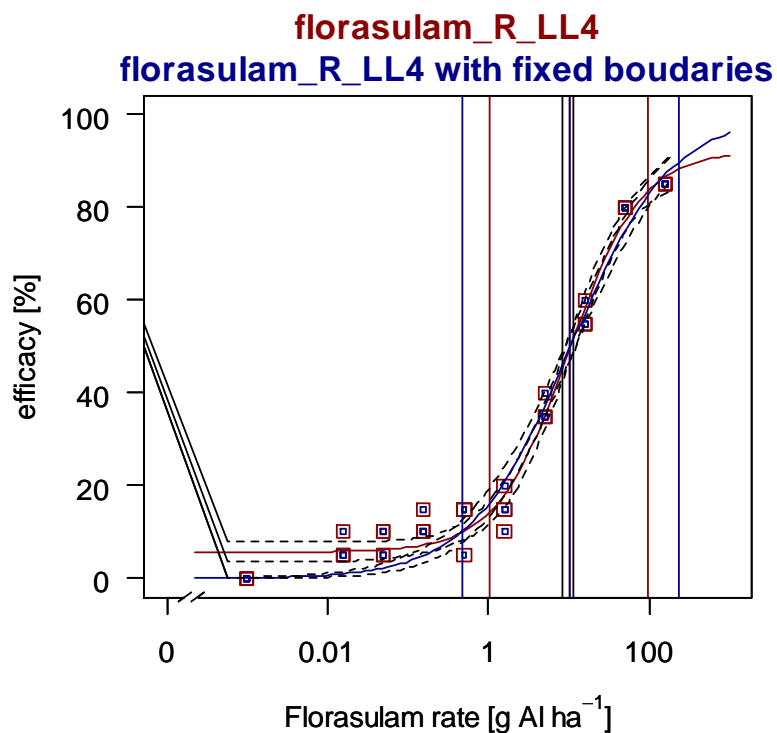


Fig. 36

# see estimated parameters

> `summary(florasulam_R_LL4.0_100fixed)`

Model fitted: Log-logistic (ED50 as parameter) (2 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-0.702917	0.039974	-17.584333	0
e:(Intercept)	10.543756	0.797897	13.214427	0

Residual standard error:

4.81359 (38 degrees of freedom)

# check both models for Lack of fit

> `modelFit(florasulam_R_LL4)`

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	30	225.00			
DRC model	36	554.21	6	7.3157	0.0001

> `modelFit(florasulam_R_LL4.0_100fixed)`

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	30	225.00			
DRC model	38	880.48	8	10.925	0.000

```

# sum of squares model
> sum((florasulam_R$response - predict(florasulam_R_LL4))^2)
[1] 554.2078

# sum of squares ANOVA
> anova_flu <- 0
> for(i in 1:10){anova_flu <- anova_flu + sum((florasulam_R$response[(4*i-3):(4*i)] -
mean(florasulam_R$response[(4*i-3):(4*i)]))^2)}
> anova_flu
[1] 225
# or:
> aov(response ~ as.factor(dose), data = florasulam_R)
Call:
> aov(formula = response ~ as.factor(dose), data = florasulam_R)

Terms:
      as.factor(dose) Residuals
Sum of Squares      36385      225
Deg. of Freedom       9      30

Residual standard error: 2.738613
Estimated effects may be unbalanced
# see how other models/curve types perform
> mselect(florasulam_R_LL4, list(LL.2(), LL.3(), LL.4(fixed = c(NA, 0, 100, NA)), LL.5(), LL.5(fixed = c(NA,
0, 100, NA, NA)), W1.3(), W1.4(), W2.4(), baro5()))
      logLik      IC Lack of fit Res var
LL.5 -105.6413 223.2826 4.204130e-04 13.16709
W2.4 -106.6765 223.3529 4.279719e-04 13.48134
baro5 -106.3067 224.6135 2.658502e-04 13.61253
LL.4 -109.3307 228.6615 7.010636e-05 15.39466
W1.4 -113.5434 237.0867 3.708455e-06 19.00405
LL.5 -114.6327 237.2654 3.664165e-06 19.52544
LL.4 -118.5894 243.1788 4.648271e-07 23.17065
LL.3 -118.2652 244.5303 2.834733e-07 23.41420
W1.3 -125.4626 258.9253 1.617078e-09 33.55604
LL.2    NA    NA    NA    NA

#####
##### try LL5 models
> florasulam_R_LL5 <- drm(response ~ dose, data = florasulam_R, fct = LL.5(fixed = c(NA, NA, NA, NA,
NA)))

# plot fitted curve
> plot(florasulam_R_LL5, broken = T, ylim = c(0, 100), xlim = c(0, 1000), col = "darkred", pch = 0, xlab
= expression("Florasulam rate [g Al ha^-1*]"), ylab = "efficacy [%]", type = "all")

```

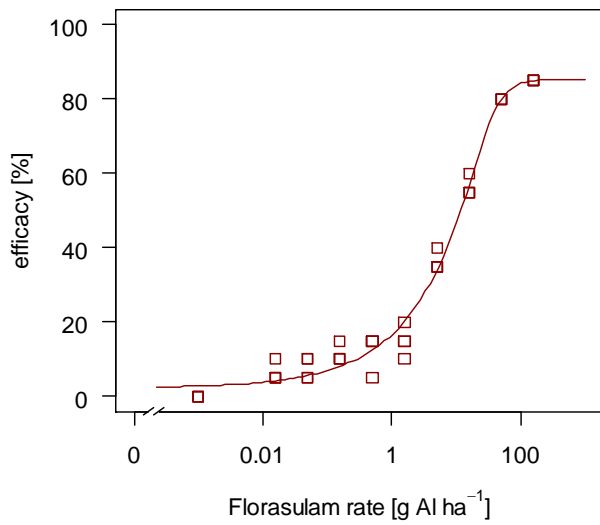


Fig. 37

# add 95 % confidence interval

```
> pm <- predict(florasulam_R_LL5 , newdata = data.frame(dose=exp(seq(log(0.0005), log(200),
length=100))), interval="confidence")
```

```
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 2], lty = 2)
```

```
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 3], lty = 2)
```

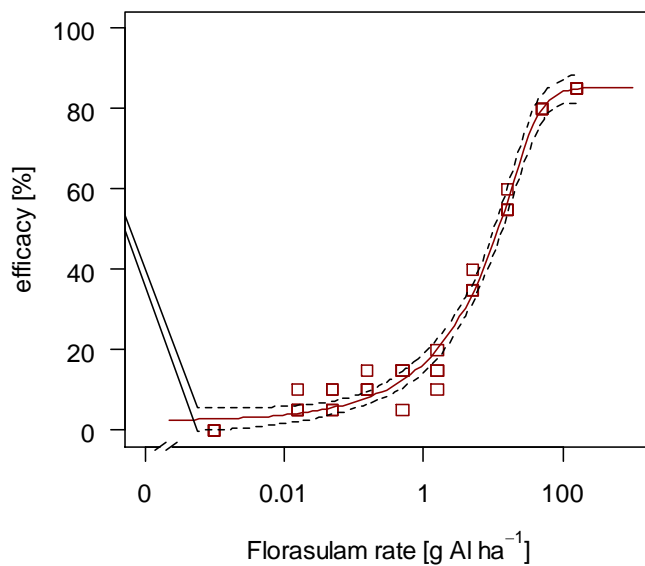


Fig. 38

# add ED10, ED50 and ED90

```
> abline(v = ED(florasulam_R_LL5, c(10,50,90))[,1], col = "darkred")
```

Estimated effective doses



	Estimate	Std. Error
1:10	0.33027	0.1473
1:50	8.85766	0.8050
1:90	40.45144	8.0495

# see estimated parameters

> summary(florasulam\_R\_LL5)

Model fitted: Generalized log-logistic (ED50 as parameter) (5 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-2.579198	0.995637	-2.590501	0.0139
c:(Intercept)	2.333519	1.569251	1.487027	0.1460
d:(Intercept)	85.347408	2.157195	39.564067	0.0000
e:(Intercept)	35.985451	6.792509	5.297814	0.0000
f:(Intercept)	0.190313	0.090166	2.110682	0.0420

Residual standard error:

3.628648 (35 degrees of freedom)

#### LL5 model with fixed boundaries at 0 and 100

> florasulam\_R\_LL5.0\_100fixed <- drm(response ~ dose, data = florasulam\_R, fct = LL.5(fixed = c(NA, 0, 100, NA, NA)))

# plot fitted curve, add it to previous one by setting parameter new = T

> par(new=T)

> plot(florasulam\_R\_LL5.0\_100fixed, broken = T, ylim = c(0, 100), xlim = c(0, 1000), col = "darkblue", pch = 0, xlab = expression("Florasulam rate [g AI ha<sup>-1</sup>"]"), ylab = "efficacy [%]", type = "all")

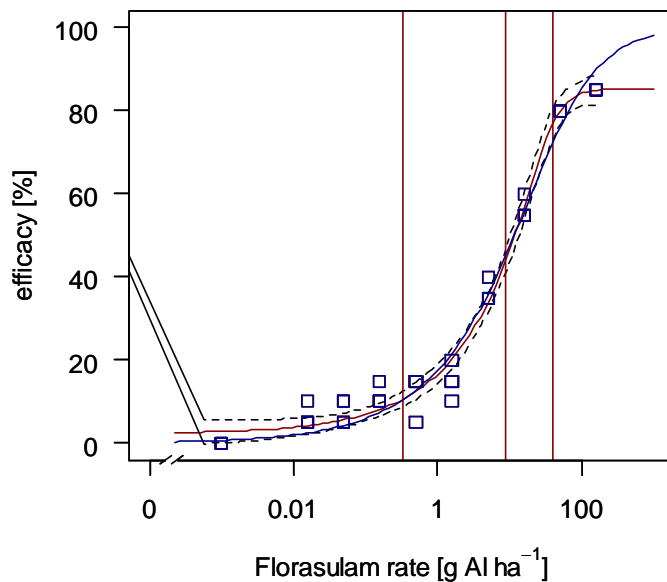


Fig. 39

# add 95 % confidence interval

```

> pm <- predict(florasulam_R_LL5.0_100fixed , newdata = data.frame(dose=exp(seq(log(0.0005),
log(200), length=100))), interval="confidence")
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 2], lty = 2)
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 3], lty = 2)

```

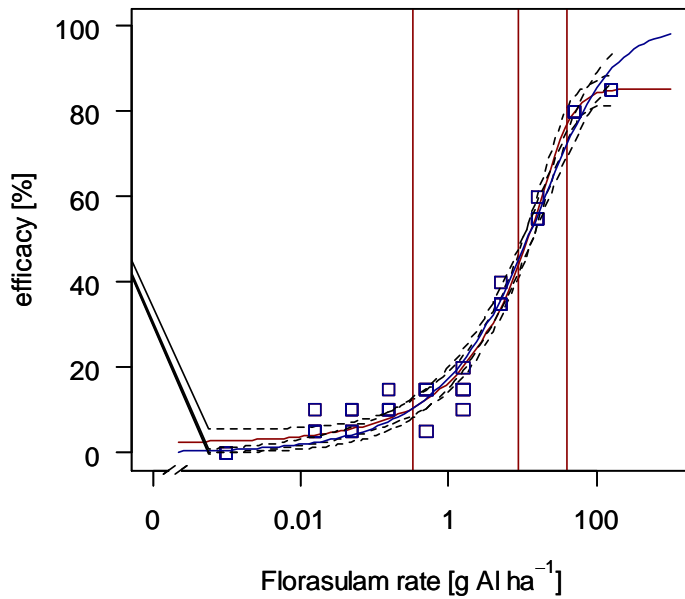


Fig. 40

```

# add ED10, ED50 and ED90
> abline(v = ED(florasulam_R_LL5.0_100fixed, c(10,50,90))[,1], col = "darkblue")

```

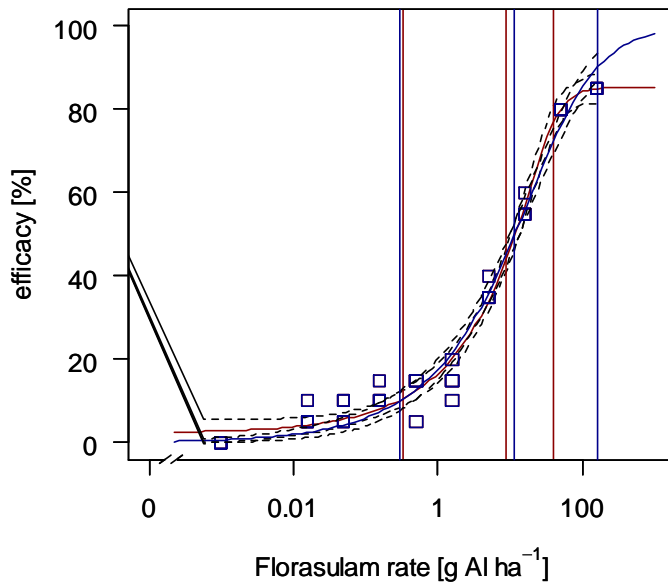


Fig. 41

```

# see estimated parameters
> summary(florasulam_R_LL5.0_100fixed)
fitted: Generalized log-logistic (ED50 as parameter) (3 parms)

```

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-0.98460	0.14098	-6.98411	0.0000
e:(Intercept)	38.62160	12.87850	2.99892	0.0048
f:(Intercept)	0.47976	0.11076	4.33149	0.0001

Residual standard error:

4.41876 (37 degrees of freedom)

# are there significant difference between models?

```
> anova(florasulam_R_LL4, florasulam_R_LL4.0_100fixed)
```

1st model

```
fct: LL.4(fixed = c(NA, 0, 100, NA))
```

2nd model

```
fct: LL.4(fixed = c(NA, NA, NA, NA))
```

ANOVA table

	ModelDf	RSS	Df	F value	p value
2nd model	38	880.48			
1st model	36	554.21	2	10.5971	0.0002

```
> anova(florasulam_R_LL4, florasulam_R_LL5)
```

1st model

```
fct: LL.4(fixed = c(NA, NA, NA, NA))
```

2nd model

```
fct: LL.5(fixed = c(NA, NA, NA, NA, NA))
```

ANOVA table

	ModelDf	RSS	Df	F value	p value
1st model	36	554.21			
2nd model	35	460.85	1	7.0904	0.0116

```
> anova(florasulam_R_LL4.0_100fixed, florasulam_R_LL5.0_100fixed)
```

1st model

```
fct: LL.4(fixed = c(NA, 0, 100, NA))
```

2nd model

```
fct: LL.5(fixed = c(NA, 0, 100, NA, NA))
```

ANOVA table

	ModelDf	RSS	Df	F value	p value
1st model	38	880.48			
2nd model	37	722.44	1	8.0942	0.0072

```
> anova(florasulam_R_LL5, florasulam_R_LL5.0_100fixed)
```

1st model

```
fct: LL.5(fixed = c(NA, 0, 100, NA, NA))
```

2nd model

```
fct: LL.5(fixed = c(NA, NA, NA, NA, NA))
```

ANOVA table

```
ModelDf  RSS Df F value p value
2nd model  37 722.44
1st model  35 460.85  2  9.9336 0.0004
```

```
#####
#####
# THE SAME FOR SUCEPTIBLE BIOTYPE #####
#####
```

# fit and plot model and see summary:

```
> florasulam_S_LL4 <- drm(response ~ dose, data = florasulam_S, fct = LL.4(fixed = c(NA, NA, NA, NA)))
> plot(florasulam_S_LL4, broken = T, ylim = c(0, 100), xlim = c(0, 1000), col = "darkred", pch = 0, xlab = expression("Florasulam rate [g AI ha-1*"]"), ylab = "efficacy [%]")
abline(v = ED(florasulam_S_LL4, c(10,50,90))[,1], col = "darkred")
```

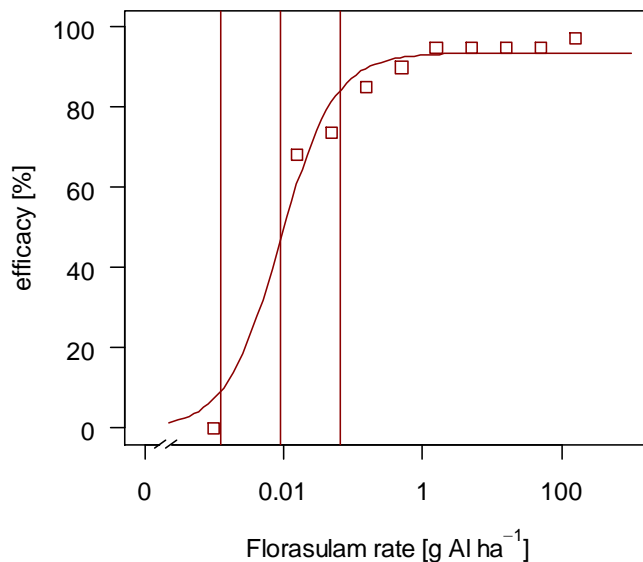


Fig. 42  
Estimated effective doses

```
Estimate Std. Error
1:10 0.0012522  0.0004
1:50 0.0089419  0.0012
1:90 0.0638515  0.0139
```

# add 95 % confidence interval

```
> pm <- predict(florasulam_S_LL4 , newdata = data.frame(dose=exp(seq(log(0.0005), log(200), length=100))), interval="confidence")
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 2], lty = 2)
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 3], lty = 2)
```

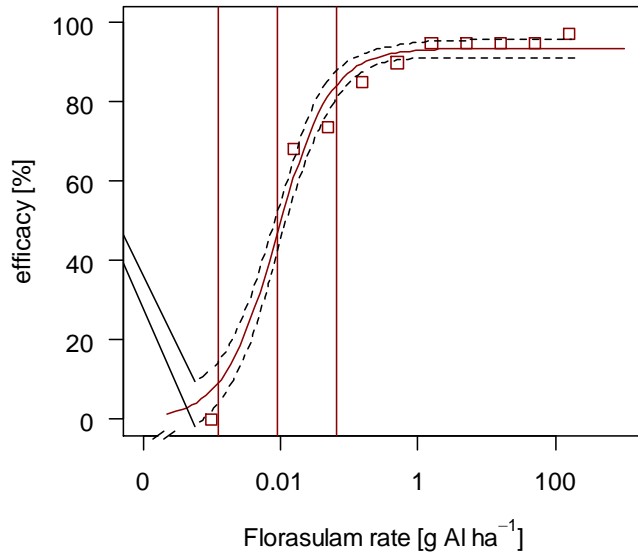


Fig. 43

```
> summary(florasulam_S_LL4)
```

Model fitted: Log-logistic (ED50 as parameter) (4 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-1.1177190	0.1320106	-8.4668883	0.0000
c:(Intercept)	-0.1199149	3.8222678	-0.0313727	0.9751
d:(Intercept)	93.5341073	1.1143033	83.9395373	0.0000
e:(Intercept)	0.0089419	0.0011795	7.5813841	0.0000

Residual standard error:

5.062525 (36 degrees of freedom)

# check for lack of fit and compare to other models

```
> modelFit(florasulam_S_LL4)
```

Lack-of-fit test

ModelDf	RSS	Df	F value	p value	
ANOVA	30	42.25			
DRC model	36	922.65	6	104.19	0.00

```
> mselect(florasulam_S_LL4, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4(), baro5()))
```

	logLik	IC	Lack of fit	Res var
baro5	-77.26231	166.5246	1.214809e-05	3.186011
W1.3	-104.81574	217.6315	1.376271e-13	11.951673
W2.4	-110.90242	231.8048	6.231031e-16	16.653238
LL.3	-119.63947	247.2789	2.316334e-18	25.079683
LL.4	-119.52495	249.0499	1.016897e-18	25.629161
LL.2	NA	NA	NA	NA
LL.5	NA	NA	NA	NA

W1.4    NA    NA    NA    NA

```
# try baro5 model
```

```
> florasulam_S_baro5 <- drm(response ~ dose, data = florasulam_S, fct = baro5())  
> plot(florasulam_S_baro5, broken = T, bp = 0.02, ylim = c(0, 100), xlim = c(0, 1000), lty = 2, pch = 5,  
xlab = "", ylab = "")
```

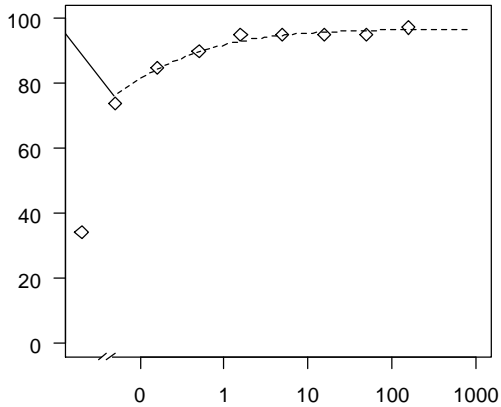


Fig. 44

```
> abline(v = ED(florasulam_S_baro5, c(10,50,90))[,1], col = "darkred")
```

```
> summary(florasulam_S_baro5)
```

Model fitted: Baroflex (5 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b1:(Intercept)	-0.2132061	0.0132989	-16.0318865	0.0000
b2:(Intercept)	-39.0452628	68.5721934	-0.5694037	0.5727
c:(Intercept)	0.1258881	0.8906347	0.1413465	0.8884
d:(Intercept)	96.8300102	0.5555880	174.2838302	0.0000
e:(Intercept)	0.0124812	0.0012095	10.3191830	0.0000

Residual standard error:

1.78494 (35 degrees of freedom)

```
# try W1.3 model
```

```
> florasulam_S_W1.3 <- drm(response ~ dose, data = florasulam_S, fct = W1.3())  
> plot(florasulam_S_W1.3, broken = T, ylim = c(0, 100), xlim = c(0, 1000), col = "darkred", pch = 0,  
xlab = expression("Florasulam rate [g AI ha^-1*]"), ylab = "efficacy [%]")
```

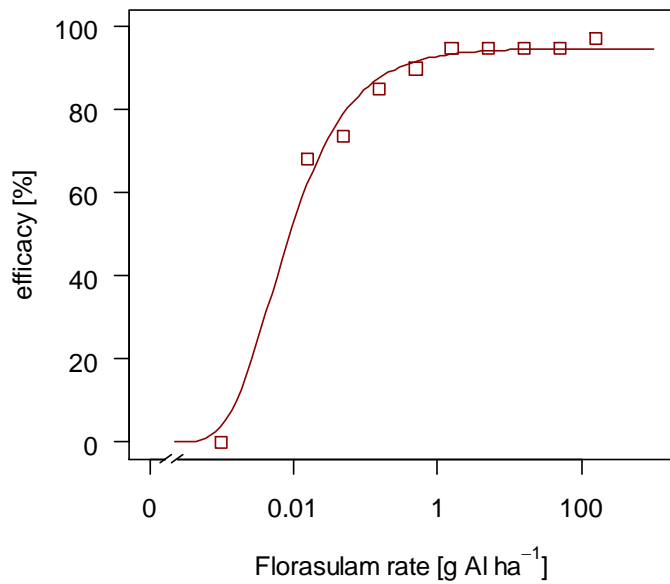


Fig. 45

```
> abline(v = ED(florasulam_S_W1.3, c(10,50,90))[,1], col = "darkred")
```

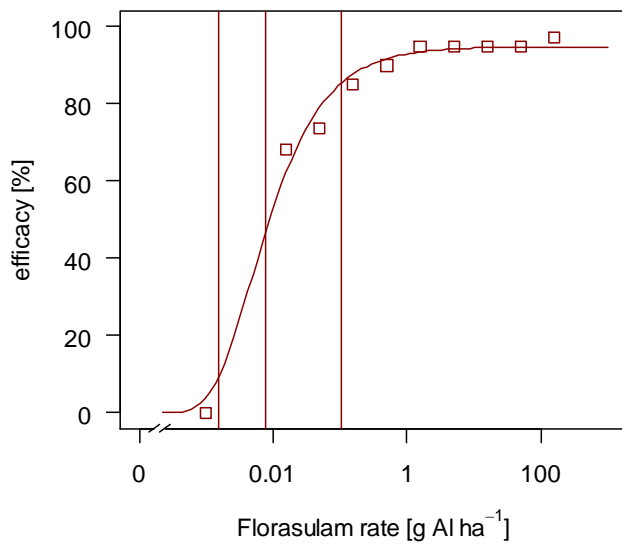


Fig. 46

effective doses

	Estimate	Std. Error
1:10	0.0015441	0.0002
1:50	0.0079242	0.0006
1:90	0.1031545	0.0172

```
# add 95 % confidence interval
```

```
> pm <- predict(florasulam_S_W1.3 , newdata = data.frame(dose=exp(seq(log(0.0005), log(200),  
length=100))), interval="confidence")  
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 2], lty = 2)  
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 3], lty = 2)
```

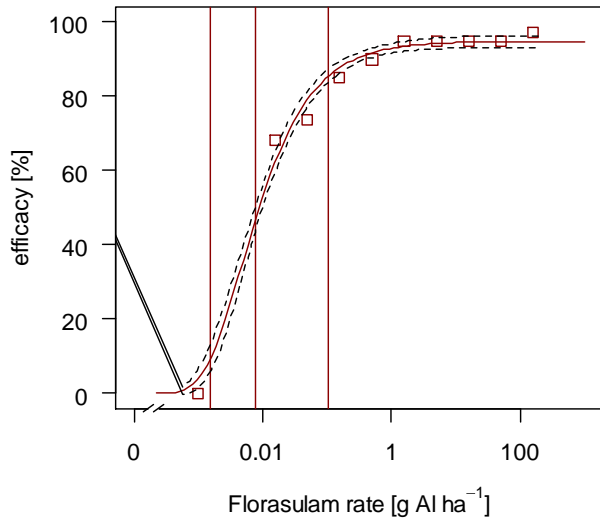


Fig. 47

```
> summary(florasulam_S_W1.3)
```

Model fitted: Weibull (type 1) with lower limit at 0 (3 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	-7.3407e-01	4.1508e-02	-1.7685e+01	0
d:(Intercept)	9.4849e+01	7.9943e-01	1.1865e+02	0
e:(Intercept)	4.8097e-03	3.5891e-04	1.3401e+01	0

Residual standard error:

3.457119 (37 degrees of freedom)

```
#####
```

```
#####
```

```
# plots in ggplot2:
```

```
> expand_dose <- exp(seq(log(0.001), log(200), length=100))
```



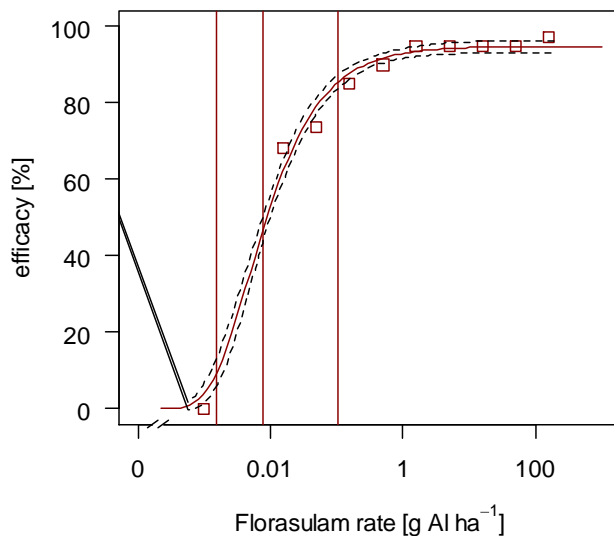


Fig. 48

```
## predictions and confidence intervals
```

```
> pm_R <- as.data.frame(predict(florasulam_R_LL4, newdata= data.frame(dose=expand_dose),
interval="confidence"))
```

```
> pm_S <- as.data.frame(predict(florasulam_S_LL4, newdata= data.frame(dose=expand_dose),
interval="confidence"))
```

```
#
```

```
## new data with predictions
```

```
> pm_R$dose <- expand_dose
```

```
> pm_S$dose <- expand_dose
```

```
#
```

```
## plotting the curve
```

```
> ggplot() + geom_point(data = florasulam_R,aes(x = dose, y = response), color = "red", shape = 18,
size = 2) + geom_point(data = florasulam_S,aes(x = dose, y = response), color = "black", shape = 16,
size = 2) + geom_ribbon(data = pm_R, aes(x = dose, ymin = Lower, ymax = Upper), alpha=0.2, fill =
"red") + geom_ribbon(data = pm_S, aes(x = dose, ymin = Lower, ymax = Upper), alpha=0.2) +
geom_line(data=pm_R, aes(x=dose, y=Prediction)) + geom_line(data=pm_S, aes(x=dose,
y=Prediction)) + coord_trans(x="log") + xlab(expression("Florasulam rate [g AI ha-1"]")) +
ylab("efficacy [%]")
```

```
# Fresh and dry matter
```

```
> plot(florasulam$dose, florasulam$fresh.matter, log = "x", col = ifelse(florasulam$biotype == "R",
"darkred", "black"), pch = ifelse(florasulam$biotype == "R", 0, 5))
```

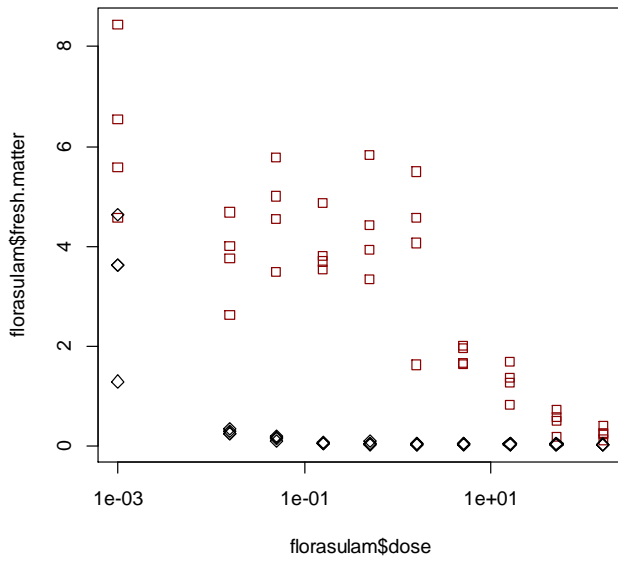


Fig. 49

```
> plot(florasulam$dose, florasulam$dry.matter, log = "x", col = ifelse(florasulam$biotype == "R",
"darkred", "black"), pch = ifelse(florasulam$biotype == "R", 0, 5))
```

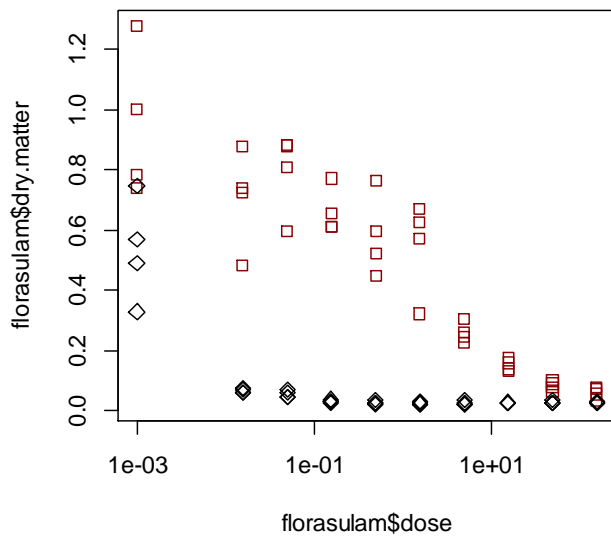


Fig. 50

```
> florasulam_FM_R_LL4 <- drm(fresh.matter ~ dose, data = florasulam_R, fct = LL.4(fixed = c(NA, NA,
NA, NA)))
> summary(florasulam_FM_R_LL4)
```

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	0.43690	0.42445	1.02932	0.3102

```

c:(Intercept) -1.14120  2.63193 -0.43360  0.6672
d:(Intercept)  5.53323  1.46148  3.78603  0.0006
e:(Intercept)  5.47024  6.35799  0.86037  0.3953

```

Residual standard error:

1.083403 (36 degrees of freedom)

```
plot(florasulam_FM_R_LL4)
```

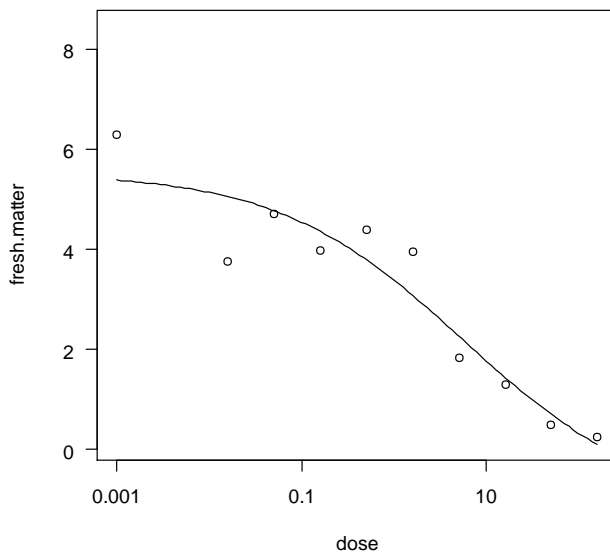


Fig. 51

```
> modelFit(florasulam_FM_R_LL4)
```

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	30	26.350			
DRC model	36	42.255	6	3.0181	0.0198

```
mselect(florasulam_FM_R_LL4, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
```

	logLik	IC	Lack of fit	Res var
W1.3	-57.50425	123.0085	3.990483e-02	1.122207
LL.3	-57.97717	123.9543	3.039950e-02	1.149058
W1.4	-57.50036	125.0007	2.459666e-02	1.153155
LL.4	-57.85462	125.7092	1.982947e-02	1.173763
W2.4	-58.61769	127.2354	1.238264e-02	1.219411
LL.5	-57.70770	127.4154	1.219109e-02	1.198463
LL.2	-100.16534	206.3307	9.089043e-15	9.222844

```
> florasulam_DM_R_LL4 <- drm(dry.matter ~ dose, data = florasulam_R, fct = LL.4(fixed = c(NA, NA, NA, NA)))
```

```
> summary(florasulam_DM_R_LL4)
```

Model fitted: Log-logistic (ED50 as parameter) (4 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	0.465944	0.216698	2.150203	0.0383
c:(Intercept)	-0.094946	0.188348	-0.504101	0.6173
d:(Intercept)	0.909282	0.112035	8.116057	0.0000
e:(Intercept)	2.205335	1.673274	1.317976	0.1958

Residual standard error:

0.1300009 (36 degrees of freedom)

```
> plot(florasulam_DM_R_LL4)
```

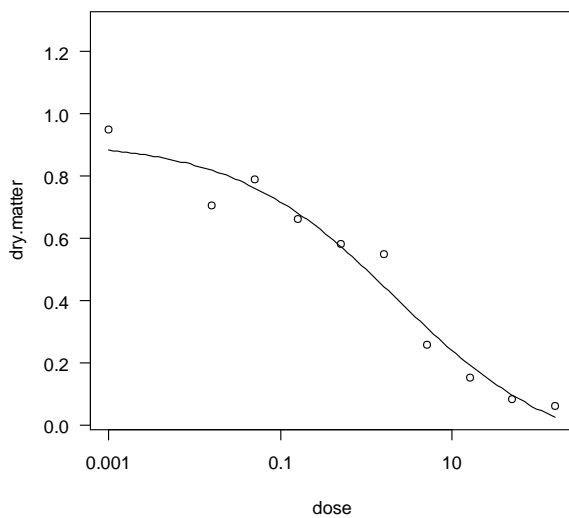


Fig. 52

```
> modelFit(florasulam_DM_R_LL4)
```

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	30	0.46975			
DRC model	36	0.60841	6	1.4758	0.2200

```
> mselect(florasulam_DM_R_LL4, list(LL.2(), LL.3(), LL.5()), W1.3(), W1.4(), W2.4()))
```

	logLik	IC	Lack of fit	Res var
W1.3	27.43008	-46.86015	0.3678616	0.01606007
LL.3	26.71643	-45.43287	0.2697590	0.01664347
W1.4	27.53490	-45.06980	0.2909383	0.01641990
LL.2	25.41233	-44.82466	0.1998832	0.01729738
LL.4	26.95822	-43.91644	0.2200403	0.01690024
LL.5	27.16548	-42.33096	0.1672485	0.01720389

W2.4 26.15223 -42.30445 0.1454695 0.01759522

# here, only few models actually converge, start with LL2

```
> florasulam_FM_S_LL2 <- drm(fresh.matter ~ dose, data = florasulam_S, fct = LL.2())  
summary(florasulam_FM_S_LL2)
```

Model fitted: Log-logistic (ED50 as parameter) with lower limit at 0 and upper limit at 1 (2 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	179.036116	10.000000	17.903612	0.0000
e:(Intercept)	0.376149	10.000000	0.037615	0.9702

Residual standard error:

0.9656149 (38 degrees of freedom)

```
> plot(florasulam_FM_S_LL2)
```

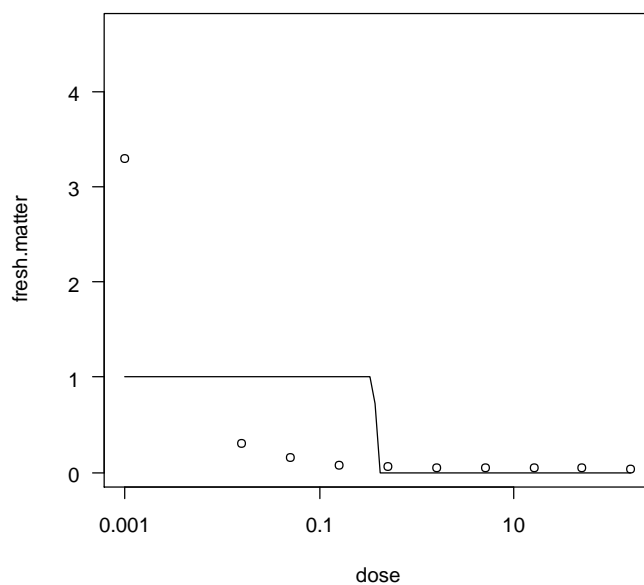


Fig. 53

```
> modelFit(florasulam_FM_S_LL2)
```

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	30	6.005			
DRC model	38	35.432	8	18.376	0.000

```
> mselect(florasulam_FM_S_LL2, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
```

	logLik	IC	Lack of fit	Res var
W1.3	-20.54798	49.09597	9.044245e-01	0.1768389
LL.2	-54.33207	114.66413	1.328459e-09	0.9324121

```

LL.2 -54.33207 114.66413 1.328459e-09 0.9324121
LL.3 NA NA NA NA
LL.5 NA NA NA NA
W1.4 NA NA NA NA
W2.4 NA NA NA NA

```

# here, only few models actually converge, start with LL2

```
> florasulam_DM_S_LL4 <- drm(dry.matter ~ dose, data = florasulam_S, fct = LL.2())
```

```
> summary(florasulam_DM_S_LL4)
```

istic (ED50 as parameter) with lower limit at 0 and upper limit at 1 (2 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:(Intercept)	8.8503e-01	1.2168e-01	7.2735e+00	0
e:(Intercept)	1.1471e-03	7.1189e-05	1.6113e+01	0

Residual standard error:

0.0542031 (38 degrees of freedom)

```
> plot(florasulam_DM_S_LL4)
```

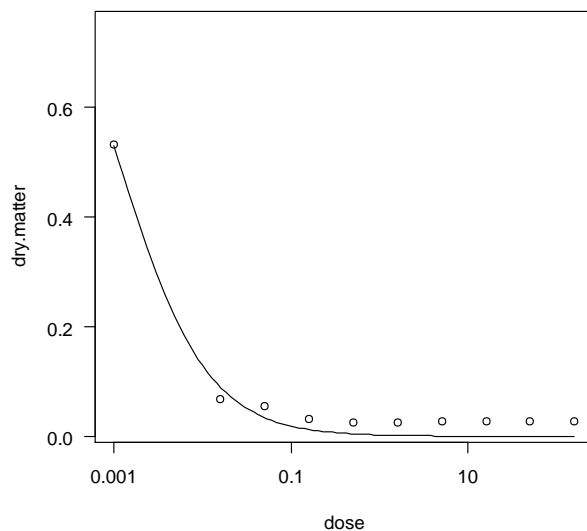


Fig. 54

```
> modelFit(florasulam_DM_S_LL4)
```

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	30	0.091036			
DRC model	38	0.111643	8	0.8489	0.5685

```
> mselect(florasulam_DM_S_LL4, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
```

	logLik	IC	Lack of fit	Res var
LL.2	60.86901	-115.738	0.5684599	0.002937976
LL.2	60.86901	-115.738	0.5684599	0.002937976

LL.3	NA	NA	NA	NA
LL.5	NA	NA	NA	NA
W1.3	NA	NA	NA	NA
W1.4	NA	NA	NA	NA
W2.4	NA	NA	NA	NA

##### COMPARE TWO biotypes

```
> two_biotypes <- rbind(florasulam_S, florasulam_R)
> two_biotypes.LL.4.0 <- drm(response ~ dose, data = two_biotypes, fct = LL.4())
> plot(two_biotypes.LL.4.0, type="all")
```

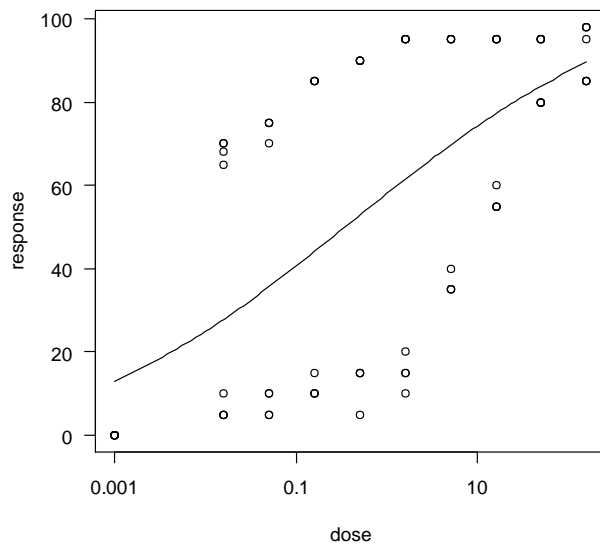


Fig. 55

```
> two_biotypes.LL.4.1 <- drm(response ~ dose, biotype, data = two_biotypes, fct = LL.4()),
pmodels=list(~biotype-1, ~1, ~1, ~biotype-1))
> plot(two_biotypes.LL.4.1, type="all")
```

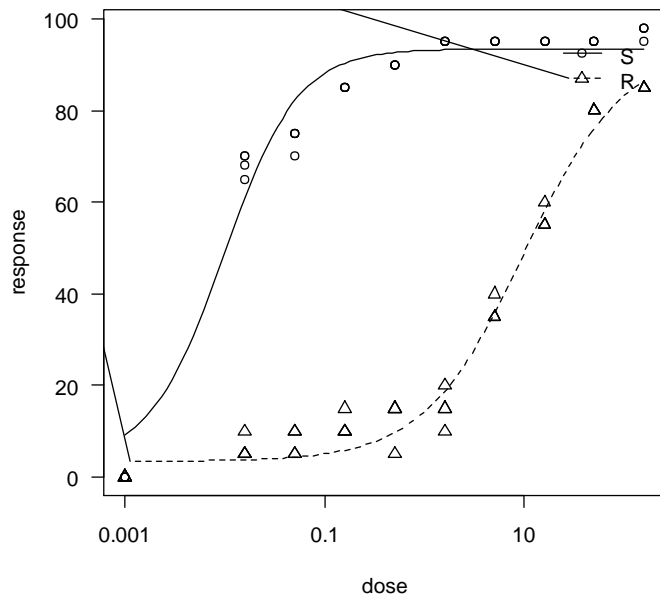


Fig. 56

```
> summary(two_biotypes.LL.4.1)
```

fitted: Log-logistic (ED50 as parameter) (4 parms)

Parameter estimates:

	Estimate	Std. Error	t-value	p-value
b:biotypeR	-8.7852e-01	7.1273e-02	-1.2326e+01	0.0000
b:biotypeS	-1.1768e+00	1.2222e-01	-9.6285e+00	0.0000
c:(Intercept)	3.4332e+00	1.2681e+00	2.7074e+00	0.0084
d:(Intercept)	9.3474e+01	1.0287e+00	9.0866e+01	0.0000
e:biotypeR	9.6633e+00	9.0103e-01	1.0725e+01	0.0000
e:biotypeS	9.7201e-03	9.2104e-04	1.0553e+01	0.0000

Residual standard error:

4.825739 (74 degrees of freedom)

```
> modelFit(two_biotypes.LL.4.1)
```

Lack-of-fit test

	ModelDf	RSS	Df	F value	p value
ANOVA	60	267.25			
DRC model	74	1723.29	14	23.35	0.00

```
> mselect(two_biotypes.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
```

	logLik	IC	Lack of fit	Res var
W1.4	-227.8951	469.7902	1.640459e-16	18.86790
LL.4	-236.3137	486.6275	3.826316e-19	23.28776
W2.4	-250.3981	514.7963	1.343332e-23	33.11670
W1.3	-346.6876	703.3752	5.614307e-54	358.02611



```
LL.3 -362.5719 735.1438 3.845854e-59 532.56955
LL.2 NA NA NA NA
LL.5 NA NA NA NA
```

```
> anova(two_biotypes.LL.4.1, two_biotypes.LL.4.0)
```

```
1st model
```

```
fct: LL.4()
```

```
pmodels: 1 (for all parameters)
```

```
2nd model
```

```
fct: LL.4()
```

```
pmodels: ~biotype - 1, ~1, ~1, ~biotype - 1
```

```
ANOVA table
```

Model	Df	RSS	Df	F value	p value
2nd model	76	66110			
1st model	74	1723	2	1382.4	0.0

```
#·EDcomp(two_biotypes.LL.4.1, c(10, 50, 50), interval = "delta")
```

```
> compParm(two_biotypes.LL.4.1, "e", "-")
```

```
Comparison of parameter 'e'
```

	Estimate	Std. Error	t-value	p-value
biotypeR-biotypesS	9.65354	0.90081	10.71647	0

##### fresh and dry matter

```
> two_biotypes.LL.4.1 <- drm(dry.matter ~ dose, biotype, data = two_biotypes fct = LL.4()),
pmodels=list(~biotype-1, ~1, ~1, ~biotype-1))
> plot(two_biotypes.LL.4.1, type="all")
```

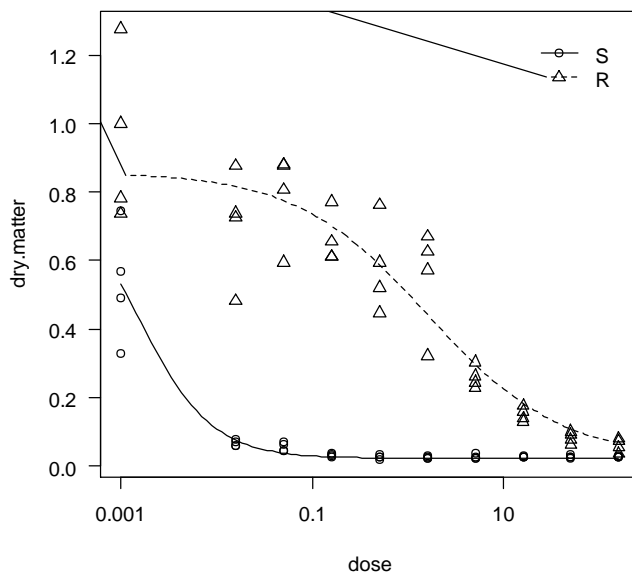


Fig. 57

```
> modelFit(two_biotypes.LL.4.1)
```

```
lack-of-fit test
```

	ModelDf	RSS	Df	F value	p value
ANOVA	60	0.56079			
DRC model	74	0.71445	14	1.1743	0.3180

```
> mselect(two_biotypes.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
```

### TRIBENURON

```
> tribenuron <- read.csv("tribenuron.csv", sep = ";", stringsAsFactors = FALSE)
> plot(tribenuron$dose, tribenuron$response, log = "x", col = ifelse(tribenuron$biotype == "R",
"darkred", "black"), pch = c(rep(0, sum(tribenuron$biotype=="R")), rep(5, sum(tribenuron$biotype
=="S"))))
> tribenuron_R <- tribenuron[tribenuron$biotype=="R",]
> tribenuron_S <- tribenuron[tribenuron$biotype=="S",]
> tribenuron_R_LL4 <- drm(response ~ dose, data = tribenuron_R, fct = LL.4(fixed = c(NA, NA, NA,
NA)))
> summary(tribenuron_R_LL4)
> modelFit(tribenuron_R_LL4)
> mselect(tribenuron_R_LL4, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
> tribenuron_S_LL4 <- drm(response ~ dose, data = tribenuron_S, fct = LL.4(fixed = c(NA, NA, NA,
NA)))
> summary(tribenuron_S_LL4)
> modelFit(tribenuron_S_LL4)
> mselect(tribenuron_S_LL4, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
> plot(tribenuron_R_LL4, broken = T, bp = 0.002, ylim = c(0, 100), xlim = c(0, 10000), col = "darkred",
pch = 0)
> par(new=T)
> plot(tribenuron_S_LL4, broken = T, bp = 0.002, ylim = c(0, 100), xlim = c(0, 10000), lty = 2, pch = 5)
> legend(200, 20, c("R", "S"), col = c("darkred", "black"), lty = c(1, 2), pch = c(0, 5), box.lty = 0)
```

### ### dry/fresh

```
> tribenuron_FM_R_LL4 <- drm(fresh.matter ~ dose, data = tribenuron_R, fct = W1.3())
> summary(tribenuron_FM_R_LL4)
> plot(tribenuron_FM_R_LL4)
> modelFit(tribenuron_FM_R_LL4)
> mselect(tribenuron_FM_R_LL4, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4(), baro5()))
> tribenuron_R_LL4 <- drm(dry.matter ~ dose, data = tribenuron_R, fct = LL.4(fixed = c(NA, 0, 1, NA)))
> summary(tribenuron_R_LL4)
> plot(tribenuron_R_LL4)
> tribenuron_S_LL4 <- drm(fresh.matter ~ dose, data = tribenuron_S, fct = LL.4(fixed = c(NA, NA, NA,
NA)))
> summary(tribenuron_S_LL4)
> plot(tribenuron_S_LL4)
> tribenuron_S_LL4 <- drm(dry.matter ~ dose, data = tribenuron_S, fct = LL.4(fixed = c(NA, NA, NA,
NA)))
> summary(tribenuron_S_LL4)
> plot(tribenuron_S_LL4)
```

### ### compare herbicides - susceptible biotype

```
> two_herbicides_S <- rbind(florasulam_S, tribenuron_S)
```

```

> two_herbicides_S.LL.4.1 <- drm(response ~ dose, herbicide, data = two_herbicides_S, fct = LL.4()),
pmodels=list(~herbicide-1, ~1, ~1, ~herbicide-1))
> summary(two_herbicides_S.LL.4.1)
> plot(two_herbicides_S.LL.4.1, type="all")
> two_herbicides_S.LL.4.2 <- drm(response ~ dose, data = two_herbicides_S, fct = LL.4())
> plot(two_herbicides_S.LL.4.1, type="all")
> anova(two_herbicides_S.LL.4.1, two_herbicides_S.LL.4.2)
> EDcomp(two_herbicides_S.LL.4.1, c(10, 50, 90), interval = "delta")
> compParm(two_herbicides_S.LL.4.1, "e", "-")
> comped()
> ED(two_herbicides_S.LL.4.1, 50)
> comped(ED(two_herbicides_S.LL.4.1, 50)[,1], ED(two_herbicides_S.LL.4.1, 50)[,2])
> modelFit(two_herbicides_S.LL.4.1)
> mselect(two_herbicides_S.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
> two_herbicides_S.LL.4.1 <- drm(fresh.matter ~ dose, herbicide, data = two_herbicides_S, fct =
LL.4()), pmodels=list(~herbicide-1, ~1, ~1, ~herbicide-1))
> plot(two_herbicides_S.LL.4.1, type="all")
> compParm(two_herbicides_S.LL.4.1, "e", "-")
> modelFit(two_herbicides_S.LL.4.1)
> mselect(two_herbicides_S.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
> two_herbicides_S.LL.4.1 <- drm(fresh.matter ~ dose, herbicide, data = two_herbicides_S, fct =
LL.4()), pmodels=list(~herbicide-1, ~1, ~1, ~herbicide-1))
> plot(two_herbicides_S.LL.4.1, type="all")
> compParm(two_herbicides_S.LL.4.1, "e", "-")
> modelFit(two_herbicides_S.LL.4.1)
> mselect(two_herbicides_S.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))

# compare herbicides - resistant biotype
> two_herbicides_R <- rbind(florasulam_R, tribenuron_R)
> two_herbicides_R.LL.4.1 <- drm(response ~ dose, herbicide, data = two_herbicides_R, fct = LL.4()),
pmodels=list(~herbicide-1, ~1, ~1, ~herbicide-1))
> summary(two_herbicides_R.LL.4.1)
> plot(two_herbicides_R.LL.4.1, type="all", col = c("red", "blue"))
> abline(v = ED(two_herbicides_R.LL.4.1, c(50))[,1], col = c("red", "blue"))
# add 95 % confidence interval
> pm <- predict(two_herbicides_R.LL.4.1 , newdata = data.frame(dose=exp(seq(log(0.0005),
log(200), length=100))), interval="confidence")
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 2], lty = 2)
> lines(exp(seq(log(0.0005), log(200), length=100)), pm[, 3], lty = 2)
> two_herbicides_R.LL.4.2 <- drm(response ~ dose, data = two_herbicides_R, fct = LL.4())
> anova(two_herbicides_R.LL.4.1, two_herbicides_R.LL.4.2)
> compParm(two_herbicides_R.LL.4.1, "e", "-")
> modelFit(two_herbicides_R.LL.4.1)
> mselect(two_herbicides_R.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))
> two_herbicides_R.LL.4.1 <- drm(fresh.matter ~ dose, herbicide, data = two_herbicides_R, fct =
LL.4()), pmodels=list(~herbicide-1, ~1, ~1, ~herbicide-1))
> plot(two_herbicides_R.LL.4.1, type="all", col = c("red", "blue"))
> abline(v = ED(two_herbicides_R.LL.4.1, c(50))[,1], col = c("red", "blue"))
> compParm(two_herbicides_R.LL.4.1, "e", "-")
> modelFit(two_herbicides_R.LL.4.1)
> mselect(two_herbicides_R.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))

```

```

> two_herbicides_R.LL.4.1 <- drm(dry.matter ~ dose, herbicide, data = two_herbicides_R, fct = LL.4(),
pmodels=list(~herbicide-1, ~1, ~1, ~herbicide-1))
> plot(two_herbicides_R.LL.4.1, type="all", col = c("red", "blue"))
> abline(v = ED(two_herbicides_R.LL.4.1, c(50))[,1], col = c("red", "blue"))
> compParm(two_herbicides_R.LL.4.1, "e", "-")
> modelFit(two_herbicides_R.LL.4.1)
> mselect(two_herbicides_R.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))

```

# better fitting LL.2() model

```

> two_herbicides_R.LL.4.1 <- drm(dry.matter ~ dose, herbicide, data = two_herbicides_R, fct = LL.2(),
pmodels=list(~herbicide-1, ~herbicide-1))
> plot(two_herbicides_R.LL.4.1, type="all", col = c("red", "blue"))
> abline(v = ED(two_herbicides_R.LL.4.1, c(50))[,1], col = c("red", "blue"))
> compParm(two_herbicides_R.LL.4.1, "e", "-")
> modelFit(two_herbicides_R.LL.4.1)
> mselect(two_herbicides_R.LL.4.1, list(LL.2(), LL.3(), LL.5(), W1.3(), W1.4(), W2.4()))

```

####

# Fresh and dry matter

```

> plot(florasulam_S_LL4, ylim = c(0, 100), xlim = c(0, 10000),col="red")
> par(new=T)
> plot(tribenuron_S_LL4, ylim = c(0, 100), xlim = c(0, 10000))
> EDcomp()
> log(unique(florasulam[florasulam$biotype=="S",]$dose))
> exp(1)^log(unique(florasulam[florasulam$biotype=="S",]$dose))
> log10(unique(florasulam[florasulam$biotype=="S",]$dose))
> log_florasulam_R <- lm(response ~ log10(dose), data = florasulam[florasulam$biotype=="S",])
> logpred <- predict(log_florasulam_R)
> lines(florasulam[florasulam$biotype=="S",]$dose, logpred)
> m <- nls(response ~ log(dose), data = florasulam[florasulam$biotype=="S",])
> x<-seq(0,50,1)
> y<-((runif(1,10,20)*x)/(runif(1,0,10)+x))+rnorm(51,0,1)

```

#for simple models nls find good starting values for the parameters even if it throw a warning

```

> m<-nls(y ~ a*x/(b+x))
#get some estimation of goodness of fit
> cor(y,predict(m))
> hill <- function(KA, L, n){1/(1 + (KA / L) ^ n)}
> plot(1:100,hill(1,1:100,1))

```

## References

- Archontoulis, S. V., Miguez, F. E. (2015) Nonlinear regression models and applications in agricultural research, *Agronomy journal*, Vo. 07, Issue 2.
- Baty, F., Ritz, Ch., Sandrine, C., Brutsche, M., Flandrois, J.-P., Delignette-Muller, M.-L. (2015) A toolbox for nonlinear regression in R: The Package nlstools. *Journal of Statistical Software*, Vol. 66, Issue 5.
- Belz, RG, Piepho H-P (2012) Modeling Effective Dosages in Hormetic Dose-Response Studies. *PLoS ONE* 7(3): e33432. doi:10.1371/journal.pone.0033432
- Ritz, C., Streibig, J. C. (2005). Bioassay Analysis using R. *Journal of Statistical Software*, **12**, 5. <http://www.jstatsoft.org/v12/i05>
- Ritz, Ch., Cedergreen, N., Jensen J.-E., Streibig, J. C. (2006) Relative potency in nonsimilar dose-response curves. *Weed Science*, 54:407-412.
- Ritz, C., Baty, F., Streibig, J. C., Gerhard, D. (2015). Dose-Response Analysis Using R. *PLoS ONE*, **10**, e0146021. <https://doi.org/10.1371/journal.pone.0146021>
- Ritz, C., Jensen, S. M., Gerhard, D., Streibig, J. C. (2019). Dose-Response Analysis Using R. CRC Press. *Forthcoming*. <https://www.crcpress.com/Dose-Response-Analysis-Using-R/Ritz-Jensen-Gerhard-Streibig/p/book/9781138034310>

## Supplementary data

### Data for analysis - florasulam

herbicide	dose	biotype	response	fresh matter	dry matter
florasulam	0.001	R	0	8.433	1.277
florasulam	0.001	R	0	6.537	1
florasulam	0.001	R	0	5.586	0.783
florasulam	0.001	R	0	4.576	0.737
florasulam	0.0158	R	5	2.626	0.481
florasulam	0.0158	R	5	4.003	0.725
florasulam	0.0158	R	5	4.686	0.877
florasulam	0.0158	R	10	3.755	0.738
florasulam	0.05	R	10	3.501	0.595
florasulam	0.05	R	5	5.788	0.879
florasulam	0.05	R	5	4.546	0.881
florasulam	0.05	R	10	5.007	0.807
florasulam	0.158	R	10	3.804	0.611
florasulam	0.158	R	15	3.545	0.655
florasulam	0.158	R	10	3.703	0.612
florasulam	0.158	R	10	4.867	0.771
florasulam	0.5	R	15	4.423	0.595
florasulam	0.5	R	15	3.347	0.447
florasulam	0.5	R	15	3.938	0.521
florasulam	0.5	R	5	5.836	0.763
florasulam	1.58	R	10	5.498	0.573
florasulam	1.58	R	20	1.634	0.322
florasulam	1.58	R	15	4.071	0.626
florasulam	1.58	R	15	4.571	0.671
florasulam	5	R	35	1.966	0.303
florasulam	5	R	40	1.649	0.262
florasulam	5	R	35	2.021	0.227
florasulam	5	R	35	1.675	0.244
florasulam	15.8	R	55	1.275	0.159
florasulam	15.8	R	55	1.698	0.177
florasulam	15.8	R	55	1.379	0.139
florasulam	15.8	R	60	0.83	0.13
florasulam	50	R	80	0.505	0.076
florasulam	50	R	80	0.186	0.062
florasulam	50	R	80	0.588	0.09
florasulam	50	R	80	0.728	0.101
florasulam	158	R	85	0.24	0.079
florasulam	158	R	85	0.133	0.055
florasulam	158	R	85	0.262	0.072
florasulam	158	R	85	0.407	0.036
florasulam	0.001	S	0	1.301	0.329

florasulam	0.001	S	0	3.631	0.491
florasulam	0.001	S	0	3.627	0.567
florasulam	0.001	S	0	4.632	0.746
florasulam	0.0158	S	65	0.314	0.061
florasulam	0.0158	S	70	0.362	0.076
florasulam	0.0158	S	68	0.248	0.061
florasulam	0.0158	S	70	0.268	0.07
florasulam	0.05	S	70	0.181	0.062
florasulam	0.05	S	75	0.156	0.044
florasulam	0.05	S	75	0.1	0.044
florasulam	0.05	S	75	0.21	0.071
florasulam	0.158	S	85	0.078	0.038
florasulam	0.158	S	85	0.063	0.029
florasulam	0.158	S	85	0.067	0.032
florasulam	0.158	S	85	0.068	0.027
florasulam	0.5	S	90	0.048	0.019
florasulam	0.5	S	90	0.052	0.025
florasulam	0.5	S	90	0.043	0.025
florasulam	0.5	S	90	0.111	0.034
florasulam	1.58	S	95	0.047	0.025
florasulam	1.58	S	95	0.069	0.023
florasulam	1.58	S	95	0.039	0.023
florasulam	1.58	S	95	0.045	0.031
florasulam	5	S	95	0.054	0.026
florasulam	5	S	95	0.042	0.022
florasulam	5	S	95	0.054	0.036
florasulam	5	S	95	0.046	0.023
florasulam	15.8	S	95	0.048	0.026
florasulam	15.8	S	95	0.052	0.028
florasulam	15.8	S	95	0.049	0.026
florasulam	15.8	S	95	0.038	0.026
florasulam	50	S	95	0.036	0.024
florasulam	50	S	95	0.052	0.034
florasulam	50	S	95	0.049	0.026
florasulam	50	S	95	0.041	0.026
florasulam	158	S	95	0.041	0.029
florasulam	158	S	98	0.038	0.025
florasulam	158	S	98	0.045	0.027
florasulam	158	S	98	0.039	0.025

#### Data for analysis - tribenuron

herbicide	dose	biotype	response	fresh matter	dry matter
tribenuron	0.001	R	0	8.433	1.277
tribenuron	0.001	R	0	6.537	1
tribenuron	0.001	R	0	5.586	0.783

tribenuron	0.001	R	0	4.576	0.737
tribenuron	0.03555	R	0	NA	NA
tribenuron	0.03555	R	0	NA	NA
tribenuron	0.03555	R	0	NA	NA
tribenuron	0.03555	R	0	NA	NA
tribenuron	0.1125	R	0	3.237	0.447
tribenuron	0.1125	R	0	1.865	0.296
tribenuron	0.1125	R	0	2.753	0.414
tribenuron	0.1125	R	0	2.045	0.304
tribenuron	0.3555	R	10	4.322	0.557
tribenuron	0.3555	R	10	3.275	0.487
tribenuron	0.3555	R	15	3.251	0.538
tribenuron	0.3555	R	10	3.507	0.539
tribenuron	1.125	R	10	3.49	0.454
tribenuron	1.125	R	10	4.678	0.737
tribenuron	1.125	R	15	0.925	0.147
tribenuron	1.125	R	10	2.392	0.438
tribenuron	3.555	R	15	3.377	0.442
tribenuron	3.555	R	15	2.642	0.391
tribenuron	3.555	R	20	2.11	0.298
tribenuron	3.555	R	20	2.654	0.429
tribenuron	11.25	R	20	3.717	0.417
tribenuron	11.25	R	20	2.797	0.292
tribenuron	11.25	R	25	2.376	0.285
tribenuron	11.25	R	20	2.795	0.409
tribenuron	35.55	R	25	2.388	0.317
tribenuron	35.55	R	30	2.326	0.336
tribenuron	35.55	R	25	1.945	0.31
tribenuron	35.55	R	25	2.95	0.507
tribenuron	112.5	R	30	2.403	0.272
tribenuron	112.5	R	35	2.111	0.272
tribenuron	112.5	R	30	3.891	0.511
tribenuron	112.5	R	30	4.157	0.409
tribenuron	355.5	R	60	0.68	0.12
tribenuron	355.5	R	60	1.162	0.126
tribenuron	355.5	R	55	0.856	0.15
tribenuron	355.5	R	50	1.165	0.166
tribenuron	1125	R	60	0.277	0.076
tribenuron	1125	R	75	0.325	0.077
tribenuron	1125	R	85	0.101	0.05
tribenuron	1125	R	85	0.121	0.057
tribenuron	0.001	S	0	1.301	0.229
tribenuron	0.001	S	0	3.631	0.491
tribenuron	0.001	S	0	3.627	0.567
tribenuron	0.001	S	0	4.632	0.746
tribenuron	0.03555	S	15	1.988	0.373



tribenuron	0.03555	S	15	2.405	0.364
tribenuron	0.03555	S	20	2.285	0.353
tribenuron	0.03555	S	20	2.697	0.4
tribenuron	0.1125	S	85	0.154	0.049
tribenuron	0.1125	S	80	0.082	0.041
tribenuron	0.1125	S	85	0.143	0.05
tribenuron	0.1125	S	85	0.114	0.039
tribenuron	0.3555	S	90	0.0112	0.046
tribenuron	0.3555	S	88	0.0119	0.044
tribenuron	0.3555	S	90	0.062	0.031
tribenuron	0.3555	S	90	0.077	0.027
tribenuron	1.125	S	92	0.09	0.036
tribenuron	1.125	S	92	0.078	0.03
tribenuron	1.125	S	92	0.044	0.029
tribenuron	1.125	S	92	0.096	0.037
tribenuron	3.555	S	95	0.068	0.031
tribenuron	3.555	S	95	0.07	0.033
tribenuron	3.555	S	95	0.04	0.029
tribenuron	3.555	S	95	0.059	0.037
tribenuron	11.25	S	95	0.065	0.037
tribenuron	11.25	S	95	0.074	0.043
tribenuron	11.25	S	95	0.04	0.038
tribenuron	11.25	S	95	0.043	0.038
tribenuron	35.55	S	95	0.043	0.032
tribenuron	35.55	S	95	0.046	0.033
tribenuron	35.55	S	95	0.049	0.033
tribenuron	35.55	S	95	0.057	0.036
tribenuron	112.5	S	95	0.067	0.031
tribenuron	112.5	S	95	0.057	0.035
tribenuron	112.5	S	95	0.057	0.027
tribenuron	112.5	S	95	0.079	0.035
tribenuron	355.5	S	100	0.041	0.032
tribenuron	355.5	S	98	0.052	0.024
tribenuron	355.5	S	98	0.045	0.025
tribenuron	355.5	S	98	0.057	0.031
tribenuron	1125	S	100	NA	NA
tribenuron	1125	S	100	NA	NA
tribenuron	1125	S	100	NA	NA
tribenuron	1125	S	100	NA	NA

**Název:** Statistical modelling of dose response curves - with a special focus on herbicide resistance

**Autor:** Kateřina Hamouzová

**Vydavatel:** Česká zemědělská univerzita v Praze

**Adresa vydavatele:** Česká zemědělská univerzita v Praze, Kamýcká 129, Praha – Suchdol, 165 00

**Tiskárna:** elektronická skripta

**Lektoroval:** prof. Husrev Mennan

**Pořadí vydání:** 1.

**Rok vydání:** 2024

**ISBN:** 978-80-213-3371-0

**Vydala Česká zemědělská univerzita ve svém nakladatelství**