Sensitivity to starting point

Mark Naylor

2022-10-27

Multiple analyses of two catalogues using different initial values

The inlabru algorithm takes an initial guess on the mode of the parameters we are inverting for and iteratively updates this initial guess. There is the potential that there may be bad initial conditions such that they find different solutions and the runtime will differ depending upon how the solution converges.

In this notebook, we explore the robustness of the posteriors on two synthetic catalogues where the true parameters are known. Both catalogues contain 2000 days of data. One catalogue does not contain a large event and the second contains a M6.7 event on day 1000.

```
library(ETAS.inlabru)

library(tidyquant)
library(gridExtra)
library(grid)
library(lemon)
library(ggplot2)
library(ggpubr)
library(GGally)

library(inlabru)
library(INLA)
library(ETAS.inlabru)

# inla.setOption(pardiso.license="~/sys/licences/pardiso.lic")

library(dplyr)

# Increase num.threads if you have more cores on your computer
INLA::inla.setOption(num.threads = 2)
```

Define the parameters for the synthetic catalogues and starting values for inversion

```
# Parameters we use to generate synthetics, which we will refer to as the 'true' parameters
mu <- 0.1
K <- 0.089
alpha <- 2.29
c <- 0.11
p <- 1.08

# Format the true ETAS parameters for code to generate the synthetics
theta_etas <- data.frame(mu = mu, K = K, alpha = alpha, c = c, p = p)
```

```
# A dataframe containing different starting values for the algorithm
startingValues <- data.frame(
  mu = c(5., mu),
  K = c(1., K),
  alpha = c(5., alpha),
  c = c(0.3, c),
  p = c(1.5, p)
)
nRealisations <- length(startingValues$mu)


# Temporal duration of the synthetic catalogue in days
modelledDuration <- 1000 # [days]

# The minimum magnitude that will be modelled
M0 <- 2.5
```

Generate new catalogues and save them

```
############
#### Generate the first catalogue with no large events

samp.etas.list <- generate_temporal_ETAS_synthetic(
  theta = theta_etas,
  beta.p = log(10), M0 = M0, T1 = 0, T2 = modelledDuration, Ht = NULL
)

quiet.ETAS.cat <- bind_rows(samp.etas.list)
quiet.ETAS.cat <- quiet.ETAS.cat[order(quiet.ETAS.cat$ts), ]
# quiet.ETAS.cat <- na.omit(quiet.ETAS.cat)
```

```
############
#### Generate the second catalogue with a M6.7 event on day 1000

Ht <- data.frame(ts = c(500), magnitudes = c(6.7)) # Impose a M6.7 event on day 1000

samp.etas.list <- generate_temporal_ETAS_synthetic(theta = theta_etas, beta.p = log(10), M0 = M0, T1 = 

M6p7.ETAS.cat <- bind_rows(samp.etas.list)
M6p7.ETAS.cat <- M6p7.ETAS.cat[order(M6p7.ETAS.cat$ts), ]
```

In this vignette we use Rmd caching, but you can save the catalogues in files explicitly:

```
# save(M6p7.ETAS.cat,file="M6p7_ETAS_cat.Rda")
# save(quiet.ETAS.cat,file="quiet.ETAS.cat.Rda")
```

Present the catalogues

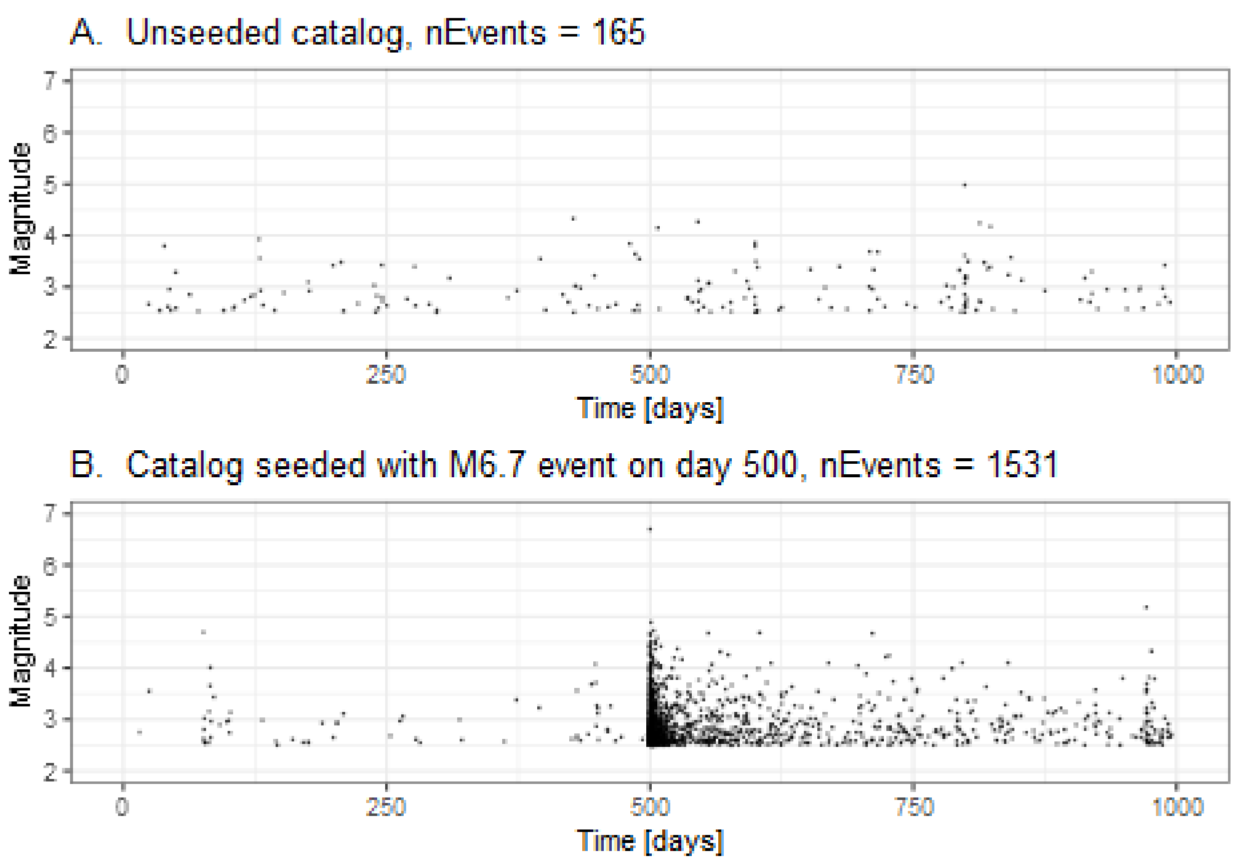
Load the catalogues

To read from saved object files:

```
# load("M6p7_ETAS_cat.Rda")
# load("quiet.ETAS.cat.Rda")
```

Plot properties of the catalogues

```
plots <- list()

plots[[1]] <- ggplot() +
  geom_point(data = quiet.ETAS.cat, aes(x = ts, y = magnitudes), size = 0.1, alpha = 0.5) +
  xlim(0, modelledDuration) +
  ggtitle(paste("A.  Unseeded catalog, nEvents =", length(quiet.ETAS.cat$ts))) +
  ylim(2, 7) +
  xlab("Time [days]") +
  ylab("Magnitude") +
  theme_bw()

plots[[2]] <- ggplot() +
  geom_point(data = M6p7.ETAS.cat, aes(x = ts, y = magnitudes), size = 0.1, alpha = 0.5) +
  xlim(0, modelledDuration) +
  ggtitle(paste("B.  Catalog seeded with M6.7 event on day 500, nEvents =", length(M6p7.ETAS.cat$ts))) +
  ylim(2, 7) +
  xlab("Time [days]") +
  ylab("Magnitude") +
  theme_bw()


plt <- grid.arrange(plots[[1]], plots[[2]], ncol = 1, nrow = 2)
```



```
# ggsave("initialConditionCats.png", plt)
# ggsave("initialConditionCats.pdf", plt)
```

Analyse the sensitivity to starting conditions

Analysis of quiet catalogue

```
list.output.quietScenario <- list()

for (i in seq_len(nRealisations)) {
  if (exists("list.input")) remove("list.input")

  # Load a set of parameters that we will need to tweak for this application
  fpath <- system.file("extdata", "user_input_synthetic_noCatalogue.txt", package = "ETAS.inlabru")
  list.input <- create_input_list_temporal_noCatalogue(fpath)

  ####################
  # Tweak the variables loaded from the input file
  list.input$M0 <- M0
  list.input$time.int <- c(0, modelledDuration)
  list.input$T12 <- c(0, modelledDuration)

  # Change the starting location, measured on the ETAS scale
  list.input$mu.init <- startingValues$mu[i]
  list.input$alpha.init <- startingValues$alpha[i]
  list.input$K.init <- startingValues$K[i]
  list.input$c.init <- startingValues$c[i]
  list.input$p.init <- startingValues$p[i]

  link.f <- list(
    mu = \(x) gamma_t(x, a_mu, b_mu),
    K = \(x) loggaus_t(x, a_K, b_K),
    alpha = \(x) unif_t(x, a_alpha, b_alpha),
    c_ = \(x) unif_t(x, a_c, b_c),
    p = \(x) unif_t(x, a_p, b_p)
  )

  # initial value - convert from ETAS scale to internal scale
  list.input$th.init <- list(
    th.mu = inv_gamma_t(list.input$mu.init, list.input$a_mu, list.input$b_mu),
    th.K = inv_loggaus_t(list.input$K.init, list.input$a_K, list.input$b_K),
    th.alpha = inv_unif_t(list.input$alpha.init, list.input$a_alpha, list.input$b_alpha),
    th.c = inv_unif_t(list.input$c.init, list.input$a_c, list.input$b_c),
    th.p = inv_unif_t(list.input$p.init, list.input$a_p, list.input$b_p)
  )

  # Define options for inlabru
  if (is.null(list.input$max_step)) {
    list.input$bru.opt.list <- list(
      bru_verbose = 0, # type of visual output
      bru_max_iter = list.input$max_iter, # maximum number of iterations
      # bru_method = list(max_step = 0.5),
```

```
      bru_initial = list.input$th.init
    ) # parameters initial values
  } else {
    list.input$bru.opt.list <- list(
      bru_verbose = 0, # type of visual output
      bru_max_iter = list.input$max_iter, # maximum number of iterations
      bru_method = list(max_step = list.input$max_step),
      bru_initial = list.input$th.init
    ) # parameters initial values
  }

  ## Add out catalogue to the input list
  list.input$catalog <- data.frame(
    time_diff = quiet.ETAS.cat$ts,
    magnitudes = quiet.ETAS.cat$magnitudes
  )

  ## Add the catalogue formatted for bru
  list.input$catalog.bru <- data.frame(
    ts = quiet.ETAS.cat$ts,
    magnitudes = quiet.ETAS.cat$magnitudes,
    idx.p = seq_len(nrow(quiet.ETAS.cat))
  )

  ## Input list is now formatted
  ####################

  ## Run the model according to the input list
  ETAS.model.fit <- Temporal.ETAS.fit(list.input)

  ## Small bit of post processing
  list.output.quietScenario[[i]] <- append(list.input, list(model.fit = ETAS.model.fit))
  list.output.quietScenario[[i]]$runtime <- sum(list.output.quietScenario[[i]]$model.fit$bru_timings$Ti
  list.output.quietScenario[[i]]$nEvents <- length(list.output.quietScenario[[i]]$catalog[, 1])
}
#> Start model fitting
#> Start creating grid...
#> Finished creating grid, time  0.424727
#> Finish model fitting
#> Start model fitting
#> Start creating grid...
#> Finished creating grid, time  0.582531
#> Finish model fitting
```

Analysis of M6.7 catalogue

```
list.output.M6p7Scenario <- list()

for (i in seq_len(nRealisations)) {
  if (exists("list.input")) {
    remove(list.input)
  }
```

```r
# Load a set of parameters that we will need to tweak for this application
fpath <- system.file("extdata", "user_input_synthetic_noCatalogue.txt", package = "ETAS.inlabru")
list.input <- create_input_list_temporal_noCatalogue(fpath)

####################
# Tweak the variables laoded from the input file
list.input$M0 <- M0
list.input$time.int <- c(0, modelledDuration)
list.input$T12 <- c(0, modelledDuration)

# Change the starting location, measured on the ETAS scale
list.input$mu.init <- startingValues$mu[i]
list.input$alpha.init <- startingValues$alpha[i]
list.input$K.init <- startingValues$K[i]
list.input$c.init <- startingValues$c[i]
list.input$p.init <- startingValues$p[i]

link.f <- list(
  mu = \(x) gamma_t(x, a_mu, b_mu),
  K = \(x) loggaus_t(x, a_K, b_K),
  alpha = \(x) unif_t(x, a_alpha, b_alpha),
  c_ = \(x) unif_t(x, a_c, b_c),
  p = \(x) unif_t(x, a_p, b_p)
)

# initial value - convert from ETAS scale to internal scale
list.input$th.init <- list(
  th.mu = inv_gamma_t(list.input$mu.init, list.input$a_mu, list.input$b_mu),
  th.K = inv_loggaus_t(list.input$K.init, list.input$a_K, list.input$b_K),
  th.alpha = inv_unif_t(list.input$alpha.init, list.input$a_alpha, list.input$b_alpha),
  th.c = inv_unif_t(list.input$c.init, list.input$a_c, list.input$b_c),
  th.p = inv_unif_t(list.input$p.init, list.input$a_p, list.input$b_p)
)

# Define options for inlabru
if (is.null(list.input$max_step)) {
  list.input$bru.opt.list <- list(
    bru_verbose = 3, # type of visual output
    bru_max_iter = list.input$max_iter, # maximum number of iterations
    # bru_method = list(max_step = 0.5),
    bru_initial = list.input$th.init
  ) # parameters initial values
} else {
  list.input$bru.opt.list <- list(
    bru_verbose = 3, # type of visual output
    bru_max_iter = list.input$max_iter, # maximum number of iterations
    bru_method = list(max_step = list.input$max_step),
    bru_initial = list.input$th.init
  ) # parameters initial values
}

## Add out catalogue to the input list
list.input$catalog <- data.frame(
```

```
    time_diff = M6p7.ETAS.cat$ts,
    magnitudes = M6p7.ETAS.cat$magnitudes
  )

  ## Add the catalogue formatted for bru
  list.input$catalog.bru <- data.frame(
    ts = M6p7.ETAS.cat$ts,
    magnitudes = M6p7.ETAS.cat$magnitudes,
    idx.p = seq_len(nrow(M6p7.ETAS.cat))
  )

  ## Input list is now formatted
  ####################

  ## Run the model according to the input list
  ETAS.model.fit <- Temporal.ETAS.fit(list.input)

  ## Small bit of post processing
  list.output.M6p7Scenario[[i]] <- append(list.input, list(model.fit = ETAS.model.fit))
  list.output.M6p7Scenario[[i]]$runtime <- sum(list.output.M6p7Scenario[[i]]$model.fit$bru_timings$Time
  list.output.M6p7Scenario[[i]]$nEvents <- length(list.output.M6p7Scenario[[i]]$catalog[, 1])
}
#> Start model fitting
#> Start creating grid...
#> Finished creating grid, time  4.135106
#> iinla: Evaluate component inputs
#> iinla: Evaluate component linearisations
#> iinla: Evaluate component simplifications
#> iinla: Evaluate predictor linearisation
#> iinla: Construct inla stack
#> iinla: Model initialisation completed
#> iinla: Iteration 1 [max:100]
#> iinla: Step rescaling: 91.83%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Iteration 2 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 128.5%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 2080% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 3 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 1430% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 4 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 111.1%, Optimisation
#> iinla: Evaluate component linearisations
```

```
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 747% of SD, and line search is active [stop if: <10% and line se
#> iinla: Iteration 5 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.9%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 384% of SD, and line search is active [stop if: <10% and line se
#> iinla: Iteration 6 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 102.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 167% of SD, and line search is active [stop if: <10% and line se
#> iinla: Iteration 7 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 101.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 143% of SD, and line search is active [stop if: <10% and line se
#> iinla: Iteration 8 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.8%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 123% of SD, and line search is active [stop if: <10% and line se
#> iinla: Iteration 9 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.5%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 108% of SD, and line search is active [stop if: <10% and line se
#> iinla: Iteration 10 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.4%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 96.1% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 11 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.3%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 86% of SD, and line search is active [stop if: <10% and line sea
#> iinla: Iteration 12 [max:100]
#> iinla: Step rescaling: 162%, Expand
```

```
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.2%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 77.3% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 13 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.2%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 69.6% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 14 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.2%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 62.8% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 15 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 56.8% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 16 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 51.4% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 17 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 46.5% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 18 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 42.1% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 19 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
```

```
#> iinla: Max deviation from previous: 38.1% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 20 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 34.5% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 21 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 31.3% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 22 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 28.4% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 23 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 25.7% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 24 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.1%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 23.3% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 25 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 21.2% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 26 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 19.2% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 27 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
```

```
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 17.4% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 28 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 15.8% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 29 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 14.4% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 30 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 13% of SD, and line search is inactive [stop if: <10% and line s
#> iinla: Iteration 31 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 11.8% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 32 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 10.7% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 33 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 9.76% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Convergence criterion met, running final INLA integration with known theta mode.
#> iinla: Iteration 34 [max:100]
#> Finish model fitting
#> Start model fitting
#> Start creating grid...
#> Finished creating grid, time  3.844827
#> iinla: Evaluate component inputs
```

```
#> iinla: Evaluate component linearisations
#> iinla: Evaluate component simplifications
#> iinla: Evaluate predictor linearisation
#> iinla: Construct inla stack
#> iinla: Model initialisation completed
#> iinla: Iteration 1 [max:100]
#> iinla: Step rescaling: 90.7%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Iteration 2 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100.3%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 42.2% of SD, and line search is active [stop if: <10% and line s
#> iinla: Iteration 3 [max:100]
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 38.1% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 4 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 35% of SD, and line search is inactive [stop if: <10% and line s
#> iinla: Iteration 5 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 31.7% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 6 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 28.7% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 7 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 25.9% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 8 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
```

```
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 23.5% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 9 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 21.3% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 10 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 19.3% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 11 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 17.5% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 12 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 15.9% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 13 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 14.4% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 14 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 13.1% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 15 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 11.9% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 16 [max:100]
```

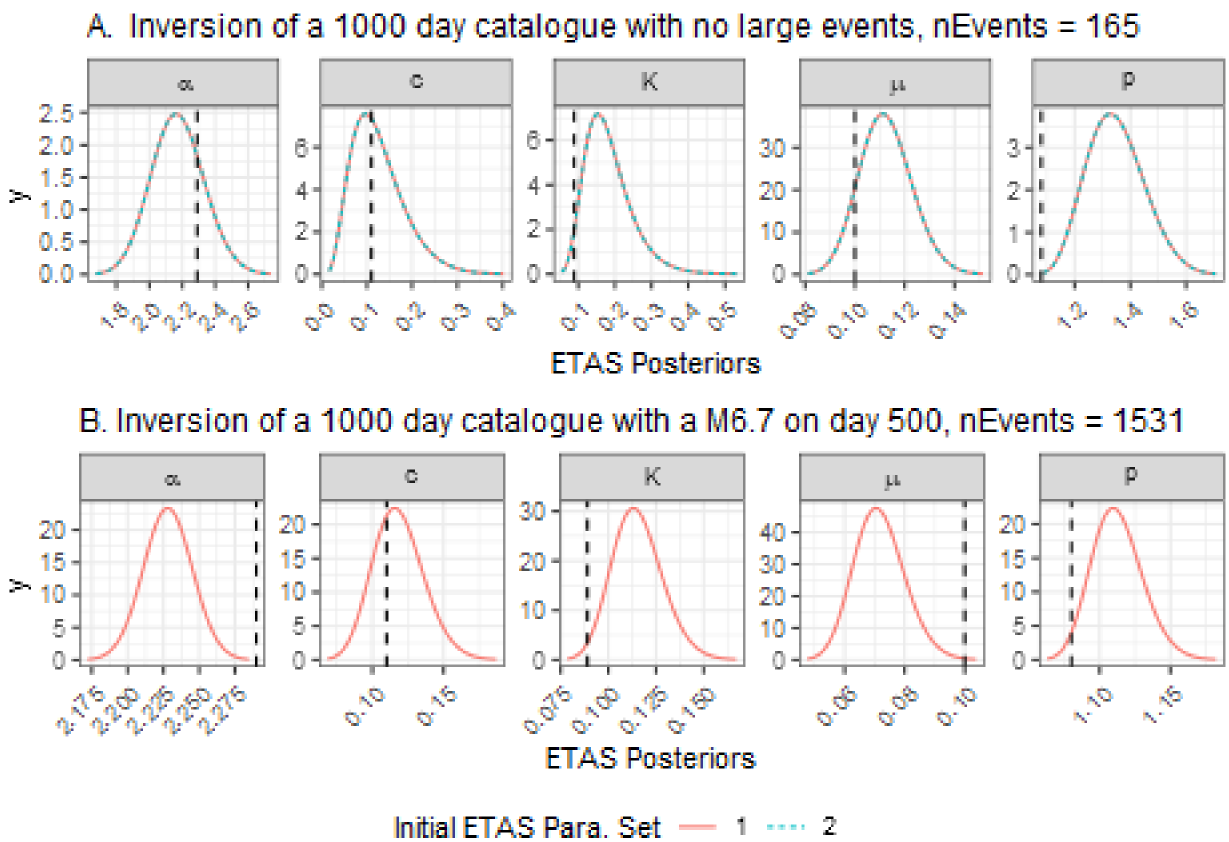
```
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 10.8% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Iteration 17 [max:100]
#> iinla: Step rescaling: 162%, Expand
#> iinla: Step rescaling: 100%, Overstep
#> iinla: Step rescaling: 100%, Optimisation
#> iinla: Evaluate component linearisations
#> iinla: Evaluate predictor linearisation
#> iinla: Max deviation from previous: 9.79% of SD, and line search is inactive [stop if: <10% and line
#> iinla: Convergence criterion met, running final INLA integration with known theta mode.
#> iinla: Iteration 18 [max:100]
#> Finish model fitting
```

Plot posteriors with corresponding starting values

```
plots <- list()

trueParas <- data.frame(value = c(mu, K, alpha, c, p), param = c("mu", "K", "alpha", "c", "p"))

post.list <- get_posterior_param(input.list = list.output.quietScenario[[1]])
post.df <- post.list[[1]]
post.df$id <- 1

for (i in 2:nRealisations) {
  post.list <- get_posterior_param(input.list = list.output.quietScenario[[i]])
  post.df.tmp <- post.list[[1]]
  post.df.tmp$id <- i

  post.df <- rbind(post.df, post.df.tmp)
}

plots[[1]] <- ggplot(post.df, aes(x = x, y = y, group = id, color = factor(id), lty = factor(id))) +
  geom_line() +
  # scale_x_discrete(guide = guide_axis(check.overlap = TRUE)) +
  facet_wrap(facets = vars(param), scales = "free", labeller = label_parsed, nrow = 1) +
  geom_vline(aes(xintercept = value),
    data = trueParas, color = "black", linetype = 2,
    label = "True value"
  ) +
  labs(color = "Initial ETAS Para. Set", linetype = "Initial ETAS Para. Set") +
  ggtitle(paste("A.  Inversion of a 1000 day catalogue with no large events, nEvents =", length(quiet.E
  xlab("ETAS Posteriors") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.position = "hidden") +
  theme(plot.title = element_text(size = 12))
#> Warning in geom_vline(aes(xintercept = value), data = trueParas, color =
#> "black", : Ignoring unknown parameters: `label`
```

```r
trueParas <- data.frame(value = c(mu, K, alpha, c, p), param = c("mu", "K", "alpha", "c", "p"))

post.list <- get_posterior_param(input.list = list.output.M6p7Scenario[[1]])
post.df <- post.list[[1]]
post.df$id <- 2

for (i in 2:nRealisations) {
  post.list <- get_posterior_param(input.list = list.output.M6p7Scenario[[i]])
  post.df.tmp <- post.list[[1]]
  post.df.tmp$id <- i

  post.df <- rbind(post.df, post.df.tmp)
}

plots[[2]] <- ggplot(post.df, aes(x = x, y = y, group = id, color = factor(id), lty = factor(id))) +
  geom_line() +
  # scale_x_discrete(guide = guide_axis(check.overlap = TRUE)) +
  facet_wrap(facets = vars(param), scales = "free", labeller = label_parsed, nrow = 1) +
  geom_vline(aes(xintercept = value),
    data = trueParas, color = "black", linetype = 2,
    label = "True value"
  ) +
  labs(color = "Initial ETAS Para. Set", linetype = "Initial ETAS Para. Set") +
  ggtitle(paste("B. Inversion of a 1000 day catalogue with a M6.7 on day 500, nEvents =", length(M6p7.ET
  xlab("ETAS Posteriors") +
  theme_bw() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  theme(legend.position = "hidden") +
  theme(plot.title = element_text(size = 12))
#> Warning in geom_vline(aes(xintercept = value), data = trueParas, color =
#> "black", : Ignoring unknown parameters: `label`
```

```r
plt <- grid_arrange_shared_legend(plots[[1]], plots[[2]], ncol = 1, nrow = 2, position = "bottom")
```

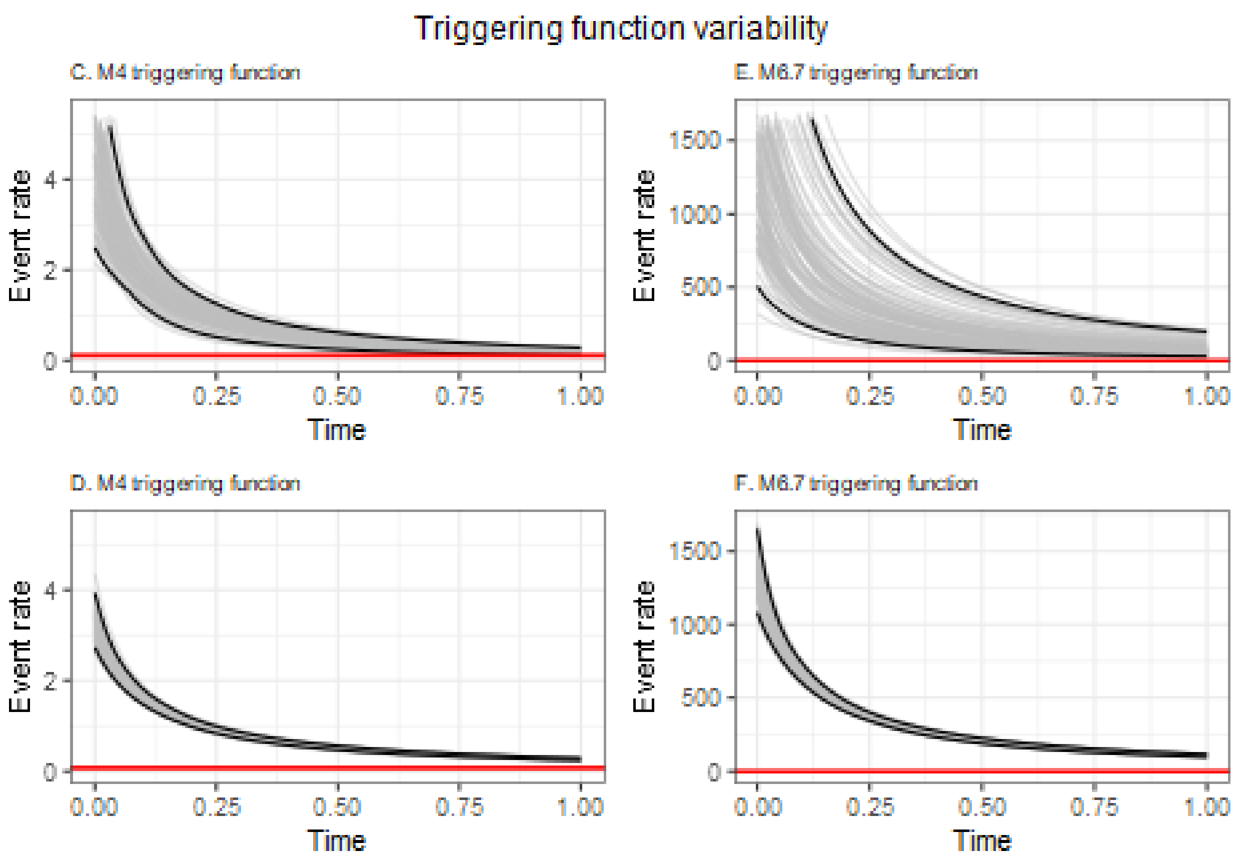



```
plt
#> TableGrob (2 x 1) "arrange": 2 grobs
#>   z     cells    name              grob
#> 1 1 (1-1,1-1) arrange   gtable[arrange]
#> 2 2 (2-2,1-1) arrange gtable[guide-box]
# ggsave("initialCondition_posteriors.png", plt)
# ggsave("initialCondition_posteriors.pdf", plt)
```

Explore ETAS triggering function using posterior samples

```
plot_triggering <- list()
plot_triggering[[1]] <- triggering_fun_plot(list.output.quietScenario[[1]], magnitude = 4, n.samp = 100)
  ggtitle("C. M4 triggering function") +
  theme_bw() +
  ylim(0, 5.5) +
  theme(plot.title = element_text(size = 8))

plot_triggering[[2]] <- triggering_fun_plot(list.output.M6p7Scenario[[1]], magnitude = 4, n.samp = 100)
  ggtitle("D. M4 triggering function") +
  theme_bw() +
  ylim(0, 5.5) +
  theme(plot.title = element_text(size = 8))

plot_triggering[[3]] <- triggering_fun_plot(list.output.quietScenario[[1]], magnitude = 6.7, n.samp = 1(
  ggtitle("E. M6.7 triggering function") +
  theme_bw() +
```

```
  ylim(0, 1700) +
  theme(plot.title = element_text(size = 8))


plot_triggering[[4]] <- triggering_fun_plot(list.output.M6p7Scenario[[1]], magnitude = 6.7, n.samp = 100
  ggtitle("F. M6.7 triggering function") +
  theme_bw() +
  ylim(0, 1700) +
  theme(plot.title = element_text(size = 8))


plt <- grid.arrange(plot_triggering[[1]], plot_triggering[[3]], plot_triggering[[2]], plot_triggering[[4
  theme(plot.title = element_text(size = 8))
#> Warning: Removed 49 rows containing missing values (`geom_line()`).
#> Warning: Removed 3 rows containing missing values (`geom_line()`).
#> Warning: Removed 203 rows containing missing values (`geom_line()`).
#> Warning: Removed 12 rows containing missing values (`geom_line()`).
```

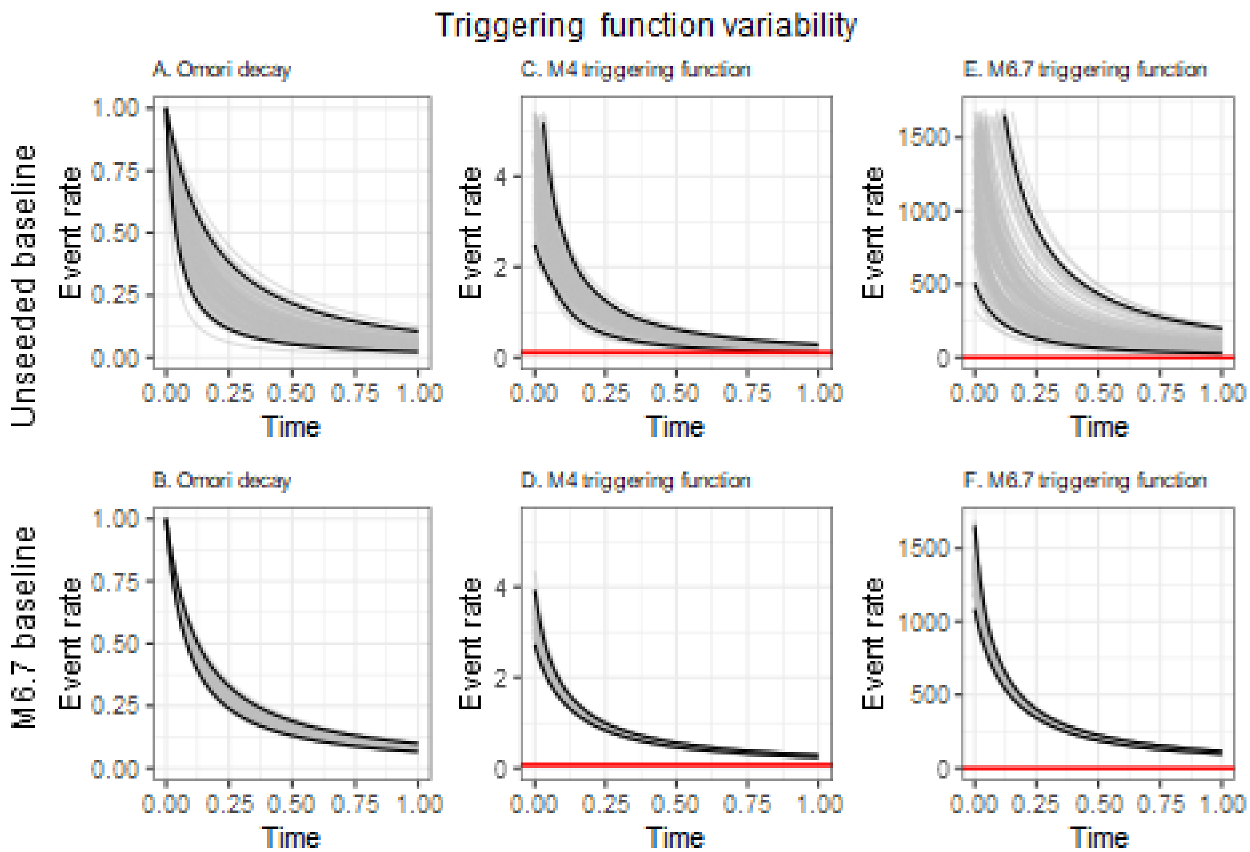



```
plt
#> NULL

# ggsave("trigFuncVariability_runin.png", plt)
# ggsave("trigFuncVariability_runin.pdf", plt)
```

```
plot_omori <- list()
plot_omori[[1]] <- omori_plot_posterior(list.output.quietScenario[[1]], n.samp = 100) +
```

```r
  ggtitle("A. Omori decay") +
  theme_bw() +
  ylim(0, 1) +
  theme(plot.title = element_text(size = 8))


plot_omori[[2]] <- omori_plot_posterior(list.output.M6p7Scenario[[1]], n.samp = 100) +
  ggtitle("B. Omori decay") +
  theme_bw() +
  ylim(0, 1) +
  theme(plot.title = element_text(size = 8))
```

```r
plt <- grid.arrange(plot_omori[[1]], plot_triggering[[1]], plot_triggering[[3]], plot_omori[[2]], plot_t
#> Warning: Removed 49 rows containing missing values (`geom_line()`).
#> Warning: Removed 3 rows containing missing values (`geom_line()`).
#> Warning: Removed 203 rows containing missing values (`geom_line()`).
#> Warning: Removed 12 rows containing missing values (`geom_line()`).
```

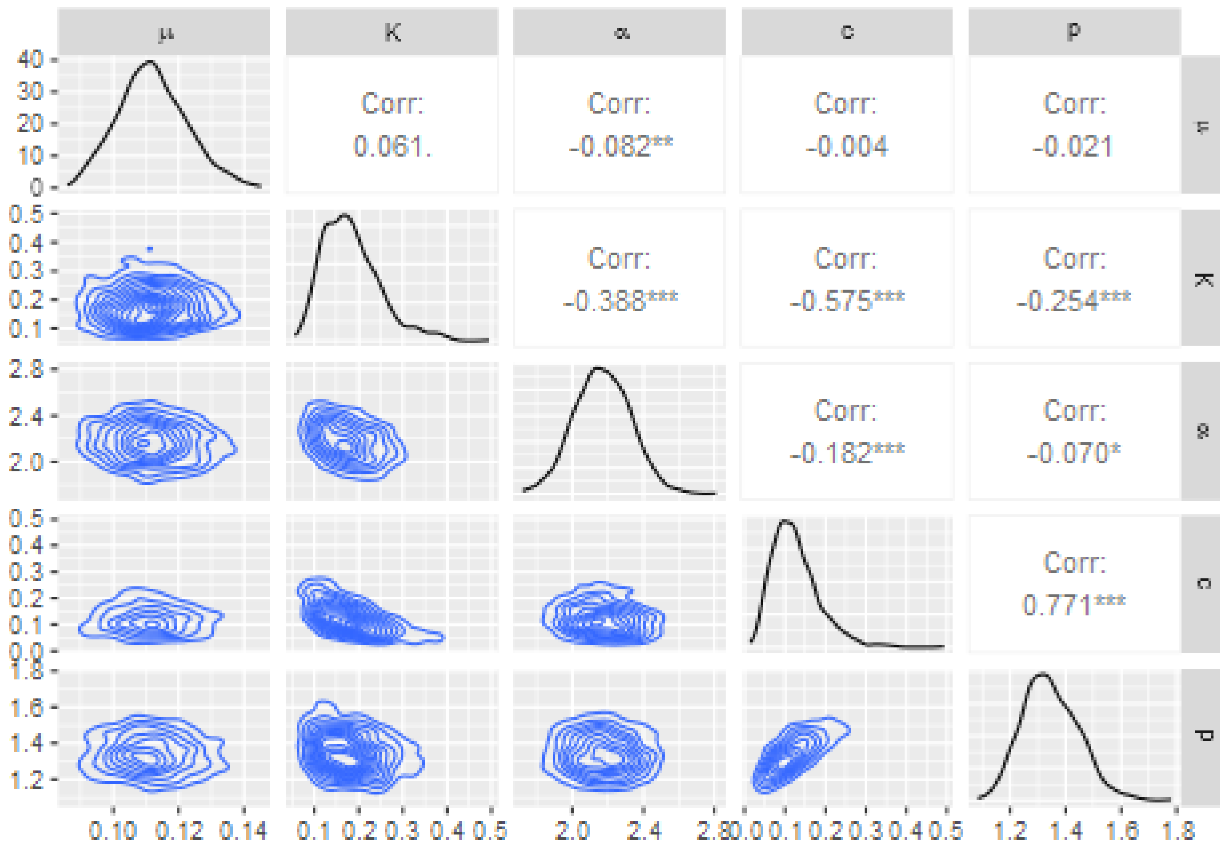



```r
plt
#> TableGrob (3 x 4) "arrange": 8 grobs
#>   z     cells    name                grob
#> 1 1 (2-2,2-2) arrange      gtable[layout]
#> 2 2 (2-2,3-3) arrange      gtable[layout]
#> 3 3 (2-2,4-4) arrange      gtable[layout]
#> 4 4 (3-3,2-2) arrange      gtable[layout]
#> 5 5 (3-3,3-3) arrange      gtable[layout]
```

```
#> 6 6 (3-3,4-4) arrange       gtable[layout]
#> 7 7 (1-1,2-4) arrange text[GRID.text.1299]
#> 8 8 (1-3,1-1) arrange text[GRID.text.1300]

# ggsave("trigFuncVariability_runin.png", plt)
# ggsave("trigFuncVariability_runin.pdf", plt)
```

Plot the samples from the joint posteriors as pairs plots

```
post_pairs_plot(list.output.quietScenario[[1]], n.samp = 1000)$pair.plot
```

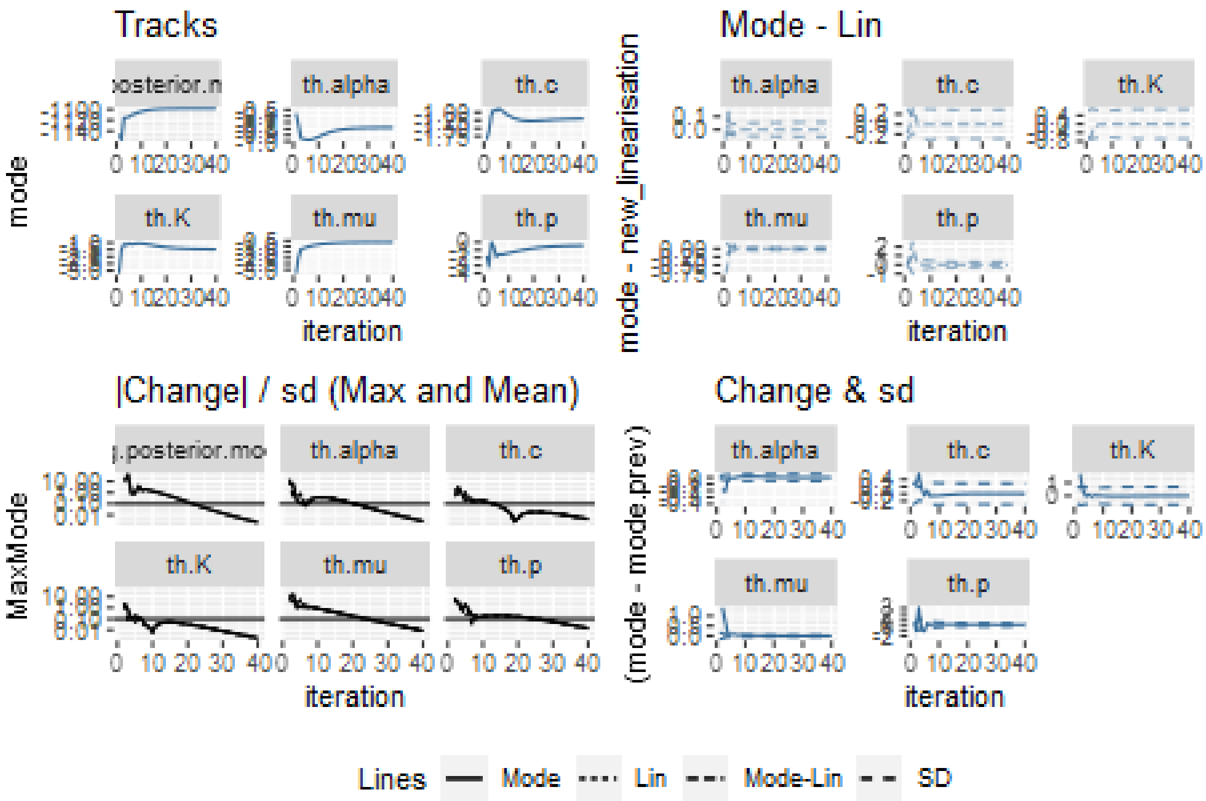



```
post_pairs_plot(list.output.M6p7Scenario[[1]], n.samp = 1000)$pair_plot
#> NULL
```

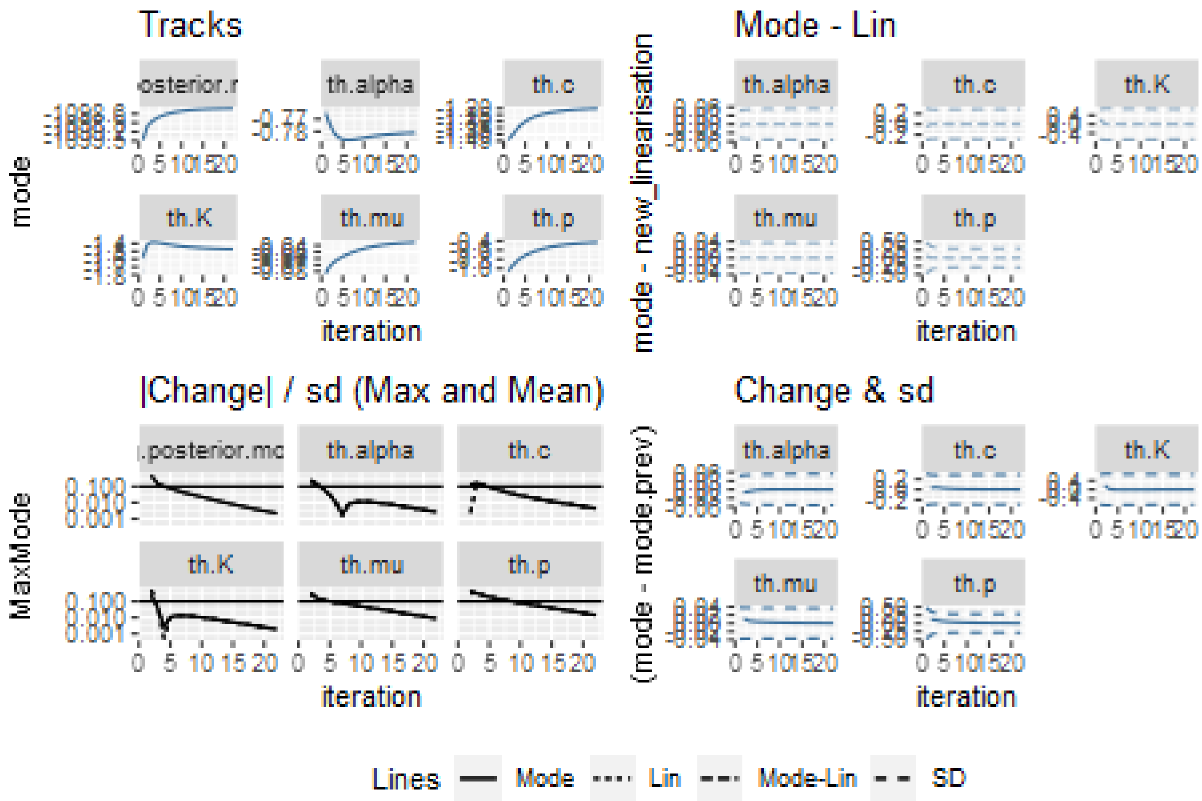
inlabru convergence diagnostics

We can also assess the convergence of the inlabru method itself, using the `bru_convergence_plot()` function. This can reveal if the starting values for the inlabru estimation are unreasonable, and better starting values for the nonlinear iterations might speed up the computations.

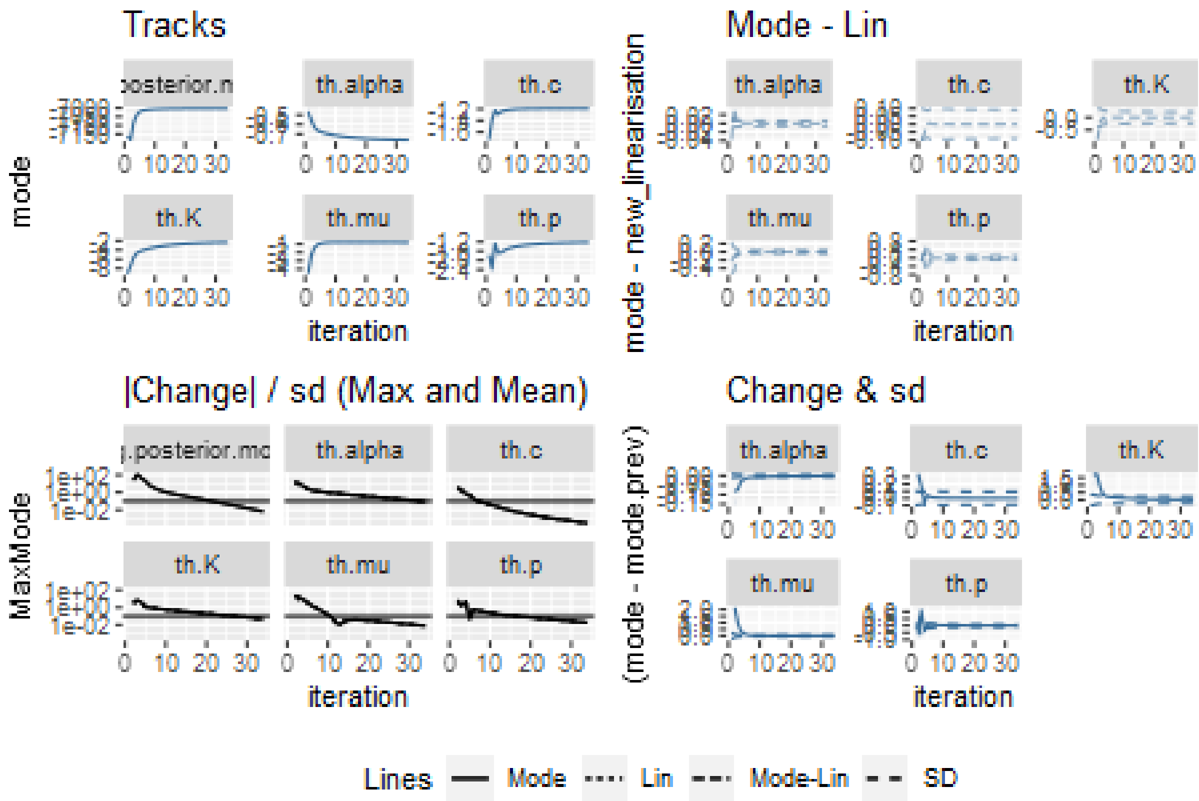
```
bru_convergence_plot(list.output.quietScenario[[1]]$model.fit)
```



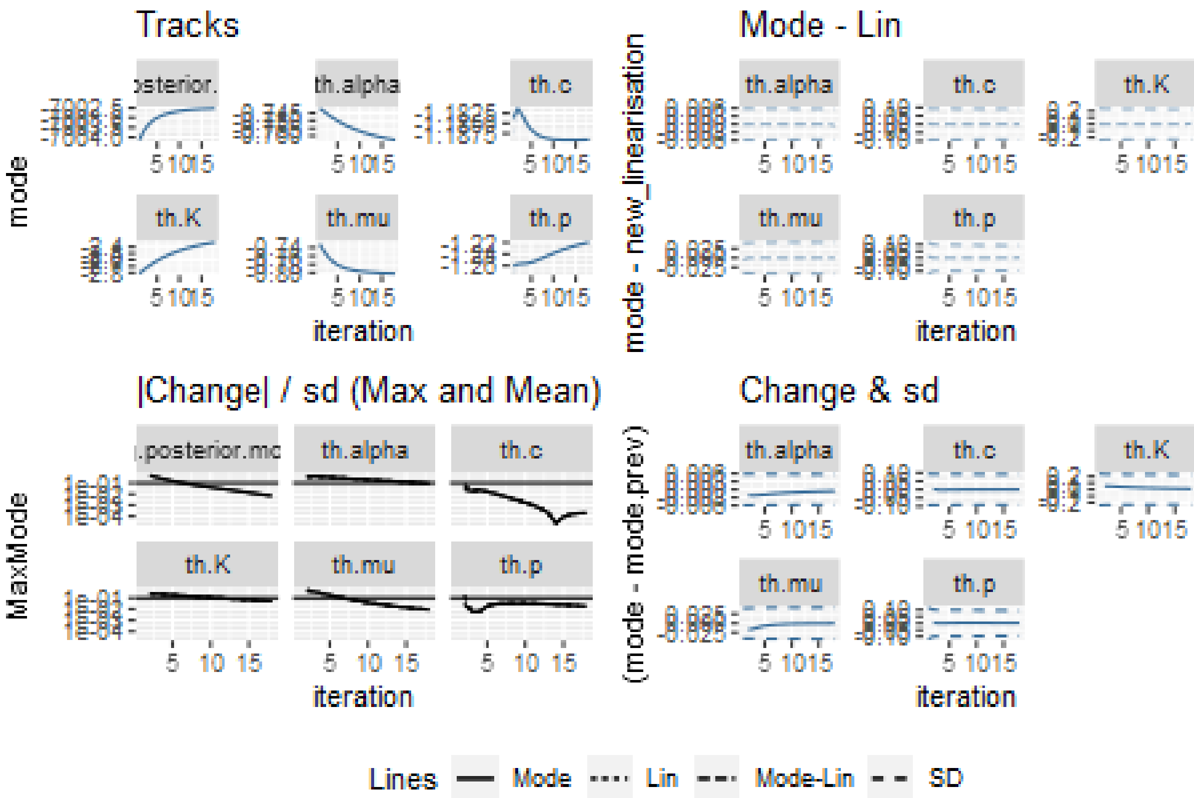

```r
bru_convergence_plot(list.output.quietScenario[[2]]$model.fit)
```




```r
bru_convergence_plot(list.output.M6p7Scenario[[1]]$model.fit)
```




```r
bru_convergence_plot(list.output.M6p7Scenario[[2]]$model.fit)
```


Tracks
mode
th.alpha
th.c
th.K
th.mu
th.p
iteration
Mode - Lin
mode - new_linearisation
th.alpha
th.c
th.K
th.mu
th.p
iteration
|Change| / sd (Max and Mean)
MaxMode
th.alpha
th.c
th.K
th.mu
th.p
iteration
Change & sd
(mode - mode.prev)
th.alpha
th.c
th.K
th.mu
th.p
iteration
Lines
Mode
Lin
Mode-Lin
SD