TDLNM

Daniel Mork

6/5/2020

Code example of TDLNM

```
library(dlmtree)
# Piecewise constant effect in exposure and time
set.seed(1)
D <- tdlnm.sim("A", 1) # try also B, C, and D!
```

Run TDLNM

- Warning: For speed, this simulation runs only a small number of iterations. We recommend ≥ 5000 burn-in with ≥ 15000 iterations thinned by 10, using 20 trees.

```
splits <- seq(min(D$exposure), max(D$exposure), length.out = 52)[-c(1,52)]
smoothing <-
res <- tdlnm(y ~ ., data = D$dat, exposure.data = as.matrix(D$exposure),
             #exposure.se = sd(D$exposure)/2, # uncomment this line to try smoothing
             exposure.splits = list("type" = "values", "split.vals" = splits),
             n.trees = 10, n.burn = 500, n.iter = 1000, n.thin = 1)
```

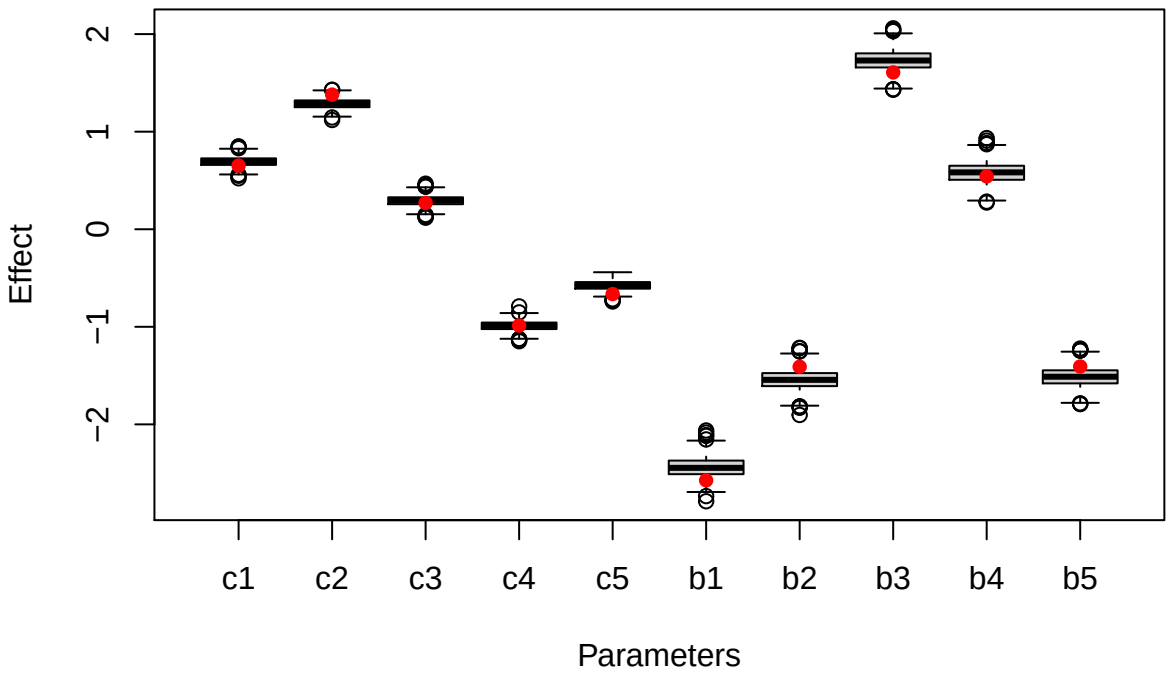
```
## Preparing data...
## Running model '.' = 100 iterations
## .....
## Burn-in time: 3.68 seconds
## Estimated time to completion: 7.36 seconds
## .........
```

```
res.sum <- summary(res,
                   pred.at = seq(min(D$exposure), max(D$exposure), length.out = 102),
                   cenval = D$cenval)
```

```
## Centered DLNM at exposure value 1
```

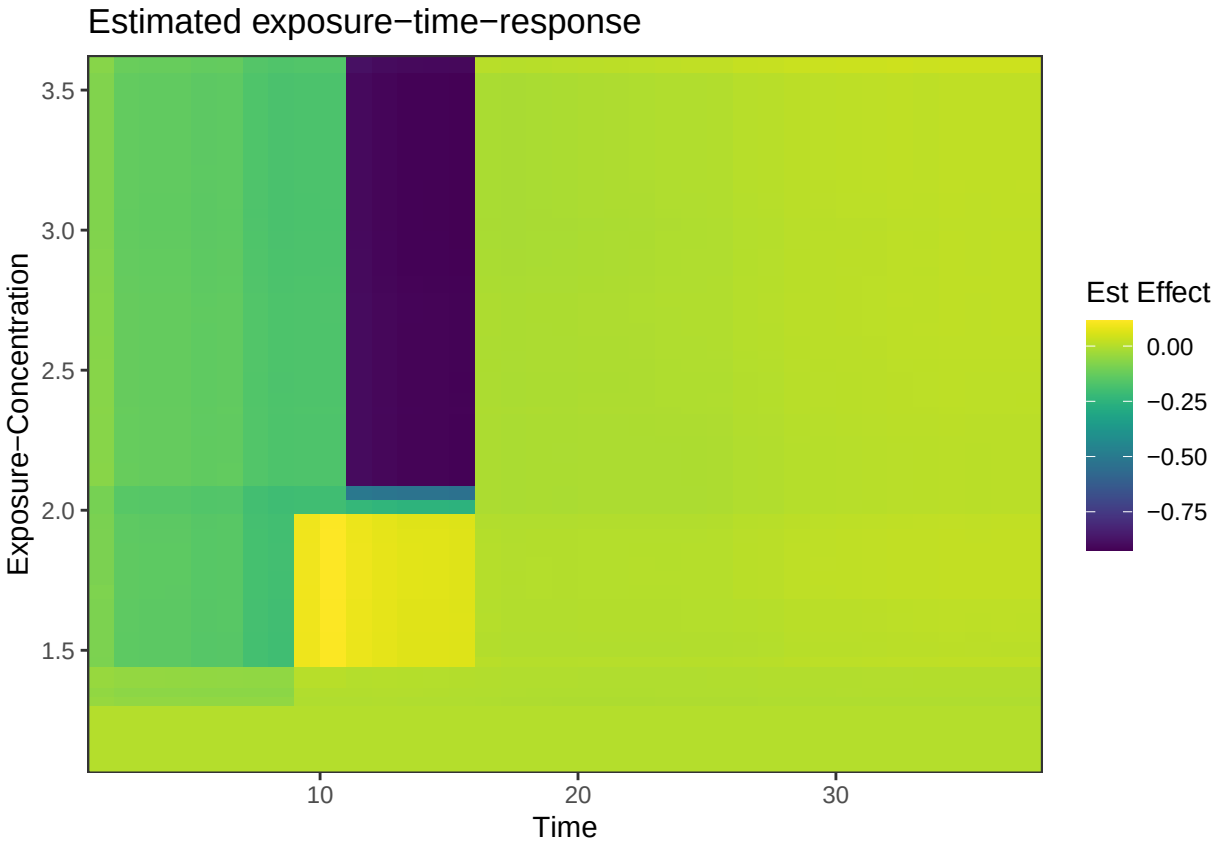
Fixed effect estimates vs. truth

```
boxplot(res$gamma[,-1], xlab = "Parameters", ylab = "Effect") # Boxplot estimated effects
points(1:10, D$params, col = "red", pch = 16) # Red dots = truth
```



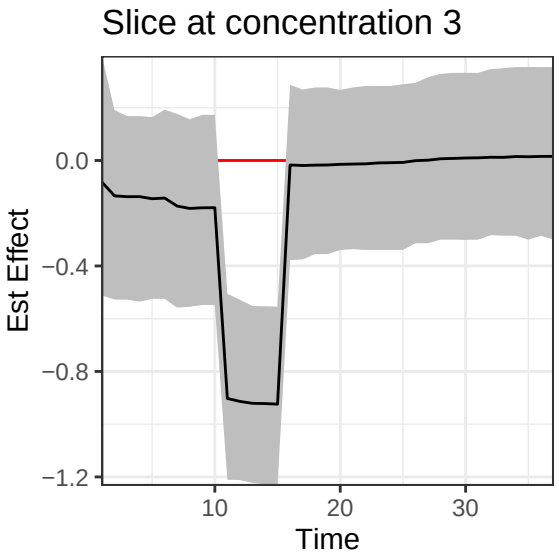
Plot of exposure-time response surface

```
plot(res.sum, main = "Estimated exposure-time-response")
```

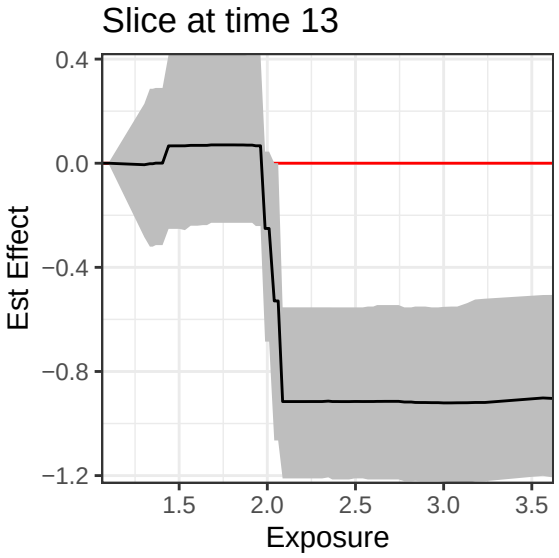

Estimated exposure−time−response
Exposure−Concentration
3.5
3.0
2.5
2.0
1.5
10
20
30
Time
Est Effect
0.00
−0.25
−0.50
−0.75

Slices of surface

```
plot(res.sum, "slice", val = 3, main = "Slice at concentration 3")
```




```
plot(res.sum, "slice", time = 13, main = "Slice at time 13")
```



Compare estimated surface to truth

```
truth <- D$dlnm.fun(sapply(1:37, function(i) res.sum$pred.vals), D$cenval, F)
# RMSE
sqrt(mean((res.sum$matfit - truth)^2))
```

```
## [1] 0.09084211
```

```
# Coverage
mean(res.sum$cilower < truth & res.sum$ciupper > truth)
```

```
## [1] 0.972271
```

```
# True positive effect classification
(length(which(res.sum$cilower > 0 & truth > 0)) +
    length(which(res.sum$ciupper < 0 & truth < 0))) /
  length(which(truth != 0))
```

```
## [1] 0.9531915
```

```
# False positive effect classification
(length(which(res.sum$cilower > 0 & truth == 0)) +
    length(which(res.sum$ciupper < 0 & truth == 0))) /
  length(which(truth == 0))
```

```
## [1] 0
```