# **Supplementary**

Supplementary Tabel 1: Reaction mixture and conditions for reverse transcription

|  |  |  |
| --- | --- | --- |
| Reaction | mixture | Temperature profile |
| reaction 1  (two preparations per sample) | 11 µl DNA digestion sample  1µl random primer  1 µl dNTPs | 5 min at 65°C  >= 1 min at 4°C |
| reaction 2  (for each sample) | 1. Preparation (+ samples):   4 µl 5x First-Strand Buffer  1 µl 0.1 M DTT (100mM)  1 µl RNase OUT  1 µl reverse transcriptase (200 U/μl)  13 µl reaction 1 | 5 min at 25°C  60 min at 50°C  15 min at 70°C  Cool-down at 4°C |
|  | 1. Preparation (- samples):   4 µl 5x First-Strand Buffer  1 µl 0.1 M DTT (100mM)  1 µl RNase OUT  1 µl DEPC water  13 µl reaction 1 |  |

**R code of the applied model**

require(ggplot2)

library(purrr)

library(dplyr)

library(tidyr)

library(broom)

library(nls.multstart)

library(nlstools)

library(nleqslv)

library(devtools)

library(ggpubr)

## Model after Duo-Sen, L. and Shui-Ming, Z. (1987)

# Kinetic model for degradative processes of pesticides in soil.

# Ecological Modelling 37, 131-138. doi:10.1016/0304-3800(87)90021-4

# set working directory

getwd()

setwd("E:/Campos/Campos/R Skript/Model\_Holger")

HBW <-read.table("Halbwertzeit\_statistik.csv",header = T, sep = ";" )

# read data

Halbwertzeit<-(Halbwertszeit\_statistik)

head(dat)

# set factors

dat$rep<-as.factor(dat$rep)

dat$mcpa<-as.factor(dat$mcpa)

dat <-dat%>%

mutate(time\_d = Incubation\_time\_hours/24)

write.table(pars\_final, file = "Halbwertszeit.csv", sep = ";", col.names = NA,qmethod = "double")

# DEFINITION OF FITTING FUNCTIONS

f\_full<-function(dat){

# fit over each set of groupings

fits <- dat %>%

group\_by(., mcpa, rep) %>%

nest() %>%

mutate(fit = map(data, ~ nls\_multstart(percent ~ C0-C0/((1-f\_k)\*exp(k1\*time\_d)+f\_k)+S0\*(1-exp(-k2\*time\_d)),

data = .x,

iter = 1000,

start\_lower = c(C0=0, f\_k=0, k1=0, S0=0, k2=0),

start\_upper = c(C0=60, f\_k=0.999, k1=10,S0=60,k2=10),

supp\_errors = 'Y',

na.action = na.omit,

lower = c(C0=0, f\_k=0, k1=0, S0=0, k2=0),

upper = c(C0=60, f\_k=0.999, k1=10,S0=60,k2=10))))

# get summary

info <- fits %>%

mutate(summary = map(fit, glance)) %>%

unnest(summary)

# get params

params <- fits %>%

mutate(., p = map(fit, tidy)) %>%

unnest(p)

# get confidence intervals

CI <- fits %>%

mutate(., cis = map(fit, confint2),

cis = map(cis, data.frame)) %>%

unnest(cis) %>%

rename(., conf.low = X2.5.., conf.high = X97.5..) %>%

group\_by(., mcpa,rep) %>%

mutate(., term = c('C0', 'f\_k','k1','S0','k2')) %>%

ungroup() %>%

select(., -data, -fit)

# merge parameters and CI estimates

params <- merge(params, CI, by = intersect(names(params), names(CI)))

# plot parameter estimates and confidence intervalls

params

# new data frame of predictions

new\_preds <- dat %>%

do(., data.frame(time\_d = seq(0, max(.$time\_d), length.out = 300), stringsAsFactors = FALSE))

# create new predictions

preds2 <- fits %>%

mutate(., p = map(fit, augment)) %>%

unnest(p)

rename(., percent = .fitted)

# plot

p<-ggplot() +

geom\_point(aes(time\_d, percent, col = rep), size = 2, dat) +

geom\_line(aes(time\_d, .fitted, col = rep, group = rep), alpha = 0.5, preds2) +

facet\_wrap(~ mcpa, labeller = labeller(.multi\_line = FALSE)) +

scale\_color\_brewer(type='qual') +

theme\_bw(base\_size = 12) +

ylab('Cumulative 14C mineralization (%)') +

xlab('time (days)') +

theme(legend.position = c(0.9, 0.15))

p

# merge parameter values

C0s<-subset(params,term=="C0",select=c(mcpa,rep,estimate))

colnames(C0s)<-c('mcpa','rep','C0')

S0s<-subset(params,term=="S0",select=c(mcpa,rep,estimate))

colnames(S0s)<-c('mcpa','rep','S0')

f\_ks<-subset(params,term=="f\_k",select=c(mcpa,rep,estimate))

colnames(f\_ks)<-c('mcpa','rep','f\_k')

k1s<-subset(params,term=="k1",select=c(mcpa,rep,estimate))

colnames(k1s)<-c('mcpa','rep','k1')

k2s<-subset(params,term=="k2",select=c(mcpa,rep,estimate))

colnames(k2s)<-c('mcpa','rep','k2')

pars<-merge(C0s,S0s,by=c('mcpa','rep'))

pars<-merge(pars,k1s,by=c('mcpa','rep'))

pars<-merge(pars,k2s,by=c('mcpa','rep'))

pars<-merge(pars,f\_ks,by=c('mcpa','rep'))

# define function to calculate half-live

f\_hl <-function(t\_hl,C0,S0,k1,k2,f\_k){

y=-exp(k2\*t\_hl)+(exp(k1\*t\_hl)\*(2\*f\_k\*S0-2\*S0)-2\*f\_k\*S0)/(exp(k1\*t\_hl)\*(f\_k\*(S0+C0)-S0-C0)+f\_k\*(-S0-C0)+2\*C0)

}

f\_hl <-function(t\_hl,C0,S0,k1,k2,f\_k){

y=exp(k2\*t\_hl)-(exp(k1\*t\_hl)\*(2\*f\_k\*S0-2\*S0)-2\*f\_k\*S0)/(exp(k1\*t\_hl)\*(f\_k\*(S0+C0)-S0-C0)-f\_k\*(S0+C0)+2\*C0)

}

# set start values for numerical estimation of half-lifes

pars<-arrange(pars,mcpa,rep)

pars$start=c(2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2,2)

# calculate half-lives

hl <- pars %>%

group\_by(., mcpa, rep) %>%

nest() %>%

mutate(t\_hl = purrr::map(data,

~nleqslv(.x$start, f\_hl,control=list(btol=1e-2,maxit=1000,allowSingular=T),jacobian=FALSE,

method="Newton",xscalm="auto",C0=.x$C0,S0=.x$S0,k1=.x$k1,k2=.x$k2,f\_k=.x$f\_k)))

hl %>%

unnest(cols = "t\_hl")

pars\_final <- hl %>%

unnest(t\_hl %>% map(~.x%>%data.frame()))

pars\_final

pars\_final <- hl %>%

mutate(., p = map(~.x, data.frame()))# %>%

data.frame(p)

unnest(p)

pars\_final<-arrange(pars\_final,mcpa,rep)

params <- fits %>%

mutate(., p = map(fit, tidy)) %>%

unnest(p)

# calculate mean half-lifes and plot half-lifes vs. initial conc.

t\_hl\_mean <- pars\_final %>%

group\_by(., mcpa) %>%

summarise(HL=mean(t\_hl),HL\_sd=sd(t\_hl))

return(list("p"=p,"t\_hl\_mean"=t\_hl\_mean,"pars"=params))

}

f\_1st\_order<- function(dat){

# model fit

# fit over each set of groupings

fits <- dat %>%

group\_by(., mcpa, rep) %>%

nest() %>%

mutate(fit = purrr::map(data, ~ nls\_multstart(percent ~ C0\*(1-exp(-k1\*time\_d)),

data = .x,

iter = 1000,

start\_lower = c(C0=0, k1=0),

start\_upper = c(C0=60, k1=10),

supp\_errors = 'Y',

na.action = na.omit,

lower = c(C0=0, k1=0),

upper = c(C0=60,k1=10))))

info <- fits %>%

mutate(summary = map(fit, glance)) %>%

unnest(summary)

# get params

params <- fits %>%

mutate(., p = map(fit, tidy)) %>%

unnest(p)

# get confidence intervals

CI <- fits %>%

mutate(., cis = map(fit, confint2),

cis = map(cis, data.frame)) %>%

unnest(cis) %>%

rename(., conf.low = X2.5.., conf.high = X97.5..) %>%

group\_by(., mcpa,rep) %>%

mutate(., term = c('C0', 'k1')) %>%

ungroup() %>%

select(., -data, -fit)

# merge parameters and CI estimates

params <- merge(params, CI, by = intersect(names(params), names(CI)))

# plot parameter estimates and confidence intervalls

params

# new data frame of predictions

new\_preds <- dat %>%

do(., data.frame(time\_d = seq(0, max(.$time\_d), length.out = 300), stringsAsFactors = FALSE))

# create new predictions

preds2 <- fits %>%

mutate(., p = map(fit, augment)) %>%

unnest(p)

rename(., percent = .fitted)

# plot

p<-ggplot() +

geom\_point(aes(time\_d, percent, col = rep), size = 2, dat) +

geom\_line(aes(time\_d, percent, col = rep, group = rep), alpha = 0.5, preds2) +

facet\_wrap(~ mcpa, labeller = labeller(.multi\_line = FALSE)) +

scale\_color\_brewer(type='qual') +

theme\_bw(base\_size = 12) +

ylab('Cumulative 14C mineralization (%)') +

xlab('time (days)') +

theme(legend.position = c(0.9, 0.15))

p

# merge parameter values

C0s<-subset(params,term=="C0",select=c(mcpa,rep,estimate))

colnames(C0s)<-c('mcpa','rep','C0')

k1s<-subset(params,term=="k1",select=c(mcpa,rep,estimate))

colnames(k1s)<-c('mcpa','rep','k1')

pars<-merge(C0s,k1s,by=c('mcpa','rep'))

# calculate half-lifes

pars\_final <- pars %>%

mutate(t\_hl = log(2)/k1)

pars\_final<-arrange(pars\_final,mcpa,rep)

# calculate mean half-lifes and plot half-lifes vs. initial conc.

t\_hl\_mean <- pars\_final %>%

group\_by(., mcpa) %>%

summarise(HL=mean(t\_hl),HL\_sd=sd(t\_hl))

ggplot(t\_hl\_mean,aes(x=mcpa,y=HL))+

geom\_point(size=3)+

geom\_errorbar(aes(ymin=HL-HL\_sd,ymax=HL+HL\_sd),width=0.2)+

xlab('Initial MCPA concentration (?g kg-1)')+

ylab('Half-life (d)')

return(list("p"=p,"t\_hl\_mean"=t\_hl\_mean,"pars"=params))

}

f\_without\_2nd\_pool<-function(dat){

# model fit

# fit over each set of groupings

fits <- dat %>%

group\_by(., mcpa, rep) %>%

nest() %>%

mutate(fit = purrr::map(data, ~ nls\_multstart(percent ~ C0-C0/((1-f\_k)\*exp(k1\*time\_d)+f\_k),

data = .x,

iter = 1000,

start\_lower = c(C0=0, f\_k=0, k1=0),

start\_upper = c(C0=60, f\_k=0.999, k1=10),

supp\_errors = 'Y',

na.action = na.omit,

lower = c(C0=0, f\_k=0, k1=0),

upper = c(C0=60, f\_k=0.999, k1=10))))

# get summary

info <- fits %>%

mutate(summary = map(fit, glance)) %>%

unnest(summary)

# get params

params <- fits %>%

mutate(., p = map(fit, tidy)) %>%

unnest(p)

# get confidence intervals

CI <- fits %>%

mutate(., cis = map(fit, confint2),

cis = map(cis, data.frame)) %>%

unnest(cis) %>%

rename(., conf.low = X2.5.., conf.high = X97.5..) %>%

group\_by(., mcpa,rep) %>%

mutate(., term = c('C0', 'k1','f\_k')) %>%

ungroup() %>%

select(., -data, -fit)

# merge parameters and CI estimates

params <- merge(params, CI, by = intersect(names(params), names(CI)))

params

# new data frame of predictions

new\_preds <- dat %>%

do(., data.frame(time\_d = seq(0, max(.$time\_d), length.out = 300), stringsAsFactors = FALSE))

# create new predictions

preds2 <- fits %>%

mutate(., p = map(fit, augment)) %>%

unnest(p)

# plot

p<-ggplot() +

geom\_point(aes(time\_d, percent, col = rep), size = 2, dat) +

geom\_line(aes(time\_d, percent, col = rep, group = rep), alpha = 0.5, preds2) +

facet\_wrap(~ mcpa, labeller = labeller(.multi\_line = FALSE)) +

scale\_color\_brewer(type='qual') +

theme\_bw(base\_size = 12) +

ylab('Cumulative 14C mineralization (%)') +

xlab('time (days)') +

theme(legend.position = c(0.9, 0.15))

p

# merge parameter values

C0s<-subset(params,term=="C0",select=c(mcpa,rep,estimate))

colnames(C0s)<-c('mcpa','rep','C0')

f\_ks<-subset(params,term=="f\_k",select=c(mcpa,rep,estimate))

colnames(f\_ks)<-c('mcpa','rep','f\_k')

k1s<-subset(params,term=="k1",select=c(mcpa,rep,estimate))

colnames(k1s)<-c('mcpa','rep','k1')

pars<-merge(C0s,k1s,by=c('mcpa','rep'))

pars<-merge(pars,f\_ks,by=c('mcpa','rep'))

# calculate half-lifes

pars\_final <- pars %>%

mutate(t\_hl = 1/k1\*log(1/(1-f\_k)+1))

pars\_final<-arrange(pars\_final,mcpa,rep)

# calculate mean half-lifes and plot half-lifes vs. initial conc.

t\_hl\_mean <- pars\_final %>%

group\_by(., mcpa) %>%

summarise(HL=mean(t\_hl),HL\_sd=sd(t\_hl))

return(list("p"=p,"t\_hl\_mean"=t\_hl\_mean,"pars"=params))

}

# FULL model

# select only data of experiments with 30-500 ?g kg-1 MCPA

dat1 <-subset(dat,mcpa %in% c(30,50,100,500))

out\_list<-f\_full(dat1)

out\_list$p

t\_hl\_mean<-out\_list$t\_hl\_mean

pars<-out\_list$pars

# SIMPLIFICATION to first-order kinetics

# select only data of experiments with 1000 and 5000 ?g kg-1 MCPA

dat2 <-subset(dat,mcpa %in% c(1000,5000))

out\_list<-f\_1st\_order(dat2)

out\_list$p

t\_hl\_mean<-rbind(t\_hl\_mean,out\_list$t\_hl\_mean)

pars<-rbind(pars,out\_list$pars)

# SIMPLIFICATION by neglecting second 14C pool (mineralizable 14C-SOM)

# select only data of experiments with 20000 ?g kg-1 MCPA

dat3 <-subset(dat,mcpa %in% c(20000))

out\_list<-f\_without\_2nd\_pool(dat3)

out\_list$p

t\_hl\_mean<-rbind(t\_hl\_mean,out\_list$t\_hl\_mean)

pars<-rbind(pars,out\_list$pars)

View(pars)

ggplot(pars,aes(x=mcpa,y=estimate))+

#geom\_jitter(aes(colour=rep),size=4)+

geom\_pointrange(aes(ymin=estimate-std.error,ymax=estimate+std.error,colour=rep),

position=position\_jitter(width=0.2),size=1)+

facet\_wrap(~term)+

xlab('Initial MCPA concentration (?g kg-1)')+

ylab('Parameter value')+

theme\_bw()+

theme(axis.text=element\_text(size=16),axis.title=element\_text(size=18,face="bold"))

THEME <- theme(panel.border = element\_rect(colour = "black", fill=NA),

axis.text = element\_text(colour = 1, size = 12),

legend.background = element\_rect(linetype = 2, size = 0.5, colour = 1),

axis.title.x = element\_text(size = 14),

axis.title.y = element\_text(size = 14),

panel.grid = element\_blank(),

strip.text.x = element\_text(size = 14),

strip.text.y = element\_text(size = 14))

Halbwertszeit\_sd<-left\_join(Halbwertzeit,t\_hl\_mean, by = c('mcpa'='mcpa'))

?stat\_compare\_means

hlf<-ggplot(Halbwertszeit\_sd,aes(x=mcpa,y=emmean))+

geom\_point(size=2, alpha =.6)+

geom\_errorbar(aes(ymin=emmean-HL\_sd,ymax=emmean+HL\_sd),width=0.1)+

labs(x = expression(Initial~MCPA~concentration~(ug~kg^-1)),

y = expression(paste("Half-life (d)"))) +

theme\_bw()+

THEME

hlf

ggsave("half-life time.tiff",hlf, dpi = 300, width = 6, height = 2.5)