TEXTE
151/2023

Final report

Application of nonlinear hierarchical models to the kinetic evaluation of chemical degradation data

Guidance for the use of an R markdown template file

by:

Johannes Ranke

Wissenschaftlicher Berater, Grenzach-Wyhlen

publisher:

German Environment Agency

Umwelt Bundesamt

TEXTE 151/2023

Project No. 173340
Report No. (UBA-FB) FB001177/ENG

Final report

Application of nonlinear hierarchical models to the kinetic evaluation of chemical degradation data

Guidance for the use of an R markdown template file

by

Johannes Ranke
Wissenschaftlicher Berater, Grenzach-Wyhlen

On behalf of the German Environment Agency

Imprint

Publisher
Umweltbundesamt
Wörlitzer Platz 1
06844 Dessau-Roßlau
Tel: +49 340-2103-0
Fax: +49 340-2103-2285
buergerservice@uba.de
Internet: www.umweltbundesamt.de

Report performed by:
Johannes Ranke, Wissenschaftlicher Berater
Kronacher Str. 12
79639 Grenzach-Wyhlen

Report completed in:
February 2023

Edited by:
IV 1.3-2 Pflanzenschutzmittel – Abbau und Exposition
Janina Wöltjen

IV 2.5 Spurenanalytik, Fließ- und Stillgewässersimulation
Stefan Meinecke

Publication as pdf:
http://www.umweltbundesamt.de/publikationen

ISSN 1862-4804

Dessau-Roßlau, October 2023

The responsibility for the content of this publication lies with the author.

Abstract: Application of nonlinear hierarchical models to the kinetic evaluation of chemical degradation data

The currently used procedures for the kinetic evaluation of chemical degradation data are based on the separate application of different nonlinear regression models to each of the available datasets. In many cases, some of the degradation parameters cannot reliably be quantified in at least some of the datasets. Current guidance suggests to use default values in such cases, which were more or less arbitrarily chosen. Also, different kinetic models provide the best fit in different datasets, so mean parameters can only be calculated using workarounds with a weak scientific basis.

Both of these problems can be avoided by the use of hierarchical nonlinear models, where parameter distributions are fitted to the complete data set. In this report, a short introduction to this type of model is presented. Furthermore, it is described how to use the R markdown template recently added to the mkin R package, in combination with the newly developed spreadsheet file for entering data, making it easier to apply this method to new data.

To implement hierarchical kinetic modelling in regulatory practice, a guidance document would need to be developed, providing recommendations how the results from hierarchical degradation kinetics should be used in the various regulatory areas where kinetic degradation endpoints are relied upon.

Kurzbeschreibung: Verwendung von nicht-linearen hierarchischen Modellen zur Ableitung von kinetischen Modellparametern aus Abbaustudien

Derzeit werden chemische Abbaudaten ausgewertet, indem verschiedene nichtlineare Regressionsmodelle einzeln auf die verfügbaren Datensätze angewandt werden. In vielen Fällen können dabei einige der Abbauparameter nicht für alle Datensätze verlässlich bestimmt werden. Die aktuell gültigen regulatorischen Leitlinien empfehlen in solchen Fällen die Verwendung von mehr oder weniger willkürlich gewählten Standardwerten für diese Parameter. Des Weiteren ergeben oft unterschiedliche Modelle die beste Anpassung in den verschiedenen Datensätzen, so dass mittlere Modellparameter mit Hilfe von Behelfslösungen mit schwacher wissenschaftlicher Grundlage bestimmt werden müssen.

Beide Probleme können vermieden werden, wenn hierarchische nichtlineare Modelle verwendet werden, bei denen Parameterverteilungen an die Gesamtheit der Daten angepasst werden. In diesem Bericht wird eine kurze Einführung in diesen Modelltyp gegeben. Weiterhin wird die Verwendung einer R markdown Vorlage und einer Tabellenkalkulationsdatei für die Eingabe von Daten beschrieben. Beide Dateien wurden kürzlich in das R-Paket mkin integriert und erleichtern damit die Anwendung dieser Methode auf neue Daten.

Um hierarchische kinetische Modelle in der regulatorischen Auswertung von Abbaudaten zu etablieren, müsste ein Leitfaden erarbeitet werden, in dem erläutert wird, wie die Ergebnisse der hierarchischen Abbaukinetiken in den verschiedenen regulatorischen Anwendungsbereichen verwendet werden sollten.

Table of contents

List of figures

List of tables

Acknowledgements

In addition to the financial support of the German Environment Agency in this and previous related projects, the fruitful discussions with staff members Janina Wöltjen and Stefan Meinecke, responsible for the supervision of this project are gratefully acknowledged, their contributions were crucial to the success of the project. Further valuable comments and feedback were received from Thomas Gräff and Jara Contardo. Last but not least, the author is highly indebted to Emmanuelle Comets from the French National Institute of Health (INRA) because of the invaluable exchange on nonlinear hierarchical models and their implementations.

List of abbreviations

AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
DFOP	Dual First-Order in Parallel Model
DLL	Dynamically Loadable Library
FOCUS	Forum for the Coordination of Pesticide Fate Models and their Use
FOMC	First-Order Multi-Compartment Model
HS	Hockey Stick Model
NLHM	Nonlinear Hierarchical Models
SFO	Simple First-Order Model
SFORB	Single First-Order Reversible Binding Model
UBA	German Environment Agency (Umweltbundesamt)

Summary

The environmental risk assessment of chemical substances and their transformation products depends on the calculation of predicted environmental concentrations. These calculations are based on degradation endpoints such as half-lives and formation fractions which are derived from experiments carried out in laboratories or in the field by data evaluation procedures often summarised as kinetic evaluations.

The currently used procedures are based on the separate application of different nonlinear regression models to each of the available datasets. In many cases, some of the degradation parameters cannot reliably be quantified in at least some of the datasets, and current guidance suggests to use default values in such cases, which were more or less arbitrarily chosen. Also, different kinetic models provide the best fit in different datasets, so mean parameters can only be calculated using workarounds with a weak scientific basis.

Both of these problems can be avoided by the use of hierarchical nonlinear models, where parameter distributions are fitted to the complete data set. The purpose of the project that resulted in this report was to facilitate the application of nonlinear hierarchical models to chemical degradation data for experts interested in testing the approach. To reach this goal, the facilities to set up such models with the R language and environment for statistical computing (R core 2022) already available in the mkin extension package (Ranke et al. 2022) were extended and improved, and a template file for performing and documenting the corresponding evaluations was developed.

In this report, a short introduction to this type of models is presented. Furthermore, it is described how to use the R markdown template recently added to the mkin package, in combination with the newly developed spreadsheet file for entering data, making it much easier to apply this method to new data. The template covers, in the order of increasing complexity, the evaluation of parent only data without consideration of covariates, the evaluation of parent only data including models for the influence of covariates like soil pH on certain model parameters, and finally the evaluation of data on a parent compound and its transformation products (pathway fits), potentially also with considering covariates.

To set up the hierarchical model fits with suitable starting values, classical separate evaluations are always performed as a first step. In order to save time, these fits can be performed in parallel, making use of the multi-core capabilities of modern computing hardware, and pathway models can be precompiled if a compiler is installed. Under some circumstances, different versions of the succeeding hierarchical model fits can also be fitted in parallel to save time.

For some of the kinetic parameters occasionally only average parameters can be found, while their variability over the assumed population of similar systems cannot be quantified. Therefore, the template also covers the option of model simplification up to the point where the model is fully identifiable. Model comparisons used for model selection are based on the Akaike Information Criterion and the Bayesian Information Criterion.

To implement hierarchical kinetic modelling in regulatory practice, a guidance document would need to be developed, providing recommendations how the results from hierarchical degradation kinetics should be used in the various regulatory areas where kinetic degradation endpoints are relied upon.

Zusammenfassung

Ein wesentliches Element der Umweltrisikoanalyse von Chemikalien ist die Berechnung von abgeschätzten Konzentrationen in der Umwelt. Für diese Berechnungen werden Halbwertszeiten und "formation fractions" (Anteile der Bildung bestimmter Transformationsprodukte) benötigt, welche durch die kinetische Auswertung von Labor- und Feldexperimenten zum Abbau gewonnen werden.

Derzeit werden die entsprechenden Daten ausgewertet, indem verschiedene nichtlineare Regressionsmodelle einzeln an die verfügbaren Datensätze angepasst werden. In vielen Fällen können dabei einige der Abbauparameter nicht für alle Datensätze verlässlich bestimmt werden. Die aktuell gültigen regulatorischen Leitlinien empfehlen in solchen Fällen die Verwendung von mehr oder weniger willkürlich gewählten Standardwerten für diese Parameter. Des Weiteren sind die Modelle die beste Anpassung ergeben, in den verschiedenen Datensätzen oft unterschiedlich, so dass mittlere Modellparameter nur mit Hilfe von Behelfslösungen mit schwacher wissenschaftlicher Grundlage bestimmt werden können.

Beide Probleme können vermieden werden, wenn hierarchische nichtlineare Modelle verwendet werden, bei denen Parameterverteilungen an die Gesamtheit der Daten angepasst werden. Das Ziel des Projektes, aus dem dieser Bericht hervorgegangen ist, war es, die Auswertung von Abbaudaten mit Hilfe von nichtlinearen hierarchischen Modellen für interessierte Experten*Expertinnen zu erleichtern. Um dies zu ermöglichen, wurden die bereits im R-Paket mkin (Ranke et al. 2022) vorhandenen Elemente zur Anwendung solcher Modelle erweitert und verbessert. Auch wurde eine R markdown Vorlagendatei zur Durchführung und Dokumentation der entsprechenden Auswertungen erstellt.

In diesem Bericht wird eine kurze Einführung in diesen Modelltyp gegeben. Weiterhin wird die Verwendung der R markdown Vorlage sowie der ebenfalls neu entwickelten Tabellenkalkulationsdatei für die strukturierte Eingabe von Daten beschrieben, durch welche die Anwendung der Methode auf neue Daten stark erleichtert wird.

Die Vorlagendatei deckt, in der Reihenfolge zunehmender Komplexität, die Auswertung der Daten zur Ausgangssubstanz ohne Berücksichtigung von Kovariablen, die Auswertung derselben Daten unter Einbezug von Kovariablen wie z.B. des pH-Wertes der Böden, und schließlich die Auswertung der Daten von Ausgangssubstanz und Transformationsprodukten, unter optionalem Einbezug von Kovariablen ab.

Um für die Anpassung der hierarchischen Modelle geeignete Startparameter zu finden, werden jeweils in einem ersten Schritt die konventionellen separaten Auswertungen durchgeführt. Diese Auswertungen können unter Ausnutzung der Fähigkeiten aktueller Hardware parallelisiert werden. Modelle, die auch die Transformationsprodukte beschreiben, können bei Verfügbarkeit eines Compilers in Binärcode übersetzt werden. Unter bestimmten Umständen können auch verschiedene Versionen der hierarchischen Modelle zeitsparend parallel angepasst werden.

Für manche kinetische Parameter können in einigen Fällen nur Durchschnittswerte für die Gesamtheit der experimentellen Einheiten bestimmt werden, während ihre Variabilität nicht verlässlich quantifiziert werden kann. Daher enthält die Vorlagendatei auch Schritte zur Vereinfachung der Parameter-Verteilungsmodelle bis zu dem Punkt, an dem das Modell vollständig identifizierbar ist. Modellvergleiche zur Auswahl der besten Modelle basieren auf dem "Akaike Information Criterion" (AIC) und dem Bayesschen Informationskriterium (BIC).

Um hierarchische kinetische Modelle in der Praxis der regulatorischen Auswertung von Abbaudaten zu etablieren, müsste ein Leitfaden erstellt werden, in dem erläutert wird, wie die

Ergebnisse der hierarchischen Abbaukinetiken in den verschiedenen regulatorischen Anwendungsbereichen verwendet werden sollten.

1 Introduction

The environmental risk assessment of chemical substances and their transformation products depends on the calculation of predicted environmental concentrations. These calculations are based on degradation endpoints such as half-lives and formation fractions which are derived from experiments carried out in laboratories or in the field by data evaluation procedures often summarised as kinetic evaluations.

Kinetic evaluations of degradation data have been standardised by the Forum for the Coordination of Pesticide Fate Models and their Use (FOCUS) in a guidance document, in the following referred to as FOCUS kinetics guidance, which is currently used for regulatory environmental risk assessments not only of pesticides but also of biocides and other chemical substances in the European Union (FOCUS 2006, 2014). To reduce the need for expert judgement in kinetic evaluations, a revision of the FOCUS kinetics guidance is currently being reviewed by member states of the European Union. The proposed improvements of the FOCUS kinetics guidance provide more precise criteria for selection of the most suitable kinetic model for each degradation dataset. However, the endpoints are derived separately for all available datasets and have to be aggregated in a second step. If different models were selected for different datasets, there is no obvious method for deriving mean degradation parameters that are representative for all datasets and the guidance that was developed to circumvent this problem is rather complex and may even appear arbitrary.

As an alternative to such separate evaluations of the available datasets and the subsequent aggregation of the obtained endpoints, it has been proposed to evaluate all datasets in one step using nonlinear hierarchical models (NLHM) also known as nonlinear mixed-effects models (Ranke et al. 2021, Ranke and Wöltjen 2022). This proposal originated in a review of model selection criteria commissioned by the German Environment Agency (UBA) with the purpose of developing objective criteria for evaluating the visual fit in the kinetic evaluation of degradation data (UBA Project Number 120667). The feasibility of that approach was then shown in a successor project (UBA Project Number 145839) which also lead to the publications cited above.

The purpose of the current project was to facilitate the application of nonlinear hierarchical models to chemical degradation data for experts interested in testing the approach. To reach this goal, the facilities to set up such models with the R language and environment for statistical computing (R core 2022) already available in the mkin extension package (Ranke et al. 2022) were improved, and a template file for performing and documenting such evaluations was developed, together with a spreadsheet for entering the data.

This report provides a very general introduction to nonlinear hierarchical models from a practical point of view, as well as step by step instructions to apply them to chemical degradation data using the template file mentioned above.

2 Nonlinear hierarchical models for degradation data

The nonlinear hierarchical models discussed here are composed of a degradation model, an error model, a parameter distribution model and an optional model for the influence of covariates like soil pH. The most relevant aspects of these models are described in this section.

Note that the FOCUS kinetics guidance recommends the separate evaluation of each degradation dataset by nonlinear regression, which is only based on a degradation model and a more or less explicit error model. Therefore, the use of parameter distribution models and covariate models is not covered in the FOCUS kinetics guidance.

2.1 Degradation models

The degradation model is an idealised mathematical description of the time course the residues of a parent compound and potentially of its transformation products take after dosing the system of interest with the parent compound. For the parent compound, the relevant degradation models as defined in the FOCUS kinetics guidance are single first-order (SFO), first-order multi-compartment (FOMC), dual first-order in parallel (DFOP), single first-order reversible binding (SFORB) and hockey stick (HS). Please refer to the FOCUS kinetics guidance for their definitions, the parameters used and a discussion of their characteristics. Note that both the DFOP and the SFORB model are biexponential degradation models that can describe exactly the same residue decline curves, but with different parameters.

Many parameters of these degradation models (called natural parameters in the tables below) cannot take on negative values and therefore have a lower bound of zero. Others have an upper bound of one. In mkin, degradation models are re-parameterised in a way that the parameters that are actually optimised do not have lower or upper bounds. The names of the degradation parameters as used in mkin are listed in Table 1, together with their bounds and their transformed counterparts. These parameter names are important for the interpretation of the output that mkin produces.

If no bounds are assumed for a parameter, as in the case of the initial concentration of the parent compound (parameter parent_0), the transformed parameter is the same as the natural parameter. If the lower bound of a parameter is zero, as in the case of all rate constants, the name of the transformed parameter is prepended by the string “log_” to indicate that it is the log-transformed version of the natural degradation parameter. If the lower bound of a parameter is zero and the upper bound is one, the so-called logit transformation is used to obtain the transformed parameter. As this can be achieved by the function “qlogis” in R, the name of the transformed parameter is obtained by appending “_qlogis” to the natural parameter name.

Table 1: Parent degradation parameters in mkin

Model	Natural parameter	Lower bound	Upper bound	Transformed parameter
SFO	parent_0	-	-	parent_0
	k_parent	0	-	log_k_parent
FOMC	parent_0	-	-	parent_0
	alpha	0	-	log_alpha
	beta	0	-	log_beta

Model	Natural parameter	Lower bound	Upper bound	Transformed parameter
DFOP	parent_0	-	-	parent_0
	k1	0	-	log_k1
	k2	0	-	log_k2
	g	0	1	g_qlogis
SFORB	parent_0	-	-	parent_0
	k_parent_free	0	-	log_k_parent_free
	k_parent_free_bound	0	-	log_k_parent_free_bound
	k_parent_bound_free	0	-	log_k_parent_bound_free
HS	parent_0	-	-	parent_0
	k1	0	-	log_k1
	k2	0	-	log_k2
	tb	0	-	log_tb

In the parameter names, “parent” will be replaced by the acronym used for the parent compound in the data

When transformation products (also known as metabolites) are formed from the parent compound, the fraction of the degraded parent compound that reacts to a specific transformation product is called its formation fraction. If a compound only has one transformation product, the corresponding formation fraction is transformed using the logit transformation. However, if a compound has more than one transformation product, there is an upper limit of one to the sum of the formation fractions, and the upper limit of each fraction depends on the other formation fractions. In mkin, such sets of formation fractions are transformed to a set of unbounded transformed parameters using the isometric logratio transformation (Ranke and Lehmann 2012). The degradation of transformation products is most often described by SFO. An overview of the degradation parameters used for transformation products is given in Table 2.

Table 2: Transformation related degradation parameters in mkin

Use case	Natural parameter	Lower bound	Upper bound	Transformed parameter
Single transformation fraction	f_parent_to_m1	0	1	f_parent_qlogis
Set of two or more transformation fractions	f_parent_to_m1, f_parent_to_m2, …	0	Sum of transformation fractions = 1	f_parent_ilr_1, f_parent_ilr_2, …
Transformation product with SFO degradation	k_m1	0	-	log_k_m1

In the parameter names, “parent” will be replaced by the acronym of the compound that is being transformed, and “m1” and “m2” will be replaced by the acronyms of the products of the transformation.

2.2 Error models

The three error models available for separate evaluations in mkin are “constant variance”, “variance by variable” and “two-component error” (Ranke and Meinecke 2019). In the FOCUS kinetics guidance, only constant variance and variance by variable are described. As the variance by variable error model has been proposed to be fitted by the iteratively reweighted least squares (IRLS) algorithm in kinetic evaluations of degradation data (Gao et al. 2011), it can be selected as “IRLS” option in the software packages CAKE and KinGUII that are commonly used for kinetic evaluations in the regulatory context.

The two-component error model has been developed in the area of analytical chemistry. In pharmacokinetics, it is also known as “combined” error model, because it combines an additive error (a constant variance component) and a proportional error (an error that increases with the magnitude of the observed value). In many cases, the absolute error is smaller for transformation products than for the parent compound. At the same time, the mean residue level of the transformation products is also smaller. Therefore, the different variances obtained for parent compound and transformation products found by variance by variable with the IRLS algorithm can often also be explained by the two-component error model, because it incorporates an error term that is proportional to the observed value. Depending on the dataset, the two-component error model is often even superior to the variance by variable error model.

Currently, the recommended backend used in mkin for fitting hierarchical models is the R package saemix (Comets et al. 2017) which only supports constant variance and two-component error as error models. The parameter names used in the output of mkin for separate evaluations and the names used by mkin for saemix model evaluations are shown in Table 3.

Table 3: Error model parameter names used in mkin and saemix

Error model	Parameter description	Name in mkin	Name in saemix
Constant variance	Standard deviation of a constant error	sigma	a.1
Two-component error	Standard deviation of a constant, additive error component	sigma_low	a.1
	Relative standard deviation, describing a proportional error component	rsd_high	b.1

2.3 Parameter distribution models

As mentioned above, no explicit assumptions about the distribution of kinetic parameters across the statistical population of interest (for example the population of agricultural soils in the European Union) are necessary for the separate evaluation of each dataset as recommended in the FOCUS kinetics guidance. However, the guidance issued by the European Food Safety Authority (EFSA) on the evaluation of field dissipation data of pesticides has established a statistical method for testing the significance of the difference between degradation half-lives obtained in the laboratory and those obtained from field data (EFSA 2014). In this guidance, the explicit assumption is used that those half-lives follow a lognormal distribution, i.e. that they follow a normal distribution when they are log-transformed. This is equivalent to assuming normal distribution for log-transformed degradation rate constants.

No explicit assumptions about parameter distributions are usually made for other degradation parameters in the context of kinetic evaluations for regulatory environmental risk assessments.

In the area of pharmacokinetics, but also in other areas where so-called repeated measures data are evaluated with nonlinear mixed-effects models, the assumption of a multivariate normal distribution is generally used for suitably transformed parameters (Pinheiro and Bates 2000, Lavielle 2015). The parameter transformations used per default in mkin are compatible with this approach, because the transformed parameters do not have any lower or upper bounds as shown in Table 1 and Table 2.

The parameters of this parameter distribution model are a vector of mean values μ and a variance-covariance matrix Ω. A more elaborate formal description has been given previously (Ranke et al. 2021).

In the field of degradation kinetics, the number of experimental units for which degradation data are available is usually small. Therefore, only a limited number of parameters in Ω can be estimated. To limit the number of parameters describing the population variance, the default parameter model used for hierarchical kinetic models in mkin assumes that the variances of the degradation parameters are uncorrelated, which means that the off-diagonal entries in Ω are zero. This parameter model can be adapted using a certain syntax defined in the saemix package. In many cases it will be necessary to further reduce the parameter distribution model to restrict it to the parameter variances that can reliably be quantified. Such model reductions are possible in saemix, and can easily be performed when preparing saemix fits with mkin, as will be described below.

In the summary output for such fits as produced by mkin, estimates and confidence intervals for the square roots of the diagonal entries of Ω are listed, i.e. the variances are expressed as standard deviations. To give an example, if a “random effect” (variability across the population) is assumed for the slower rate constant k2 of the DFOP model the corresponding parameter describing this random effect will be called “SD.log_k2”.

If the user has specified a correlation between certain random effects, i.e. if there are non-zero off-diagonal elements in Ω, the corresponding correlations on a scale between zero and one will be listed as “Corr.parameter_1.parameter_2”.

All of the variance parameters making up the variance-covariance matrix Ω have a lower bound of zero. Therefore, a confidence interval for any one of these parameters including zero indicates that the value cannot reliably be quantified. In the terminology used in the documentation of mkin, such parameters are “ill-defined”, and it is suggested to repeat the model fit with the respective parameter set to zero and to compare the results.

2.4 Covariate models

In some cases, there are reasons to assume that one or more degradation parameters depend on the value of a covariate. A typical case would be that a degradation rate constant depends on the soil pH. When setting up a saemix model fit with mkin as described below, such covariate dependencies can be specified as linear models of the transformed degradation parameters, with the covariate as explanatory variable. In the case of a rate constant correlation with pH, the parameter (for example `log_k_parent`) will be estimated for a covariate value of zero (pH 0 in the case of pH dependence), and there will be an additional parameter in the output, describing the slope of the transformed parameter with respect to the covariate. The name of the slope parameter will start with “beta_”, followed by the name of the covariate, followed by the transformed degradation parameter in parentheses. In our example, this slope parameter would be listed as `beta_pH(log_k_parent)`.

Because it is unlikely that a correlation with a covariate can be quantified for degradation parameters that are ill-defined, it is recommended that such correlations are only investigated for degradation parameters that are not found to be ill-defined in corresponding fits without covariate models.

3 Use instructions for the template file

To facilitate the application of hierarchical kinetic degradation models to chemical degradation data, an R markdown template and an example spreadsheet have been incorporated into the mkin package.

The R markdown format allows to write reports in the easy to learn markdown language while incorporating the R code used for data analysis. The R markdown files can be edited using any plain text editor. However, it is recommended to use an integrated development environment with explicit support for R markdown documents. In this report, it is shown how to use the template with the RStudio software on a Microsoft Windows operating system, in the hope that users of other R markdown development environments or operating systems will find it easy to adapt the instructions to their working environment.

In the following subsections, any R code and any identifiers for R functions or other objects defined in R are formatted using the fixed space Courier New font. For example, a variable containing the numeric value 2 could be defined by the R code `some_variable <- 2`, and in the following, the variable name `some_variable` would be formatted using this font to indicate that it refers to an R object.

When editing R markdown documents in the RStudio software a number of keyboard shortcuts can be used. Whenever reference to such keyboard shortcuts is made, the names of the respective keys are enclosed in angle brackets. As an example, to indicate that the "Control" key and the "Enter" key should be pressed at the same time, the expression <Ctrl><Enter> will be used.

The R code given in the template is not explained in detail. The user merely has to understand a small part of it in order to make the changes necessary to perform an analysis. However, users familiar with the R language will be less likely to make errors in adapting the R code in the template to their needs and interests. It is also recommended to consult the documentation of the mkin package which is installed with the package and also available online[1].

3.1 Software requirements

The following software is required for a successful use of the R markdown template:

- The R software in version 4.1 or greater[2]
- A development environment for R markdown files[3]
- The R extension packages "knitr" and "readxl"[4]
- The mkin extension package in version 1.2.2 or greater[5]

[1] Online documentation of the released mkin version is available at https://pkgdown.jrwb.de/mkin. The development version is documented at https://pkgdown.jrwb.de/mkin/dev

[2] The R software is freely available from https://cran.r-project.org.

[3] A free version of RStudio is available from https://posit.co/products/open-source/rstudio/

[4] These can be installed from within RStudio using menu item "Tools", submenu item "Install packages". Alternatively, they can be installed from the command line using the R command `install.packages(c("knitr", "readxl"))`.

[5] The mkin extension package can be installed as described above. To get the latest development version, it can also be installed using the "remotes" package with the command `remotes::install_github("jranke/mkin")`. Of course, the "remotes" package has to be installed for this to work, for example using the command `install.packages("remotes")`.

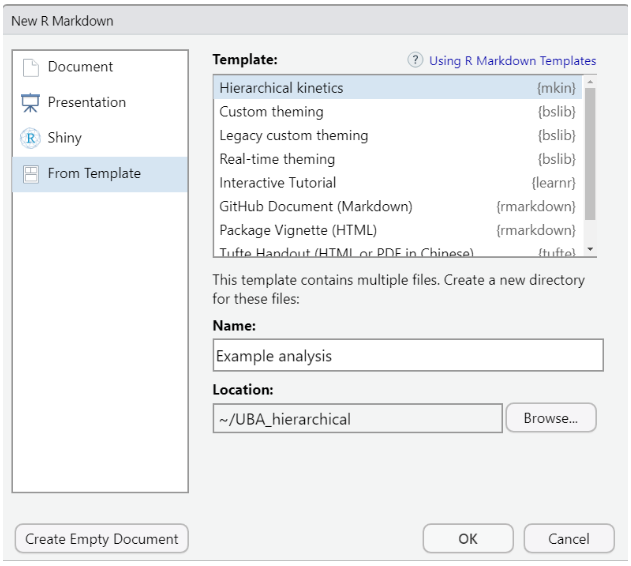
- A software package capable of editing spreadsheet files in Microsoft Excel file format[6]
- An installation of a TeX layout software distribution, such as TinyTex, MikTex or TexLive, including the TeX packages “float” and “listing”[7]

In case the degradation data to be evaluated include residue data on transformation products, it is strongly recommended to make a compiler for the C programming language available to the R software in order to benefit from faster execution times. Users of the Windows operating system can achieve this by installing the software compilation “Rtools”[8]. Users of all supported operating systems can check the availability of a compiler by installing the “pkgbuild” extension package and issuing the R command `pkgbuild::has_compiler()`. If this returns `TRUE`, pathway models set up using the mkin package will be compiled for faster execution.

3.2 How to set up a new analysis

If mkin is installed in version greater than 1.2.2, a new kinetic analysis using hierarchical models can be set up from within RStudio using the menu item “File”, submenu “New File” and selecting “R markdown”. In the window with window title “New R Markdown” that appears (Figure 1), the item “From Template” has to be selected in the selection area to the left. When this selection has been made, a list of Templates should appear to the right, including a template called “Hierarchical kinetics” originating from the mkin package. After selecting this template, a name for the analysis has to be entered, and the location for the analysis should be specified using the “Browse…” button. The name will be used for a newly created folder in the selected location and for the filename of the R markdown file that will be created from the template in the respective new folder.

Figure 1: Screenshot of the Rstudio window for template selection



Template:	
Hierarchical kinetics	{mkin}
Custom theming	{bslib}
Legacy custom theming	{bslib}
Real-time theming	{bslib}
Interactive Tutorial	{learnr}
GitHub Document (Markdown)	{rmarkdown}
Package Vignette (HTML)	{rmarkdown}
Tufte Handout (HTML or PDF in Chinese)	{tufte}

This template contains multiple files. Create a new directory for these files:

Name: Example analysis

Location: ~/UBA_hierarchical

Source: Own contribution, Johannes Ranke

[6] Tested with Microsoft Excel and LibreOffice Calc

[7] When using the Windows operating system, the easiest way to install a suitable TeX distribution is to install the R extension package “tinytex”, for example using the command `install.packages("tinytex")`. Once this is installed, the actual TeX distribution has to be installed using the R command `tinytex::install_tinytex()`. Finally, two TeX packages have to be installed using the command `tinytex::tlmgr_install(c("float", "listing"))`.

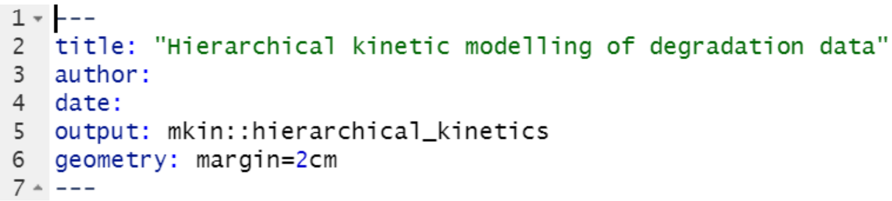
[8] Freely available from https://cran.r-project.org/bin/windows/Rtools/

Note that on some systems, selecting a folder on a so-called network drive will prevent successful processing of the markdown document. On such systems, a folder on a local disk must be selected.

3.2.1 Editing the header

The R markdown template file starts with a section written in a data serialization language called YAML. The section starts and ends with three dashes on a separate line. Within the section, there are three lines starting with the words title, author and date, followed by a colon. On the title line, a default title for the analysis is already given in quotation marks. This title should be adapted to reflect the desired content of the analysis. On the author and date lines, the corresponding information should be added after the colon. The information given after each colon can optionally be surrounded by quotation marks. This is necessary if the user would like to include a colon in the title.

Figure 2: Screenshot of the header of the R markdown template



```
---
title: "Hierarchical kinetic modelling of degradation data"
author:
date:
output: mkin::hierarchical_kinetics
geometry: margin=2cm
---
```

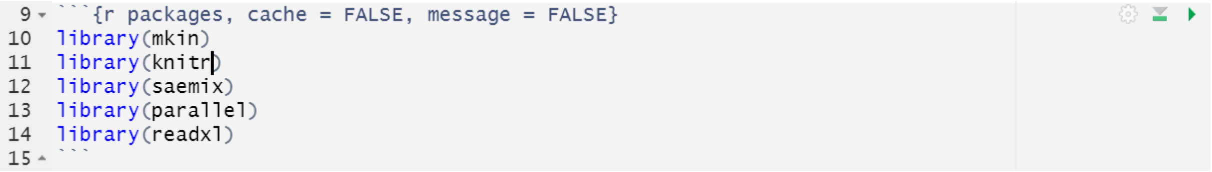
Source: Own contribution, Johannes Ranke

The lines starting with “output” and “geometry” should be left unaltered.

3.2.2 Code chunks and chunk options

After the YAML header, the first block of R code follows. Every block of R code in an R markdown document, commonly called “code chunk”, is enclosed by lines starting with three backticks. At the start of an R code chunk, the three backticks are followed by an opening brace, the letter “r” and an optional name of the code chunk. After this, further options can be specified to modify the way the R code and its output are treated in the processing of the document.

Figure 3: Example code chunk with R code loading extension packages



~~~
```{r packages, cache = FALSE, message = FALSE}
library(mkin)
library(knitr)
library(saemix)
library(parallel)
library(readxl)
```
~~~

Source: Own contribution, Johannes Ranke

The first code chunk in the template is named “packages” (Figure 3). The chunk options specify that the output of the R code in the chunk does not need to be cached, and that messages emitted during execution of the R code should be suppressed. The code loads a number of R packages, so that the functions defined in these packages can be used after the code in the chunk has been executed. This code chunk can remain unaltered.

In order to execute the R code in a code chunk, the user can place the cursor on one of the lines containing R code, and press the key combination <Ctrl><Shift><Enter>. Alternatively, the user can press the green arrow that RStudio displays in the top right corner of each code chunk. If code from an R code chunk is executed, it will appear in the R console window located below the window containing the R markdown document.

Running the code in the various code chunks in this way is only necessary if the analysis is to be performed step by step. As the R code is executed in the R console, the user can interact with the objects created in the process. Also, any objects created will be visible in the Rstudio window to the top right in the tabulator called “Environment”.

At any time, when the user is confident that all R code has been adapted correctly and a report should be generated, processing of the complete document in such a separate R process can be started with the key combination <Ctrl><Shift><K>. Alternatively, the “Knit” button above the R markdown editor window can be pressed.

3.2.3 Defining a cluster for parallel computing

The next code chunk in the template is called “n_cores”, because its purpose is to define how many computing cores should be used in parallel computations. Per default, the R variable n_cores is defined in this code chunk as the result of a call to the function `detectCores()`. This will be the number of computing cores available to the OS running the R software. If some undesired delays of other programs are experienced caused by the processing of the R markdown document, one computing core can be freed by modifying the corresponding line to read `n_cores <- detectCores - 1`.

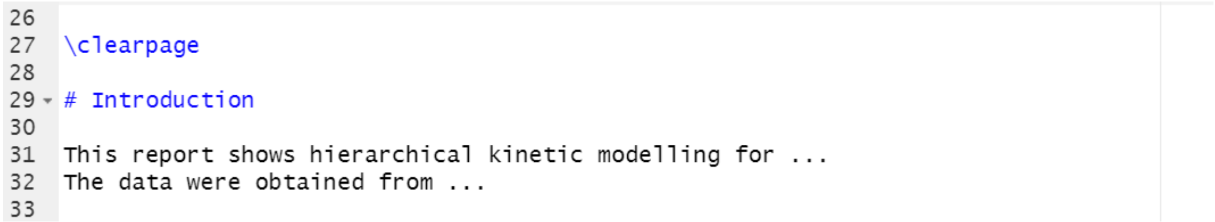
The remainder of the R code in this code chunk sets up a cluster for parallel computations. Depending on the operating system, a popup window may appear when setting up the cluster, asking for permissions to use a certain port that R uses for the communication between the different processes. Unfortunately, on Windows the message issued by this popup window is not very specific. If these permissions can be granted, working with the cluster should be possible in the following. If errors are experienced while setting up the cluster or later on when using the cluster, the user can go back to this code chunk, set the number of cores to one by modifying the line to read `n_cores <- 1` and executing the code chunk again. In this case, all analyses will be performed sequentially.

After executing the code chunk called “n_cores”, the variable `n_cores` and the cluster object `cl` should be visible in the Environment tabulator in the top right window.

3.2.4 Editing the markdown sections in the template

After the code chunk “n_cores”, a markdown section with three elements follows (Figure 4). The first element is the LaTeX formatting instruction “\clearpage”, simply specifying that a new page in the PDF output should be started at this point. The second element is “# Introduction”. The hash sign “#” at the beginning of a line means that a new level one section should start. The text after the hash sign is the header of that section.

Figure 4: Markdown section with LaTeX instruction, header markup and text



```

\clearpage

# Introduction

This report shows hierarchical kinetic modelling for ...
The data were obtained from ...

```

Source: Own contribution, Johannes Ranke

The third element is unformatted text starting with “This report”. The user should adapt this text to provide an adequate introductory description of the purpose of the report and describe where the data have been obtained.

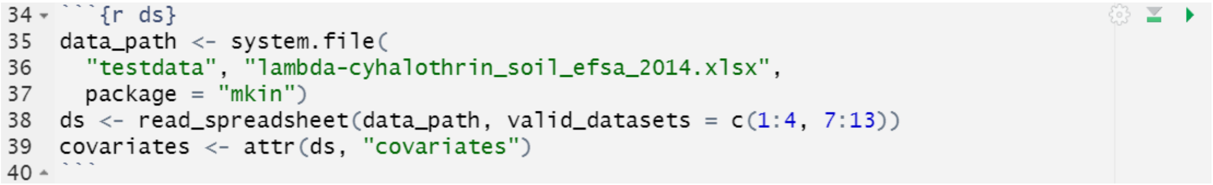
3.3 How to read in residue data from a spreadsheet file

The next code chunk named “ds” reads in the residue and covariate data from an example spreadsheet file that is included in the mkin package. This is done in three steps. In the first step, the path to the file is defined. In the template, the R function `system.file()` is used for this, because it returns the full path to the file, regardless where exactly the package has been installed.

In a second step, the function `read_spreadsheet()` is used to read in the data and to store it in an R object named `ds`. The argument `valid_datasets` is only necessary if not all datasets in the spreadsheet should be used. If all datasets given in the spreadsheet should be used, the command should be given as shown in Figure 6.

In the third step, the covariate data, which is stored as an attribute of the returned object, is made more readily available in an object named `covariates`.

Figure 5: Code chunk reading in residue and covariate data



```
34 ```{r ds}
35 data_path <- system.file(
36   "testdata", "lambda-cyhalothrin_soil_efsa_2014.xlsx",
37   package = "mkin")
38 ds <- read_spreadsheet(data_path, valid_datasets = c(1:4, 7:13))
39 covariates <- attr(ds, "covariates")
40 ```
```

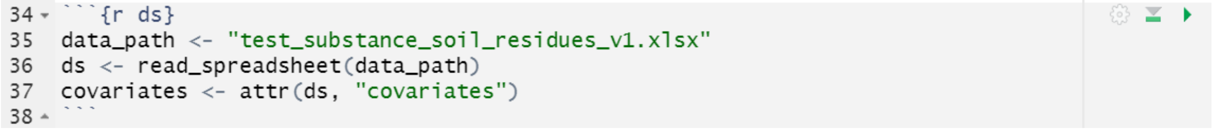
Source: Own contribution, Johannes Ranke

To read in own data, the example spreadsheet needs to be modified. One way to do this is to define the variable `data_path` by the R code given in the above code chunk, e.g. by putting the cursor on the lines containing the corresponding R command (lines 35 to 37), and pressing <Ctrl><Enter>. Then, the example spreadsheet file installed with the mkin package can be accessed by entering the command `browseURL(data_path)` in the R console window and pressing <Enter>. The spreadsheet file should then be opened in an appropriate program, provided such a program is installed.

The spreadsheet file (in the Office Open XML format which can be edited with commonly used software packages like Microsoft Excel and LibreOffice Calc) contains sheets named “Compounds”, “Sources”, “Datasets”, “Covariates”, “LOD LOQ” as well as sheets named with the numbers 1 to 13. While not all content in the spreadsheet is read in by `read_spreadsheet()`, it is recommended to edit the content of each sheet in order to maintain consistent and comprehensible information.

After editing the spreadsheet contents based on the explanations given in section 3.4, it should be saved under a different name in the folder in which the R markdown file is located using the .xlsx-format. Finally, the variable `data_path` needs to be redefined to specify the path to the new location of the spreadsheet. Figure 6 shows how the code reading in the data should be modified if the spreadsheet with the new data was saved in the same folder as the R markdown file under the name “test_substance_soil_residues_v1.xlsx”. Note that no call to `system.file()` is necessary, and that the function `read_spreadsheet()` only needs one argument (the path to the spreadsheet file) if all datasets in the file are to be used. As mentioned above, the argument `valid_datasets` is not necessary in that case.

Figure 6: Modified code chunk reading in the spreadsheet as edited by the user



~~~
```{r ds}
data_path <- "test_substance_soil_residues_v1.xlsx"
ds <- read_spreadsheet(data_path)
covariates <- attr(ds, "covariates")
```
~~~

Source: Own contribution, Johannes Ranke

3.4 Entering data into the spreadsheet file

In the following it is explained how available residue data and additional information on the test systems can be entered into the spreadsheet.

3.4.1 Sheet “Compounds”

In the column “Name” in the sheet “Compounds”, the most commonly used names of the parent compound and its transformation products should be listed as used in the data sources. In the column “Acronym”, compound acronyms should be defined that can be used as R variable names. Such variable names can contain alphabetic characters and numbers as well as the underscore character. R variable names cannot start with a number, and should not contain a dash, as the dash is interpreted as the minus sign. Using spaces in compound acronyms has not been tested and is therefore likely to cause problems. It is recommended to use six characters or less, as some of the degradation parameters are built from compound acronyms and long degradation parameter names can disturb the formatting of summary listings.

In the column “Transformation products”, comma separated acronyms of transformation products can be given for each compound. Finally, synonyms can be given in the column “Synonyms”, in order to clarify the chemical identity of the compounds.

Currently, the information given in the sheet “Compounds” is read in, but not used for degradation model building. In the future, the columns “Acronym” and “Transformation products” could be used for building candidate degradation models.

3.4.2 Sheet “Sources”

The column “Source numbers” in the sheet “Sources” can be used to number the sources, but can also be left empty. The column “Acronym” should specify acronyms that can be used in other sheets to indicate where a particular piece of information was obtained. The “Citation” column should give full bibliographic information on the sources, where available. If the source was retrieved from the internet, a uniform resource location (URL) can be given in the corresponding column.

3.4.3 Sheet “Datasets”

For each running number given in the column “Dataset Number” in the sheet “Datasets”, an additional sheet containing the corresponding data must exist. In the given example spreadsheet file, dataset numbers range from 1 to 13, and equally named sheets are defined.

The data read in from the numbered sheets containing the individual datasets will be grouped by the identifiers given in the second column of the sheet “Datasets”. In our example, this column is named “Soil”. Data from datasets with the same group identifier will be merged together. Identifiers used here can contain spaces and language specific characters like the German Umlauts or characters with accents, but their length should not exceed 14 characters in order to keep figure legends readable.

In the “Source” column of the sheet “Datasets”, the data source should be specified, preferably using the acronyms defined in the “Sources” sheet. To be more specific, page numbers can be given in the “Pages” column.

The next two columns “Temperature” and “Moisture” should contain 1, if no time-step normalisation should be performed. This would be the case when the study was already performed at the desired reference temperature and moisture conditions (e.g. 20°C and pF 2) or when a time step normalisation has already performed on the residue data before starting the kinetic evaluation (e.g. time-step normalisation of residue data from field studies). Otherwise, a time step normalisation factor should be entered to convert the residue data to identical reference temperature and moisture conditions, allowing combined evaluations of degradation data series that were obtained at different temperature and moisture conditions. Instructions how to derive time step normalisation factors for degradation data obtained in the laboratory or in field studies can be found in the FOCUS kinetic guidance.

The column “Total” simply contains a formula for multiplication of the temperature and moisture normalisation factors for each dataset.

In the “Label” column, the radioactive label position can be specified, to make it easier to uniquely identify each dataset. Replicate measurements with identical position of the radiolabel should be entered in the same data sheet. If the same substance was tested with different radiolabel positions in the same soil, the datasets can be entered separately, but will be combined while reading in, if the exact equal soil name has been used.

Finally, additional remarks that are important for the evaluation can be given for each dataset in the “Remarks” column.

3.4.4 Sheet “Covariates”

In case the influence of covariates like soil pH should be investigated, data should be entered in the sheet “Covariates”. In the first column, all identifiers used in the second column of the “Datasets” sheet must be listed. The column name of the first column is not used, but to avoid confusion, it is advisable to use the same heading that is used in the Sheet “Compounds” under “Name”. The data source of the covariate information should be specified in the columns “Source” and “Pages”.

All columns starting from the fourth column will be used as covariate data, with the exception of column “Remarks”. The column headers of the covariate data (“pH” in the example spreadsheet) should be descriptive of the content and should also be valid R variable names, as they will be used for specifying covariate models. It is recommended to use short names for the covariate identifiers, starting with a character and using only alphabetic characters, numbers and the underscore character.

If you do not want to use covariate data, you can delete the lines with the data (not the header line) from the spreadsheet. A message will be displayed that no covariate data is read in.

3.4.5 Sheet “LOD LOQ”

The sheet “LOD LOQ” can be used to store information about the limit of detection (LOD) and the limit of quantification (LOQ) that is used for any pre-processing of the data, for example based on the recommendations of the FOCUS kinetics guidance. The first column holds the acronym for which the LOD or LOQ is given, the second column indicates the range of dataset numbers for which they are valid. In the third and fourth column, the actual values of the LOD and the LOQ are given. The fifth and sixth column indicate source and page numbers, and the seventh column has place for remarks. The information in this sheet is not read in, it only serves as a basis for remarks on pre-processing in the numbered dataset sheets, because LOD and LOQ are often used to derive surrogate values for non-detected or non-reported values.

3.4.6 Numbered sheets

Each of the numbered datasheets must hold one of the datasets listed in the sheet "Datasets". The first column "Observed" holds the name of the observed variable, which is typically simply the acronym of the observed compound. The column "Time" holds the time after application of the compound when the concentration of the compound or its transformation product was measured, the column "Value" holds the numerical values, i.e. the residue concentration. Non-detects should be either left blank or set to a numerical value based on a relevant recommendation, for example the FOCUS guidance. In the "Remark" column, any pre-processing should be documented. In the case of non-numeric values in the original residue tables, at least the notation used there (e.g. "n.d." for non-detected or "<0.1") should be included in the remark.

3.5 Evaluation of residue data on the parent compound

Once the residue data are entered into the spreadsheet and read into the R object, suitable starting values for fitting the hierarchical models need to be obtained. To find such starting values, separate evaluations of the parent residue datasets are performed in the next code chunk entitled "parent-sep". The fitting function `mmkin`[9] used here performs generalised nonlinear regression of a set of degradation models to a number of datasets in parallel, using as many computing cores at the same time as specified above.

The set of degradation models that is fitted can be adapted by changing the definition of the variable `parent_deg_mods`. After successful execution of the code chunk, the fits of the selected degradation models to the datasets will be available in two arrays of fit objects, `parent_sep_const` and `parent_sep_tc`, holding the results for constant variance and two-component error, respectively.

The code in the next code chunk ("parent-mhmkin") contains the instructions to fit the hierarchical kinetic models that result from the possible combination of the degradation models defined above and the two error models. For example, if you use the definition of the parent degradation models unchanged as given in the template, you will have eight versions of hierarchical fits (four degradation models times tow error models) that you can compare.

The fitting function `mhmkin` performs these fits in parallel, using as many computing cores as specified above. Each of the parallel fits is performed using the `saem` function which in turn uses the saemix package as backend.

In this step, variation of all degradation parameters across the population of experimental entities is assumed, i.e. there is a random effect for each degradation parameter. The output produced by this code chunk should be a table with two columns for the two error models, containing an entry "OK" for each successful fit. If any of the fits were unsuccessful, the corresponding entries in the table would be "E" for error and the cause for the error should be investigated.[10] In case that there are entries "Fth" or "FO" in the output, they indicate that the

[9] The documentation of this fitting function and links to the documentation of the related functions in the mkin package can be found online under https://pkgdown.jrwb.de/mkin/reference/mmkin.html

[10] It is not within the scope of this report to give a comprehensive strategy for finding the source of such errors. In general, the error message(s) should be studied, data should be checked for transcription errors, and the separate fits should be investigated. For example, if the hierarchical fit of the HS degradation model in combination with constant variance resulted in an error, the corresponding separate fits of the HS model can be plotted using the command `plot(parent_sep_const["HS", ]).A`

model is overparameterised and should be simplified. In some cases, this can be addressed by removing some random effects from the model (see below), in other cases a simpler model with fewer parameters should be selected.

The next code chunk lists the ill-defined variance parameters for each hierarchical model fit. For the variance parameters defining the degradation parameter distribution, a different terminology is used in the output, i.e. the standard deviation of the initial residue of the compound "lambda" is called `sd(lambda_0)` instead of `SD.lambda_0`[11].

In the following code chunks, the hierarchical model fits are repeated with the variance components that were found to be ill-defined set to zero. The status of these refined fits is checked, and it is checked if all of the refined models actually result in a lower Akaike Information Criterion (AIC) compared to the respective ill-defined model (Akaike, 1974). From the refined fits, the most suitable combination of degradation model and error model is identified based on the AIC. Then, the model comparison function `anova()` is used to tabulate the degrees of freedom, AIC and Bayesian Information Criterion (BIC) values and the likelihood of the refined fits (Burnham und Anderson, 2004).

For the selected best fit, it is subsequently checked, if any ill-defined parameters remain using the `illparms()` function . If no parameter names are listed, there are no remaining ill-defined parameters.

With the next two code chunks, the best fit found is plotted, and the parameters of that fit are listed together with their approximate 95% confidence intervals.

3.6 Evaluation of the parent compound including a covariate

The correlation of the random effects in the selected best fit with the chosen covariate, e.g. the soil pH, which is available in the covariate data, is checked in the code chunk "parent-best-pH". In the template file, the first two fits check the correlation of a single degradation parameter with the covariate, and the third fit is based on a covariate model including correlations of both degradation parameters with the covariate. The parameter names in the covariate model specification have to be adapted to the degradation parameters and covariate names for the specific case under consideration. For example, the covariate model `log_k_lambda_free ~ pH` could be replaced by a covariate model `log_k_parent ~ clay` if the correlation of a first-order rate constant of a substance with acronym "parent" with soil clay content is to be investigated. Of course, this only works if the transformed degradation parameter on the left-hand side of the model formula is actually part of the degradation model, and the covariate variable name is actually available in the data frame specified in the argument `covariates`. Thus, they need to be adapted to the selected degradation model and the available covariate data.

The succeeding model comparison based on the AIC indicates that in the case of the example data, the second model with pH correlation, i.e. the one with only the desorption rate constant correlating with soil pH is the most preferable hierarchical degradation model for the parent compound. For this model, it is confirmed that no ill-defined parameters remain. The model is plotted and the parameters are listed with their approximate 95% confidence intervals.

comprehensive listing of the HS fit to the first dataset can be obtained using the command `summary(parent_sep_const["HS", 1])`.

[11] This alternative terminology is used in some functions in mkin for compatibility with the nlme package, which was the first backend used in mkin for fitting nonlinear hierarchical models.

3.7 How to evaluate residue data including transformation products

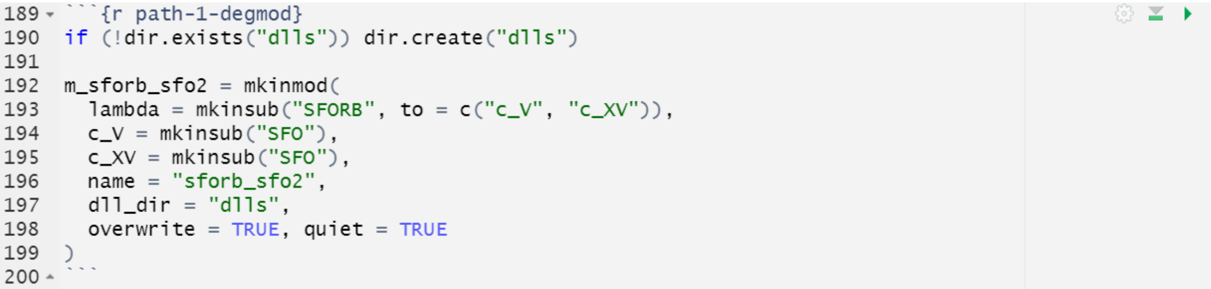
Setting up pathway fits for evaluation with nonlinear hierarchical models is an advanced procedure that requires more understanding of R and the way pathway models are set up in the mkin package. In case the reader wants to just fit the data for the parent compound, the template can be reduced by deleting the section on pathway fits and the corresponding section in the Appendix. The section on pathway fits starts with "# Pathway fits" (or, if the visual editor is used, the corresponding heading typeset in large bold letters). It ends with the line containing "\clearpage" just before the heading of the Appendix. Within the Appendix, the subsection starting with "## Listings of pathway fits" has to be deleted up to, but not including, the heading "## Session info".

Fitting the hierarchical degradation models for the parent compound is reasonably fast, because it is based on analytical solutions of the differential equations defining the models. If there is only one transformation product and the model for the parent compound is SFO or DFOP, the speed is nearly the same, because analytical solutions for these models are also implemented in mkin.

However, in most cases where transformation products are to be included, such analytical solutions are not implemented, and the degradation models have to be solved iteratively. To limit the execution time for fitting such models using the stochastic algorithm implemented in the saemix package, a compiler for the C programming language should be installed as described above. If this is the case, setting up a degradation model without an analytical solution will produce a dynamically loadable library (DLL) which is then used in the model fitting process to speed up the model solutions.

To avoid fitting the same model repeatedly when working on the R markdown template, the template used here activates the caching function of the knitr package. However, to make caching work across R sessions, e.g. when continuing the work on the next day, the DLLs have to be permanently stored. For this purpose, a directory "dlls" is created and its name is specified when the degradation model is set up using the `mkinmod` function (Figure 7).

Figure 7: Code chunk for setting up a degradation model with transformation products



~~~
```{r path-1-degmod}
if (!dir.exists("dlls")) dir.create("dlls")

m_sforb_sfo2 = mkinmod(
  lambda = mkinsub("SFORB", to = c("c_V", "c_XV")),
  c_V = mkinsub("SFO"),
  c_XV = mkinsub("SFO"),
  name = "sforb_sfo2",
  dll_dir = "dlls",
  overwrite = TRUE, quiet = TRUE
)
```
~~~

For the use of this function, it is important to know that it has two types of arguments, the "dots" arguments named after the observed variables to be modelled and the remaining optional named arguments as specified in the documentation[12]. In the example shown above, the names of the "dots" arguments are `lambda`, `c_V` and `c_XV`, because these are the compound acronyms used for the different types of residues in the data. The value given for the `lambda` argument is a submodel specification as returned by the function `mkinsub`.

In the submodel specifications, the type of submodel is specified as a character string, for example `"SFO"` or `"SFORB"`. The target compartments are given in the second argument as a

[12] https://pkgdown.jrwb.de/mkin/reference/mkinmod.html

vector of character strings, holding the acronyms used in the residue data for the target compartments. In this example, the acronyms for residues of compound V and compound XV are given, meaning that transformation of lambda-cyhalothrin to these two compounds is assumed in the model definition.

The submodel specifications for compound V and compound XV are simply expressed by mkinsub("SFO"), because they are modelled using first-order degradation and their degradation products are not included in the model.

After the submodel specifications, a number of optional arguments can be specified. If both name and dll_dir are specified, then a function for speeding up the model solution is saved in the directory specified by dll_dir, using the specified name for the filename.

In the next code chunk "path-1-sep", the model is fitted to all datasets, first assuming constant variance, and then assuming two-component error. If there are alternative pathway models that should be compared, more than one degradation model can be included in this code chunk leading to parallel fitting of both degradation models to all datasets. For example, if not only a model m_sforb_sfo2 is defined above, but also a model named e.g. m_sfo_sfo2, the first call to mmkin could be modified to be mmkin(list(sforb_path = m_sfo_sfo2, sfo_path = m_sfo_sfo2), …).

Plotting the results from the separate fits as done in the next code chunk is advisable to check the suitability of the degradation model for the parent compound before a hierarchical fit is attempted. If more than one degradation model was used as described above, a row index has to be specified to select the degradation model. In our example, the adapted plotting command for selecting the alternative pathway model with SFO used for the parent compound would be plot(mixed(sforb_sep_const["sfo_path",])).

The following code chunk "path-1" fits the corresponding hierarchical models. In the template, only one degradation model is used in the separate fits, so the resulting object is an array with two fits, one with constant variance, and one with two-component error. If two degradation models are used in the separate fits, the resulting object will hold the results for the four combinations of degradation models and error models. As for the parent fits, one more more covariate models can be specified for the kinetic parameters of the degradation model.

After checking the status of the hierarchical fits, the first set of hierarchical pathway fits is compared. After selecting the most suitable model, ill-defined variance parameters are identified for the best fit model. Then, the best-fit model is updated in the code chunk "path-1-refined", with additional random effects excluded. Note that any random effect that has been excluded in the first version of the fit (code chunk "path-1") has to be specified again in the argument no_random_effect.

If no error messages are observed, the fit is checked for any remaining ill-defined variance parameter. If no output is obtained from this check, the refined fit is plotted and the parameters of the refined fit are tabulated, together with the approximate 95% confidence intervals.

The remainder of the template sets up the Appendix and the listings of the parent fits. This part of the template generally does not have to be adapted. However, in the Appendix subsection for listings of pathway fits, adaptations to the calls to the function tex_listing() have to be made, to correctly reflect the names and descriptions of the fit objects.

4 Conclusions

To address shortcomings in the methods currently used for the kinetic evaluation of chemical degradation data, it has been proposed to use nonlinear mixed-effects models, also called hierarchical or multilevel models. The advantages of this approach have been shown in previous projects.

In this report, a short introduction to this type of models is presented. Further, it is described how to use the R markdown template recently added to the mkin package, in combination with the newly developed spreadsheet file for entering data. These tools make it much easier to apply the proposed method to new data.

In the course of this project, fitting hierarchical kinetic degradation models has additionally been tested with some more experimental datasets and the inclusion of covariates in these models has been greatly simplified. Methods for model reduction to the point where the models are fully identifiable have been implemented. Finally, the spreadsheet for entering data and the R markdown template for performing and documenting kinetic evaluations mentioned above have been developed and are now part of the mkin package, in the hope that the method can now be tested by a wider audience.

To implement hierarchical kinetic modelling in regulatory degradation kinetics, a guidance document would need to be developed, providing recommendations how the results from hierarchical degradation kinetics should be used in the various regulatory areas where kinetic degradation endpoints are relied upon.

5 References

Akaike, H. (1974): A New Look at the Statistical Model Identification. IEEE Transactions on Automatic Control 19 (6): 716–23

Burnham, K.P.; Anderson, D.R. (2004): Multimodel inference – Understanding AIC and BIC in Model Selection. Sociological Methods and Research 33(2): 261-304

Comets, E.; Lavenu, A.; Lavielle, M. (2017): Parameter estimation in nonlinear mixed effect models using saemix, an R implementation of the SAEM algorithm. Journal of Statistical Software, 80(3), 1-41.

EFSA (2014): Guidance of EFSA: EFSA Guidance document for evaluating laboratory and field dissipation studies to obtain DegT50 values of active substances of plant protection products and transformation products of these active substances in soil. EFSA Journal 2014;12(5):3662.

FOCUS (2006) Guidance document on estimating persistence and degradation kinetics from environmental fate studies on pesticides in EU registration - Report of the FOCUS Work Group on Degradation Kinetics, EC Doc. Ref. Sanco/10058/2005, version 2.0).

FOCUS (2014): Generic guidance for estimating persistence and degradation kinetics from environmental fate studies on pesticides in EU registration, document based on the official guidance document of FOCUS Degradation Kinetics in the context of 91/414/EEC and Regulation (EC) No 1107/2009, version 1.1. https://esdac.jrc.ec.europa.eu/projects/degradation-kinetics

Lavielle, M. (2015): Mixed Effects Models for the Population Approach—Models, Tasks, Methods and Tools. CRC Press, Boca Raton, FL

Pinheiro, J.C.; Bates, D.M. (2000): Mixed-Effects Models in S and S-Plus. Springer Verlag, New York

R Core Team (2022). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. https://www.R-project.org

Ranke, J. (2022): mkin 1.2.2: Kinetic Evaluation of Chemical Degradation Data. https://CRAN.R-project.org/package=mkin

Ranke, J.; Lehmann, R. (2012): Parameter reliability in kinetic evaluation of environmental metabolism data – Assessment and the influence of model specification. Poster presented at SETAC World 20-24 May 2012, Berlin, Germany.

Ranke, J.; Wöltjen, J., Schmidt, J., Comets, E. (2021): Taking kinetic evaluations of degradation data to the next level with nonlinear mixed-effects models. Environments 8 (8) 71. https://doi.org/10.3390/environments8080071

Ranke, J.; Meinecke, S. (2019): Error models for the kinetic evaluation of chemical degradation data. Environments 6 (12) 124. https://doi.org/10.3390/environments6120124

Ranke, J.; Wöltjen, J.; Meinecke S. (2018): Comparison of Software Tools for Kinetic Evaluation of Chemical Degradation Data. Environmental Sciences Europe 30 (1): 17. https://doi.org/10.1186/s12302-018-0145-1

Ranke, J.; Wöltjen, J. (2022): Degradation kinetics on the next level. SETAC Europe 32nd Annual Meeting, 12 – 15 May 2022, Copenhagen