mkin - Routines for fitting kinetic models with one or more state variables to chemical degradation data

Johannes Ranke
Product Safety
Harlan Laboratories Ltd.
Zelgliweg 1, CH–4452 Itingen, Switzerland

May 14, 2010

Abstract

In the regulatory evaluation of chemical substances like plant protection products (pesticides), biocides and other chemicals, degradation data play an important role. For the evaluation of pesticide degradation experiments, detailed guidance has been developed, based on nonlinear optimisation. The R add-on package **mkin** implements fitting some of the models recommended in this guidance from within R and calculates some statistical measures for data series within one or more compartments, for parent and metabolites.

Contents

1 Introduction 1

2 Example 1
2.1 Data format 1
2.2 Model definition 2
2.3 Fitting the model 2

Key words: Kinetics, FOCUS, nonlinear optimisation

1 Introduction

Many approaches are possible regarding the evaluation of chemical degradation data. The **kinfit** package (Ranke, 2010a) in R (R Development Core Team, 2010) implements the approach recommended in the kinetics report provided by the FOrum for Co-ordination of pesticide fate models and their USe (FOCUS Work Group on Degradation Kinetics, 2006) for simple data series for one parent compound in one compartment.

The **mkin** package (Ranke, 2010b) extends this approach to data series with metabolites and more than one compartment and includes the possibility for back reactions.

2 Example

In the following, requirements for data formatting are explained. Then the procedure for fitting the four kinetic models recommended by the FOCUS group to an example dataset for parent only given in the FOCUS kinetics report is illustrated. The explanations are kept rather verbose in order to lower the barrier for R newcomers.

2.1 Data format

The following listing shows example dataset C from the FOCUS kinetics report as distributed with the **kinfit** package

```
R> library("mkin")
R> FOCUS_2006_C
```

```
    name time value
1 parent    0  85.1
2 parent    1  57.9
3 parent    3  29.9
```

```
4 parent    7  14.6
5 parent   14   9.7
6 parent   28   6.6
7 parent   63   4.0
8 parent   91   3.9
9 parent  119   0.6
```

Note that the data needs to be in the format of a data frame containing a variable `name` specifying the observed variable, indicating the compound name and, if applicable, the compartment, a variable `time` containing sampling times, and a numeric variable `value` specifying the observed value of the variable. If a further variable `error` is present, this will be used to give different weights to the data points (the higher the error, the lower the weight, see the help page of the `modCost` function of the **FME** package (Soetaert and Petzoldt, 2010)). Replicate measurements are not recorded in extra columns but simply appended, leading to multiple occurrences of the sampling times `time`.

Small to medium size dataset can be conveniently entered directly as R code as shown in the following listing

```
R> example_data <- data.frame(
+   time = c(0, 1, 3, 7, 14, 28, 63, 91, 119),
+   parent = c(85.1, 57.9, 29.9, 14.6, 9.7, 6.6, 4, 3.9, 0.6)
+ )
```

2.2 Model definition

The next task is to define the model to be fitted to the data. In order to facilitate this task, a convenience function `mkinmod` is available.

```
R> SFO <- mkinmod(spec = list(parent = list(type = "SFO", to = NA, sink = TRUE)))
R> SFORB <- mkinmod(spec = list(parent = list(type = "SFORB", to = NA, sink = TRUE)))
R> SFO_SFO <- mkinmod(spec = list(
+   parent = list(type = "SFO", to = "m1", sink = TRUE),
+   m1 = list(type = "SFO", to = NA, sink = TRUE)))
R> SFORB_SFO <- mkinmod(spec = list(
+   parent = list(type = "SFORB", to = "m1", sink = TRUE),
+   m1 = list(type = "SFO", to = NA, sink = TRUE)))
```

2.3 Fitting the model

Then the model parameters should be fitted to the data. The function `mkinfit` internally creates a cost function using `modCost` from the **FME** package and the produces a fit using `modFit` from the same package.

```
R> # Do not show significance stars as they interfere with vignette generation
R> options(show.signif.stars = FALSE)
R> SFO.fit <- mkinfit(SFO, FOCUS_2006_C)
```

```
Model cost at call  1 :  4718.97
Model cost at call  4 :  4718.97
Model cost at call  5 :  637.0032
Model cost at call  7 :  637.0032
Model cost at call  8 :  287.3209
Model cost at call  10 :  287.3209
Model cost at call  11 :  207.3576
Model cost at call  13 :  207.3576
Model cost at call  14 :  197.3848
Model cost at call  16 :  197.3848
Model cost at call  17 :  196.5886
Model cost at call  19 :  196.5886
Model cost at call  20 :  196.5369
Model cost at call  22 :  196.5369
Model cost at call  23 :  196.5338
Model cost at call  25 :  196.5338
Model cost at call  26 :  196.5336
Model cost at call  28 :  196.5336
Model cost at call  29 :  196.5336
Model cost at call  31 :  196.5336
Model cost at call  32 :  196.5336
```

```
R> summary(SFO.fit)
```

```
Parameters:
             Estimate Std. Error t value Pr(>|t|)
parent_0      82.4920     4.7402  17.403 5.09e-07
k_parent_sink  0.3061     0.0459   6.668 0.000286

Residual standard error: 5.299 on 7 degrees of freedom

Parameter correlation:
             parent_0 k_parent_sink
parent_0       1.0000        0.5212
k_parent_sink  0.5212        1.0000
```

```
R> SFORB.fit <- mkinfit(SFORB, FOCUS_2006_C)
```

```
Model cost at call  1 :  7044.136
Model cost at call  4 :  7044.136
Model cost at call  7 :  2652.892
Model cost at call  9 :  2652.892
Model cost at call  13 :  865.8227
Model cost at call  15 :  865.8227
Model cost at call  18 :  47.70534
Model cost at call  19 :  47.70533
```

```
Model cost at call  23 :  43.25850
Model cost at call  25 :  43.2585
Model cost at call  28 :  7.011757
Model cost at call  30 :  7.011757
Model cost at call  32 :  7.011757
Model cost at call  33 :  4.42402
Model cost at call  35 :  4.42402
Model cost at call  37 :  4.42402
Model cost at call  38 :  4.363098
Model cost at call  40 :  4.363098
Model cost at call  42 :  4.363098
Model cost at call  43 :  4.362712
Model cost at call  44 :  4.362712
Model cost at call  45 :  4.362712
Model cost at call  48 :  4.36271
Model cost at call  50 :  4.36271
Model cost at call  52 :  4.36271
Model cost at call  57 :  4.36271
```

```
R> summary(SFORB.fit)
```

```
Parameters:
                     Estimate Std. Error t value Pr(>|t|)
parent_free_0       85.002734   0.890670  95.437 2.39e-09
k_parent_free_sink   0.395044   0.014308  27.610 1.17e-06
k_parent_free_bound  0.061599   0.007289   8.452 0.000381
k_parent_bound_free  0.020764   0.003752   5.534 0.002643

Residual standard error: 0.9341 on 5 degrees of freedom

Parameter correlation:
                    parent_free_0 k_parent_free_sink k_parent_free_bound
parent_free_0             1.00000             0.5217              0.1813
k_parent_free_sink        0.52168             1.0000              0.6693
k_parent_free_bound       0.18127             0.6693              1.0000
k_parent_bound_free       0.07641             0.3062              0.6756
                    k_parent_bound_free
parent_free_0                   0.07641
k_parent_free_sink              0.30622
k_parent_free_bound             0.67560
k_parent_bound_free             1.00000
```

```
R> SFO_SFO.fit <- mkinfit(SFO_SFO, FOCUS_2006_D)
```

```
Model cost at call  1 :  18994.29
Model cost at call  3 :  18994.29
Model cost at call  8 :  15888.58
Model cost at call  9 :  15888.58
Model cost at call  13 :  9262.957
Model cost at call  14 :  9262.957
```

```
Model cost at call  18 :  1784.113
Model cost at call  20 :  1784.113
Model cost at call  23 :  387.3529
Model cost at call  25 :  387.3529
Model cost at call  28 :  371.2284
Model cost at call  30 :  371.2284
Model cost at call  31 :  371.2284
Model cost at call  33 :  371.2127
Model cost at call  34 :  371.2127
Model cost at call  35 :  371.2127
Model cost at call  36 :  371.2127
Model cost at call  38 :  371.2127
```

```
R> summary(SFO_SFO.fit)
```

```
Parameters:
              Estimate Std. Error t value Pr(>|t|)
parent_0      9.960e+01  1.614e+00  61.720  < 2e-16
k_parent_sink 4.792e-02  3.750e-03  12.777 6.10e-15
k_m1_sink     5.261e-03  7.159e-04   7.349 1.15e-08
k_parent_m1   5.078e-02  2.094e-03  24.249  < 2e-16

Residual standard error: 3.211 on 36 degrees of freedom

Parameter correlation:
              parent_0 k_parent_sink k_m1_sink k_parent_m1
parent_0       1.00000       0.60753   -0.1701    -0.06623
k_parent_sink  0.60753       1.00000   -0.6253    -0.08736
k_m1_sink     -0.17008      -0.62527    1.0000     0.47160
k_parent_m1   -0.06623      -0.08736    0.4716     1.00000
```

```
R> SFORB_SFO.fit <- mkinfit(SFORB_SFO, FOCUS_2006_D)
```

```
Model cost at call  1 :  16413.78
Model cost at call  3 :  16413.78
Model cost at call  10 :  11483.33
Model cost at call  11 :  11483.33
Model cost at call  17 :  1264.832
Model cost at call  18 :  1264.832
Model cost at call  24 :  558.8548
Model cost at call  25 :  558.8547
Model cost at call  30 :  558.8547
Model cost at call  31 :  370.7622
Model cost at call  32 :  370.7621
Model cost at call  39 :  353.7515
Model cost at call  41 :  353.7515
Model cost at call  47 :  352.3025
Model cost at call  50 :  352.3025
Model cost at call  55 :  352.2217
Model cost at call  56 :  352.2217
```

```
Model cost at call  63 :  352.2049
Model cost at call  64 :  352.2049
Model cost at call  72 :  352.2047
Model cost at call  73 :  352.2047
Model cost at call  74 :  352.2047
Model cost at call  76 :  352.2047
Model cost at call  80 :  352.2047
```

```
R> summary(SFORB_SFO.fit)
```

```
Parameters:
                    Estimate Std. Error t value Pr(>|t|)
parent_free_0       1.011e+02  2.020e+00  50.034  < 2e-16
k_parent_free_sink  6.408e-02  2.691e-02   2.381   0.0230
k_parent_free_bound 1.680e-01  5.142e-01   0.327   0.7458
k_parent_bound_free 5.239e-01  8.544e-01   0.613   0.5438
k_m1_sink           5.213e-03  7.210e-04   7.230 2.29e-08
k_parent_free_m1    6.563e-02  2.542e-02   2.582   0.0143

Residual standard error: 3.219 on 34 degrees of freedom

Parameter correlation:
                    parent_free_0 k_parent_free_sink k_parent_free_bound
parent_free_0              1.0000             0.5432              0.34389
k_parent_free_sink         0.5432             1.0000              0.94317
k_parent_free_bound        0.3439             0.9432              1.00000
k_parent_bound_free        0.1950             0.8179              0.95432
k_m1_sink                 -0.1801            -0.2031             -0.09286
k_parent_free_m1           0.4402             0.9752              0.96052
                    k_parent_bound_free k_m1_sink k_parent_free_m1
parent_free_0                   0.19504  -0.18007          0.44015
k_parent_free_sink              0.81788  -0.20307          0.97518
k_parent_free_bound             0.95432  -0.09286          0.96052
k_parent_bound_free             1.00000  -0.08809          0.83988
k_m1_sink                      -0.08809   1.00000         -0.03944
k_parent_free_m1                0.83988  -0.03944          1.00000
```

References

FOCUS Work Group on Degradation Kinetics. *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*, 2006. URL http://focus.jrc.ec.europa.eu/dk. EC Document Reference Sanco/10058/2005 version 2.0.

R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2010. URL http://www.

R-project.org. ISBN 3-900051-07-0.

Johannes Ranke. *kinfit: Routines for fitting simple kinetic models to chemical degradation data*, 2010a. URL http://CRAN.R-project.org.

Johannes Ranke. *mkin: Routines for fitting kinetic models with one or more state variables to chemical degradation data*, 2010b. URL http://CRAN.R-project.org.

Karline Soetaert and Thomas Petzoldt. Inverse modelling, sensitivity and monte carlo analysis in R using package FME. *Journal of Statistical Software*, 33(3):1–28, 2010. URL http://www.jstatsoft.org/v33/i03/.