

Functional linear models

Luciano L.M. De Benedictis

2026-01-07

```
# setup -----  
  
library(tidyverse)  
  
## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --  
## v dplyr      1.1.4      v readr      2.1.6  
## v forcats    1.0.1      v stringr   1.6.0  
## v ggplot2    4.0.1      v tibble    3.3.0  
## v lubridate  1.9.4      v tidyr     1.3.2  
## v purrr      1.2.0  
## -- Conflicts ----- tidyverse_conflicts() --  
## x dplyr::filter() masks stats::filter()  
## x dplyr::lag()    masks stats::lag()  
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors  
  
library(GET)  
library(parallel)  
library(patchwork)  
  
source("0.functions.R")  
  
RNGkind("L'Ecuyer-CMRG")  
set.seed(24695)  
  
# Initiate cluster  
cl <- makeCluster(detectCores())  
parallel::clusterSetRNGStream(cl = cl, iseed = 24695)  
  
#load data  
indexcurves <- readRDS("curves.rds")  
vars <- readRDS("selected_variables.rds")  
  
## FLM testing function -----  
  
flmtest <- function (index, formula, nsim = 19999, cl=NULL){  
  graph.flm(nsim = nsim,  
    formula.full = formula,  
    formula.reduced = Y ~ 1,  
    curve_sets = list(Y = index),  
    factors = vars,
```

```

        cl = cl)
}

## plotting functions -----

#global envelope plot

plot_FLM <- function(x, title = NULL){
  plot <- plot(x)+
    scale_x_continuous( # convert x-axis from area to length
      labels = function(x) x * 10,
      breaks = c(0.01, seq(from = 0.5, to = 2.5, by = 0.5)),
      name = expression("length of sampling units" ~ (italic(m)))
    )
  subtitle <- gsub("Graphical functional GLM: ", "", plot$labels$title)
  plot[["layers"]][[1]][["aes_params"]]$fill <- rgb(188, 223, 235, maxColorValue = 255)
  plot[["layers"]][[1]][["aes_params"]]$alpha <- 1
  plot+
    theme_minimal()+
    labs(title = title,
         subtitle = subtitle)+
    theme(plot.subtitle = element_text(face = if(as.numeric(gsub("p [[:punct:]] ", "", subtitle))>0.05)
      "plain" else "bold", size = 10),
          legend.position = "none",
          strip.text.x = element_text(size = 10))
}

#plot patchwork

patch <- function(x, ncol=2){
  wrap_plots(x, ncol = ncol, byrow = T, guides = "collect")+
  plot_layout(axis_titles = "collect")
}

# select indices from qdecomp output -----

indices <- c("E_alpha", "E_gamma", "E_beta_mult", "redundancy_a", "U_gamma_star", "clustering")

#nicer labels for plotting
labels <- list(E_alpha = expression(E[alpha]),
              E_gamma = expression(E[gamma]),
              E_beta_mult = expression(E[beta]),
              redundancy_a = expression(R[alpha]^"*"),
              U_gamma_star = expression(U[gamma]^"*"),
              clustering = "Clustering")

indexcurves <- lapply(indexcurves, function (x) x[indices])

# functional linear models -----

```

Models have been fit prior to making this report. To refit the models, uncomment the following lines and comment “load results”.

```

# #main model
# flm_FD <- lapply(indexcurves, function (x)
#   lapply(x, function (y) flmtest(y, formula = formula(Y ~ RH050 + LAI), cl = cl))
# )
#
# #only LAI
# flm_LAI <- lapply(indexcurves, function (x)
#   lapply(x, function (y) flmtest(y, formula = formula(Y ~ LAI), cl = cl))
# )
#
# #only RH050
# flm_RH <- lapply(indexcurves, function (x)
#   lapply(x, function (y) flmtest(y, formula = formula(Y ~ RH050), cl = cl))
# )
#
# #with quadratic terms
# flm_quadratic <- lapply(indexcurves, function (x)
#   lapply(x, function (y) flmtest(y,
#                                   formula = formula(Y ~ RH050 + LAI + I(RH050^2) + I(LAI^2)),
#                                   cl = cl))
# )

#save results
# saveRDS(flm_FD, "FLM.rds")
# saveRDS(flm_LAI, "FLM_LAI.rds")
# saveRDS(flm_RH, "FLM_RH050.rds")
# saveRDS(flm_quadratic, "FLM_quadratic.rds")

#load results
flm_FD <- readRDS("FLM.rds")
flm_LAI <- readRDS("FLM_LAI.rds")
flm_RH <- readRDS("FLM_RH050.rds")
flm_quadratic <- readRDS("FLM_quadratic.rds")

# plotting results -----

## main model----

plots <- lapply(flm_FD, function (x)
  imap(x, ~ plot_FLM(.x, title = labels[.y])))
)

```

```

## Warning: `aes_string()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`.
## i See also `vignette("ggplot2-in-packages")` for more information.
## i The deprecated feature was likely used in the GET package.
##   Please report the issue at <https://github.com/myllym/GET/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.

```

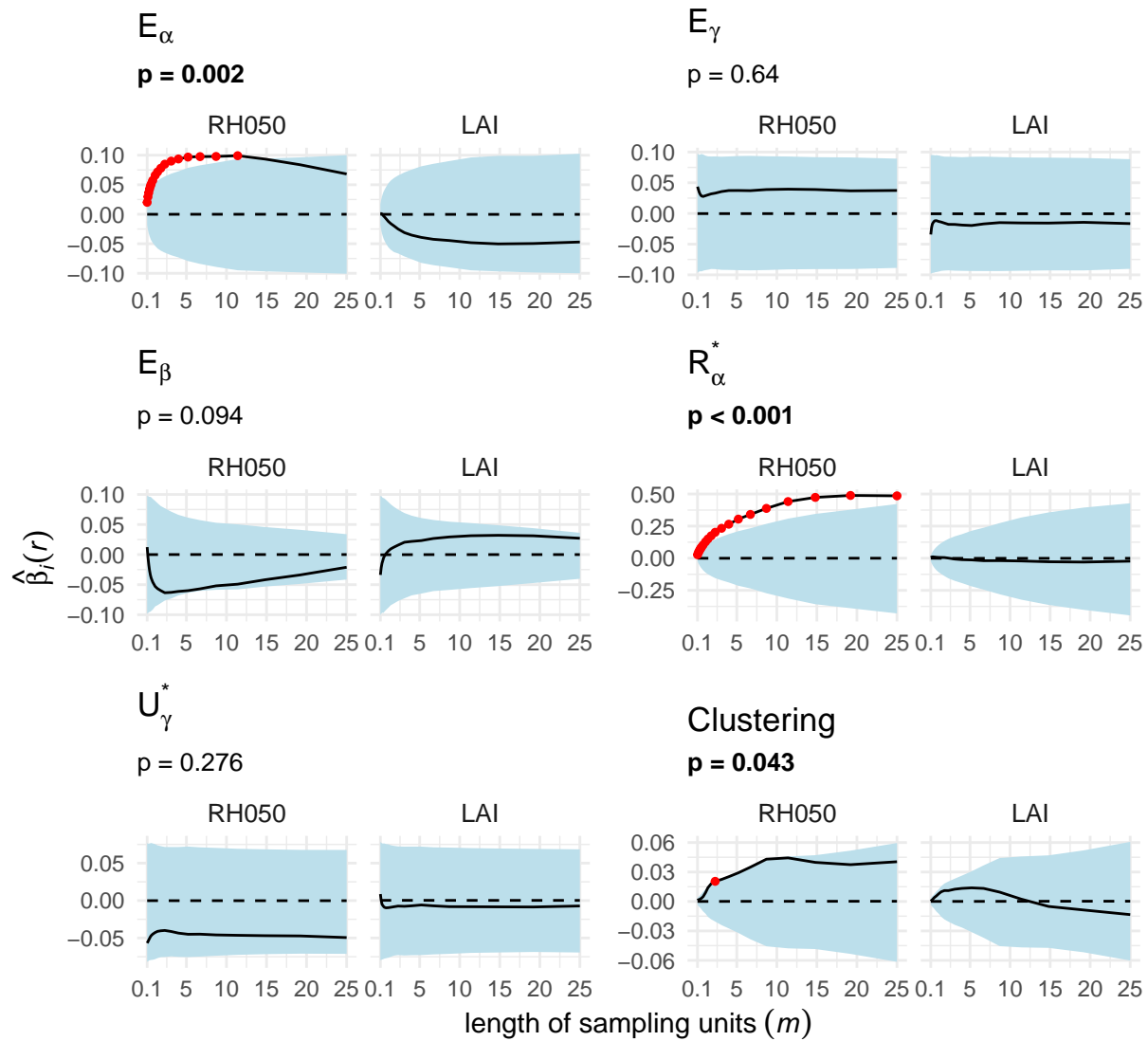
```

## Warning: `aes_()` was deprecated in ggplot2 3.0.0.
## i Please use tidy evaluation idioms with `aes()`

```

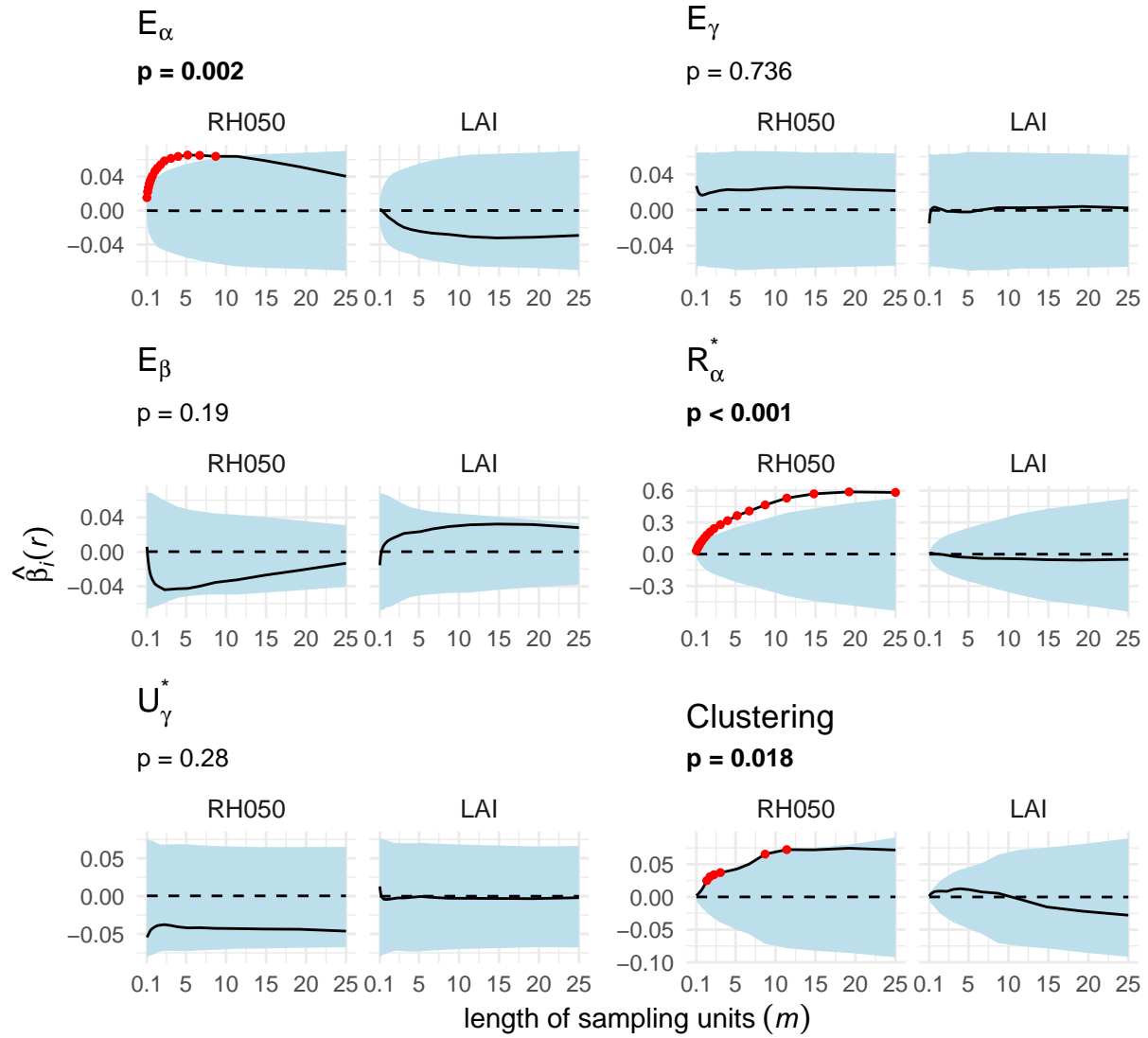
```
## i The deprecated feature was likely used in the GET package.
## Please report the issue at <https://github.com/myllym/GET/issues>.
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
patch(plots$overall)
```



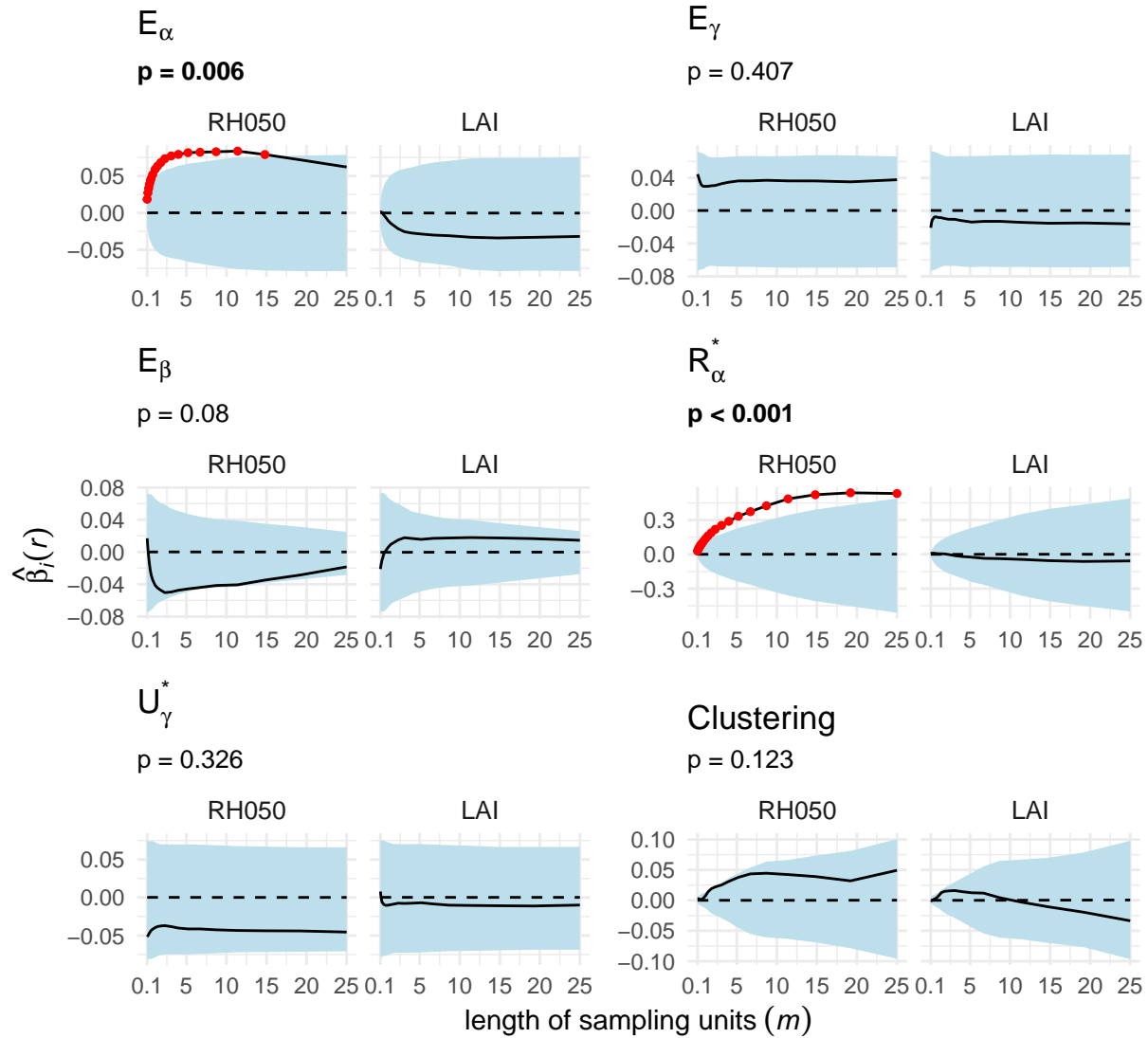
```
ggsave('plots/overall.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)
ggsave('plots/FIG3.eps', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)
```

```
patch(plots$aboveground)
```



```
ggsave('plots/above.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)
ggsave('plots/FIG4.eps', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots$clonal)
```

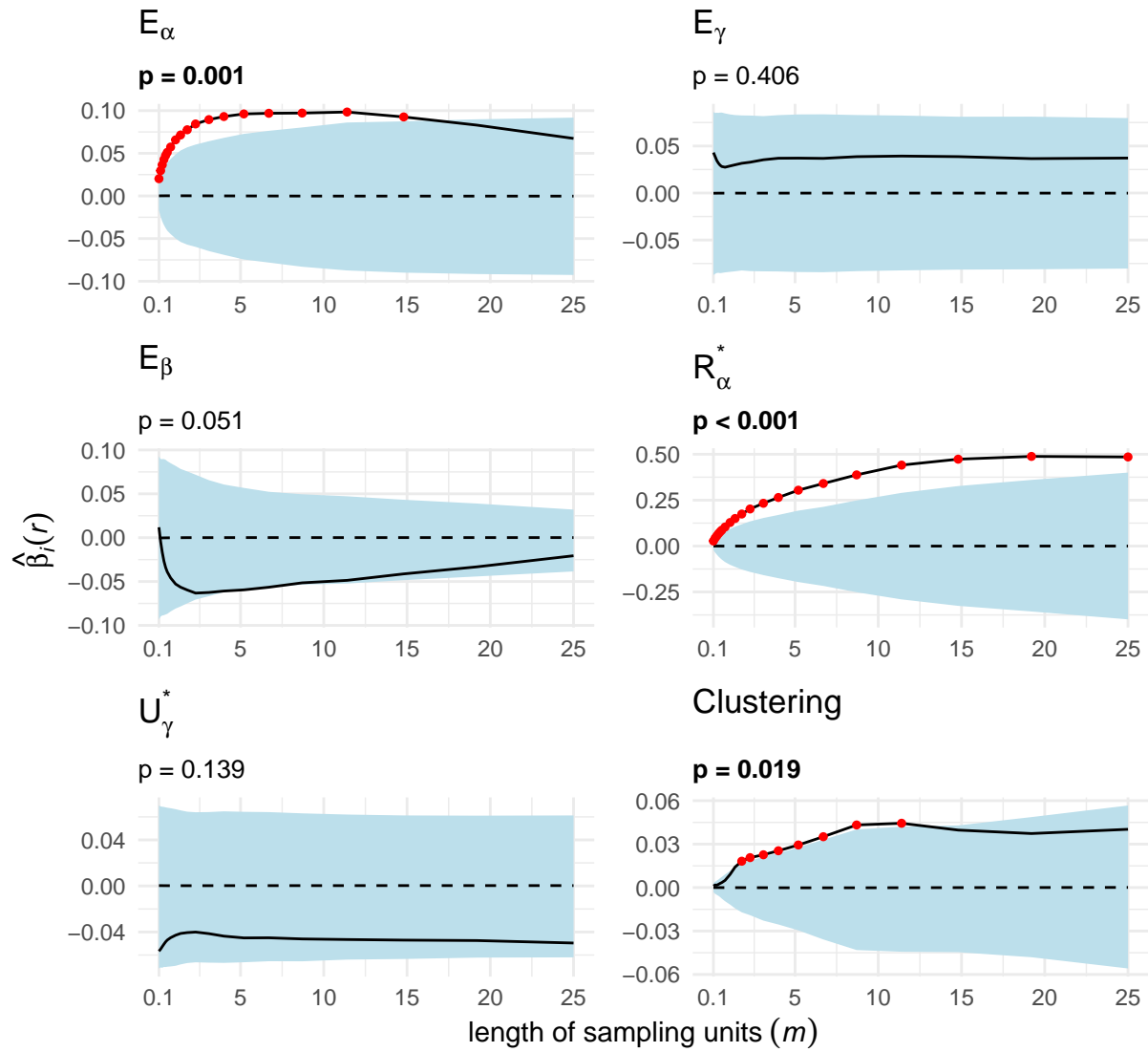


```
ggsave('plots/clonal.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)
ggsave('plots/FIG5.eps', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

## single predictor models ----

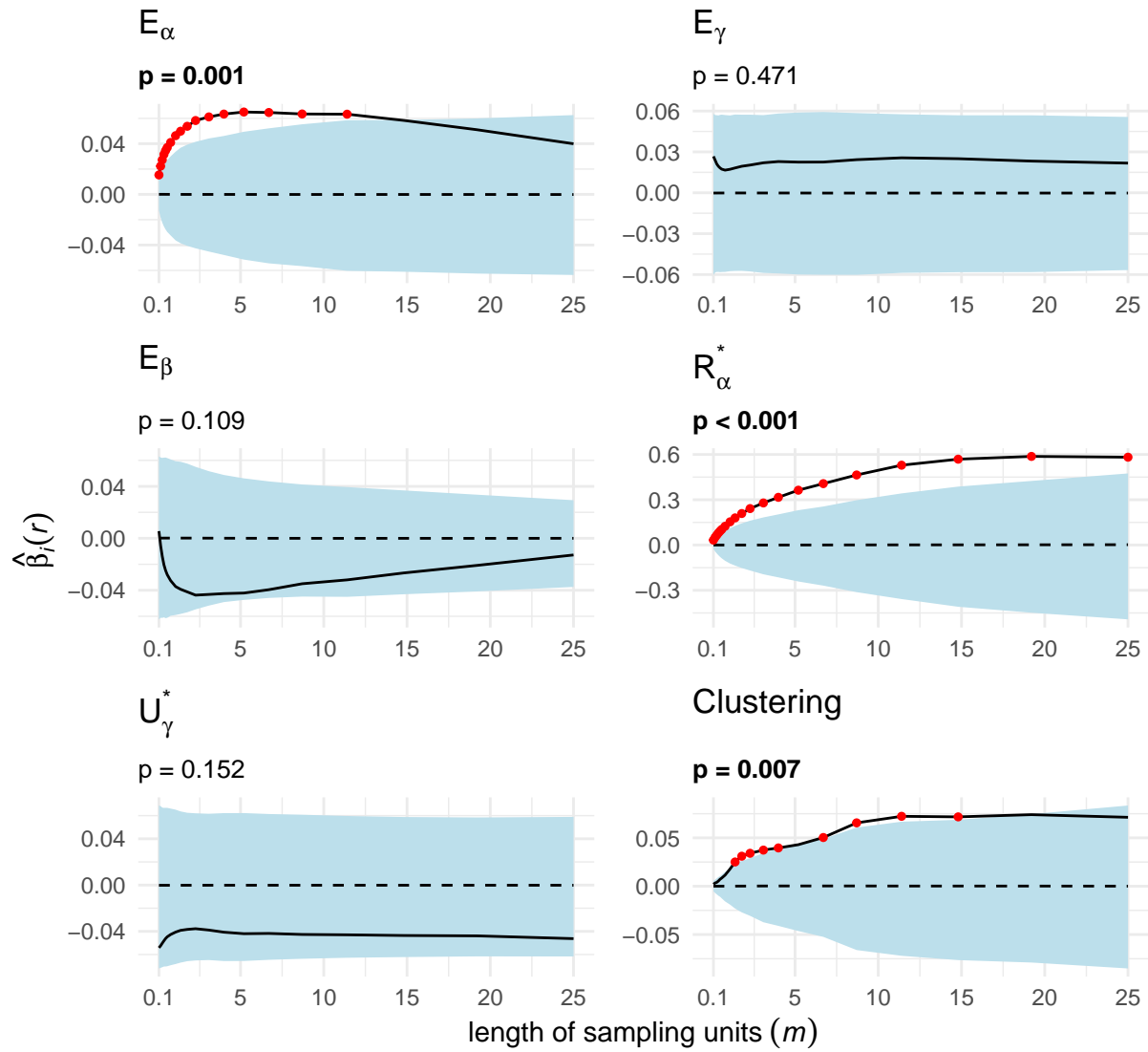
### RH050 ----
plots_RH <- lapply(flm_RH, function (x)
  imap(x, ~ plot_FLM(.x, title = labels[.y])))
)

patch(plots_RH$overall)
```



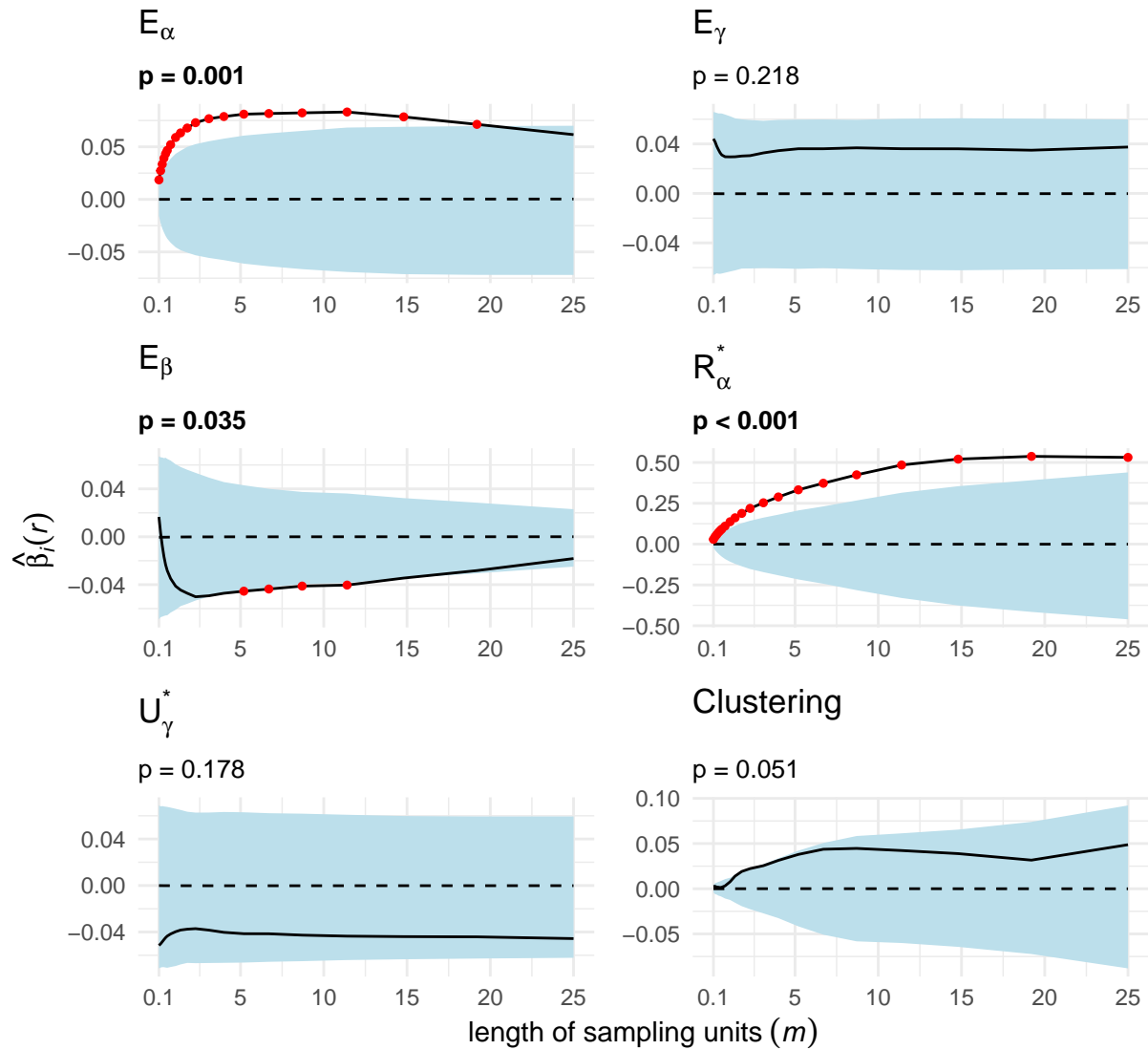
```
ggsave('plots/RHOverall.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots_RH$aboveground)
```



```
ggsave('plots/RHabove.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

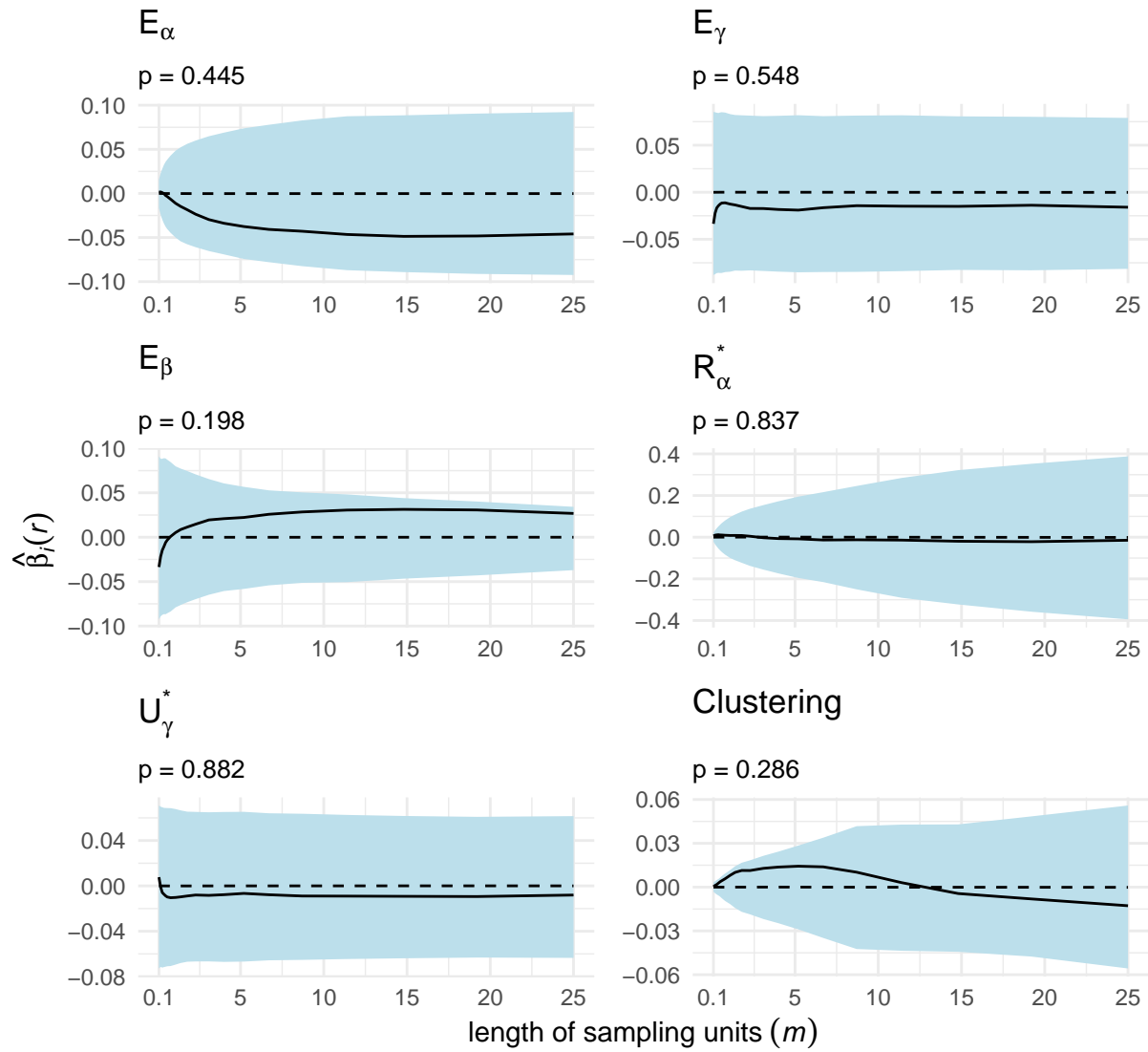
patch(plots_RH$clonal)
```

```
ggsave('plots/RHclonal.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

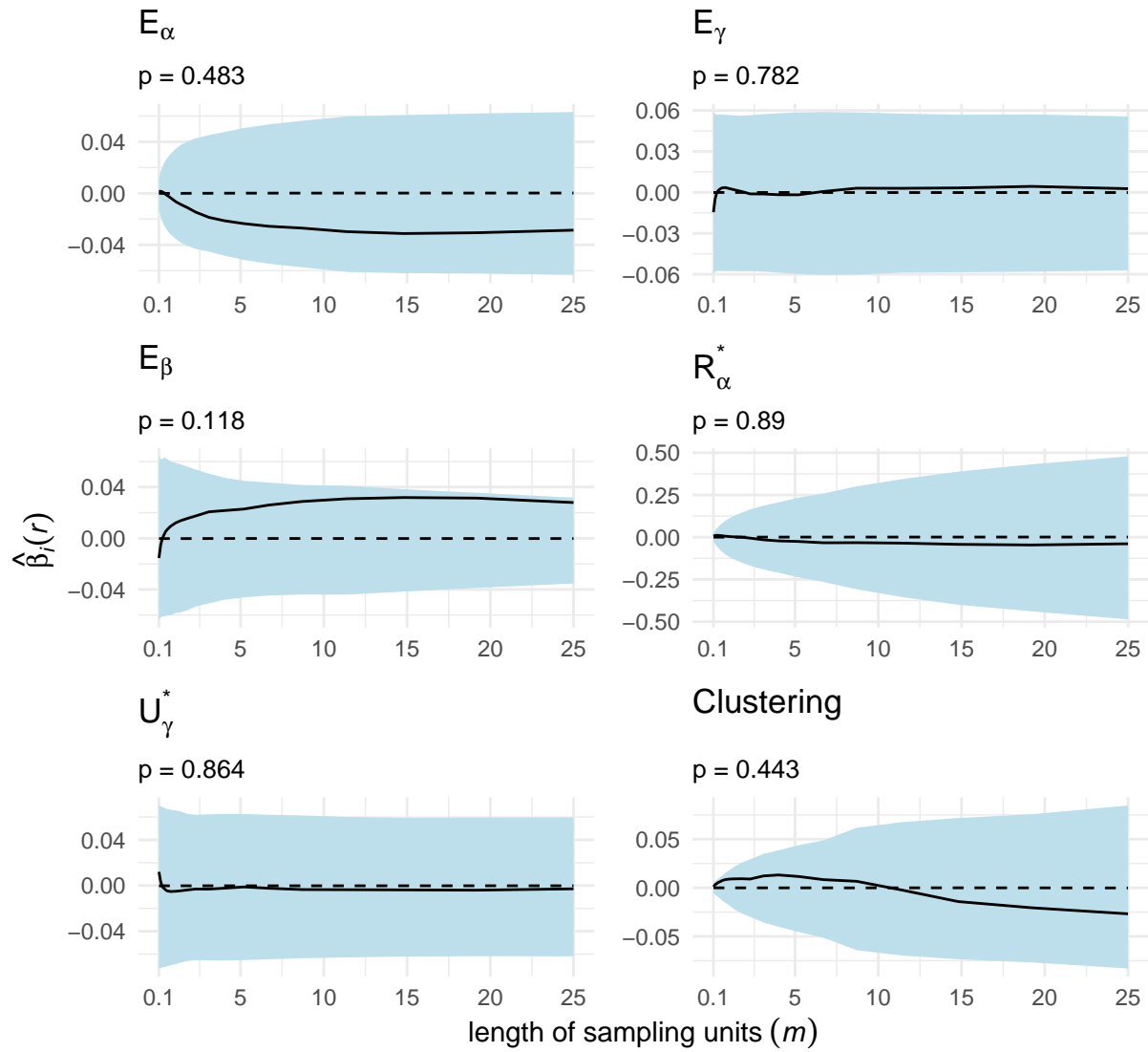
### LAI ----
plots_LAI <- lapply(flm_LAI, function (x)
  imap(x, ~ plot_FLM(.x, title = labels[.y])))
)

patch(plots_LAI$overall)
```



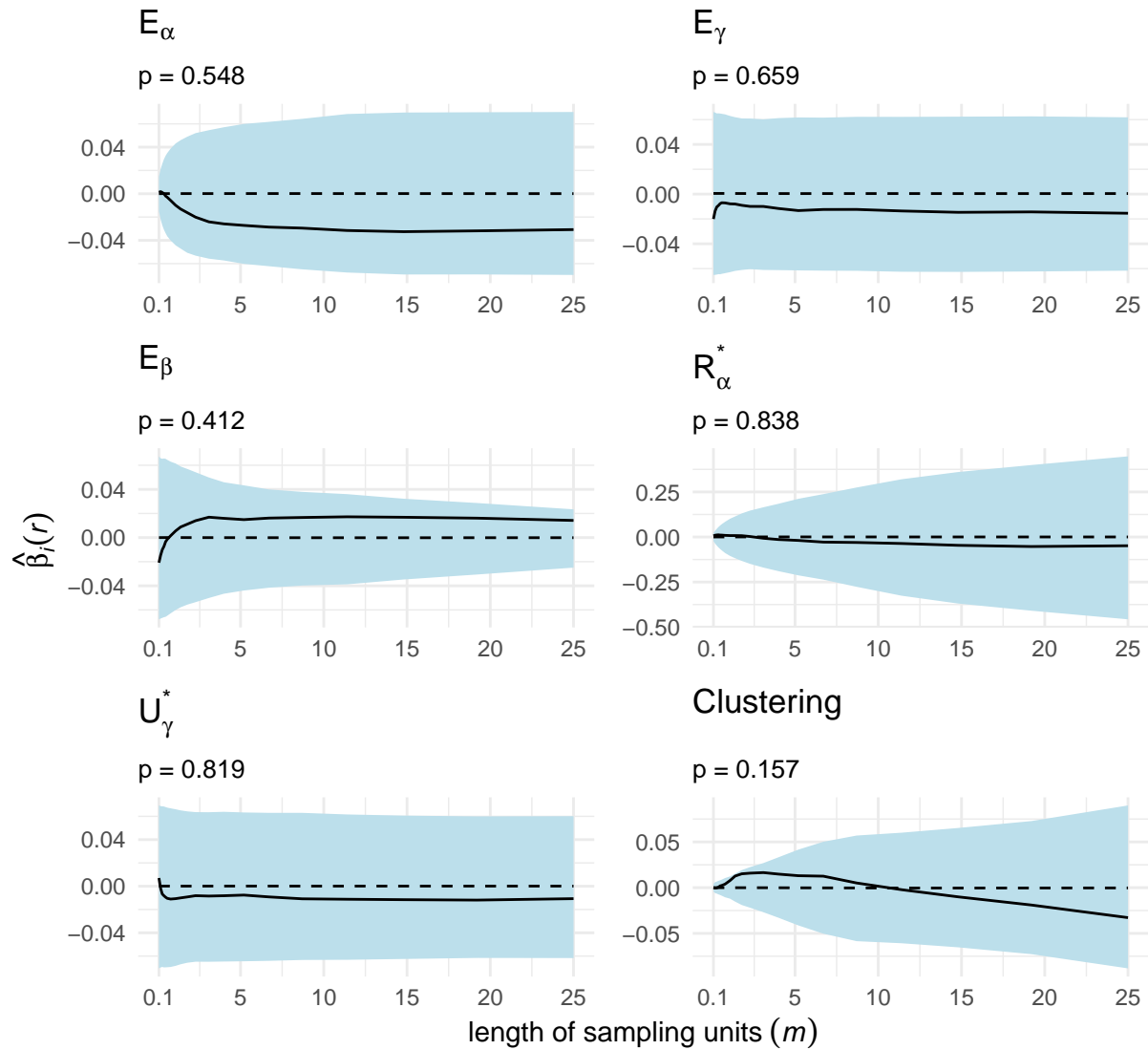
```
ggsave('plots/LAIoverall.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots_LAI$aboveground)
```



```
ggsave('plots/LAIabove.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

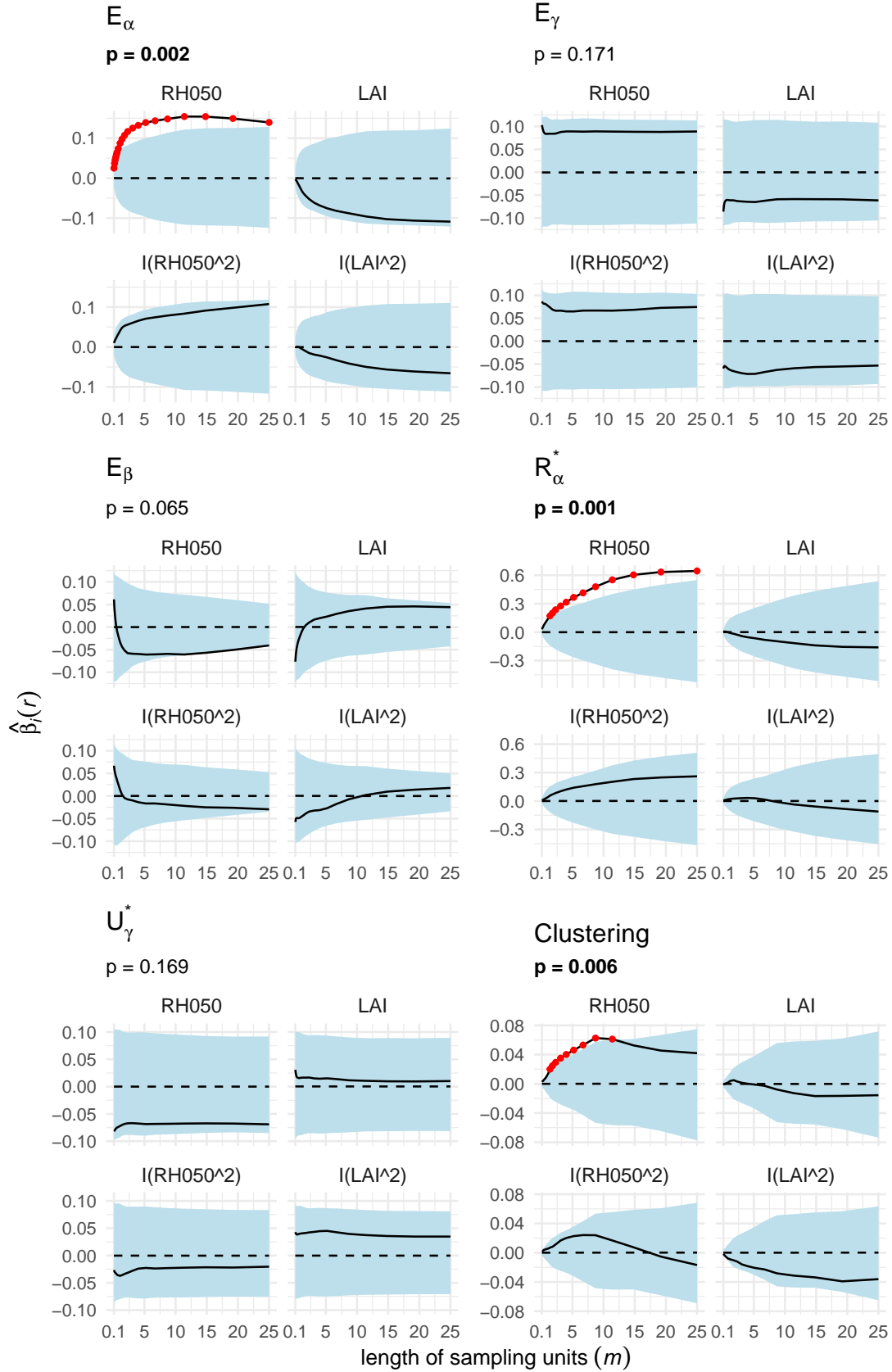
patch(plots_LAI$clonal)
```



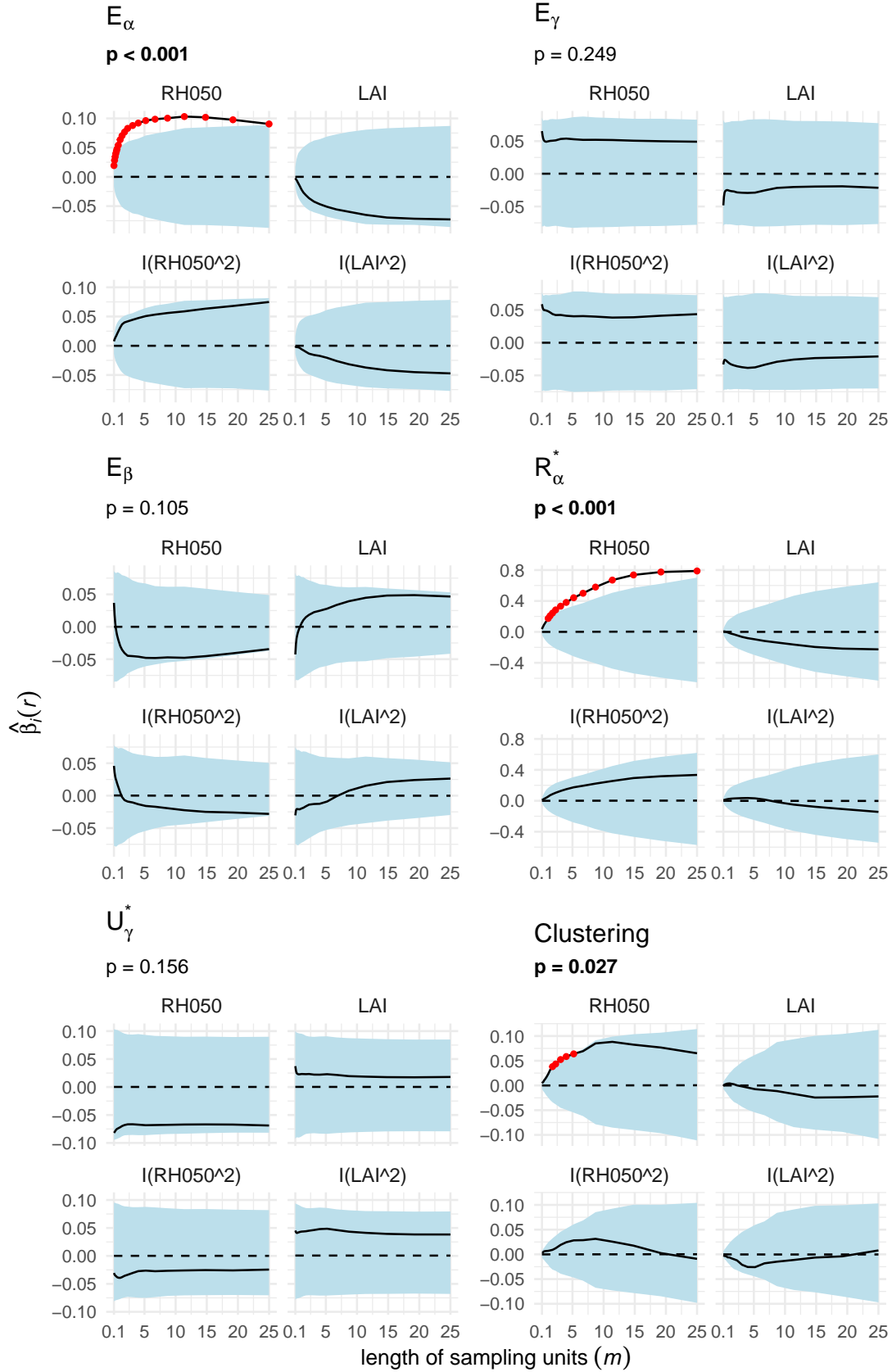
```
ggsave('plots/LAIclonal.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)
```

```
## quadratic model ----
plots_quad <- lapply(flm_quadratic, function (x)
  imap(x, ~ plot_FLM(.x, title = labels[.y])))
)
```

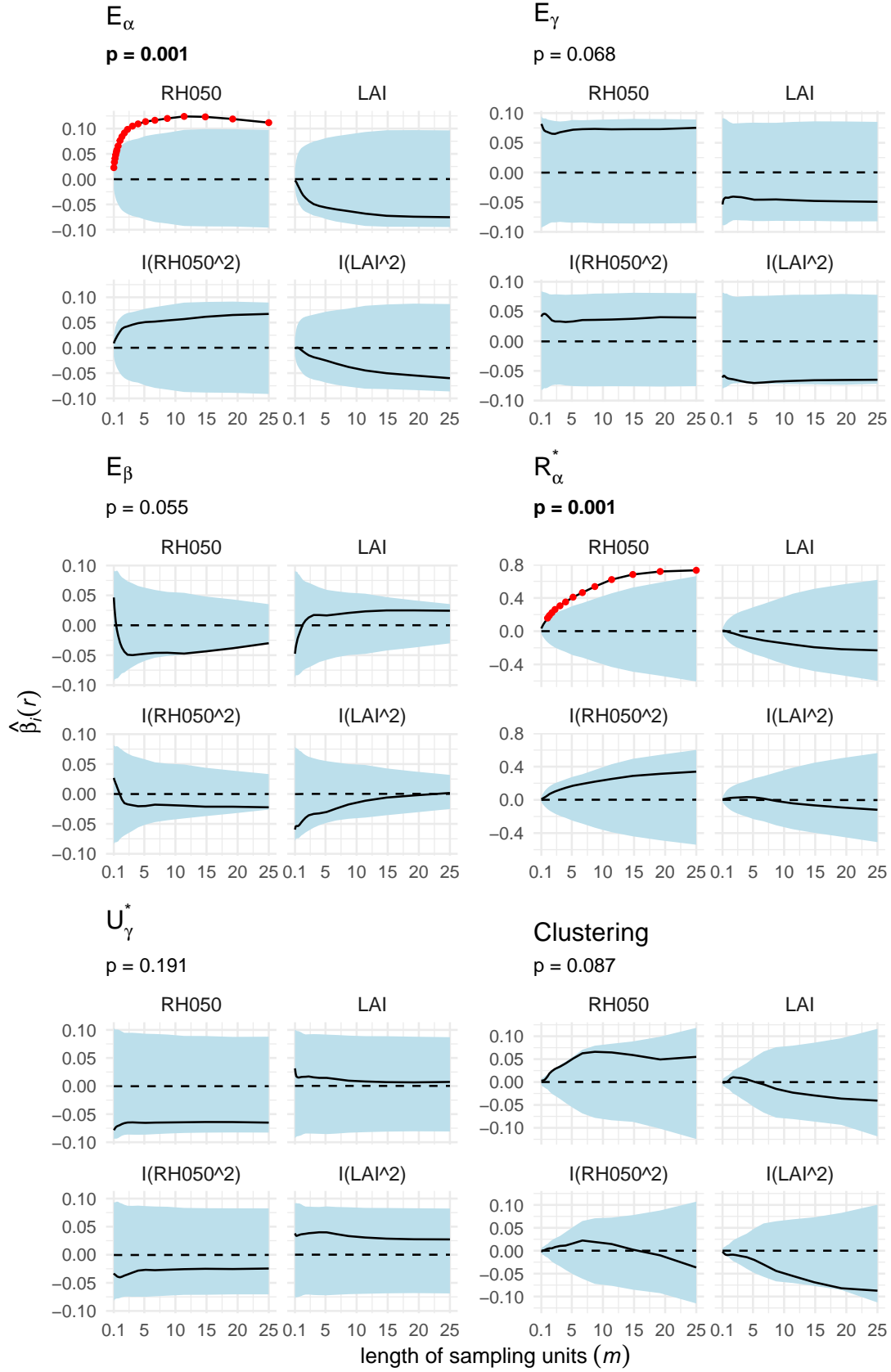
```
patch(plots_quad$overall)
```



```
ggsave('plots/quad_overall.png', width = 190, height = 117, units = "mm",  
       bg = 'white', scale = 1.5, dpi = 1000)  
  
patch(plots_quad$aboveground)
```



```
ggsave('plots/quad_above.png', width = 190, height = 117, units = "mm",  
       bg = 'white', scale = 1.5, dpi = 1000)  
  
patch(plots_quad$clonal)
```

```
ggsave('plots/quad_clonal.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)
```

```
# functional axes models -----
```

```
#load data
```

```
curves_axes <- readRDS("curves_axes.rds")
```

```
#select indices
```

```
curves_axes <- lapply(curves_axes, function (x) x[indices])
```

Models have been fit prior to making this report. To refit the models, uncomment the following lines and comment “load results”.

```
# #model
```

```
# flm_axes <- lapply(curves_axes, function (x)
```

```
#   lapply(x, function (y) flmtest(y, formula = formula(Y ~ RH050 + LAI), cl = cl))
```

```
# )
```

```
#
```

```
# #save results
```

```
# saveRDS(flm_axes, "FLM_axes.rds")
```

```
#load results
```

```
flm_axes <- readRDS("FLM_axes.rds")
```

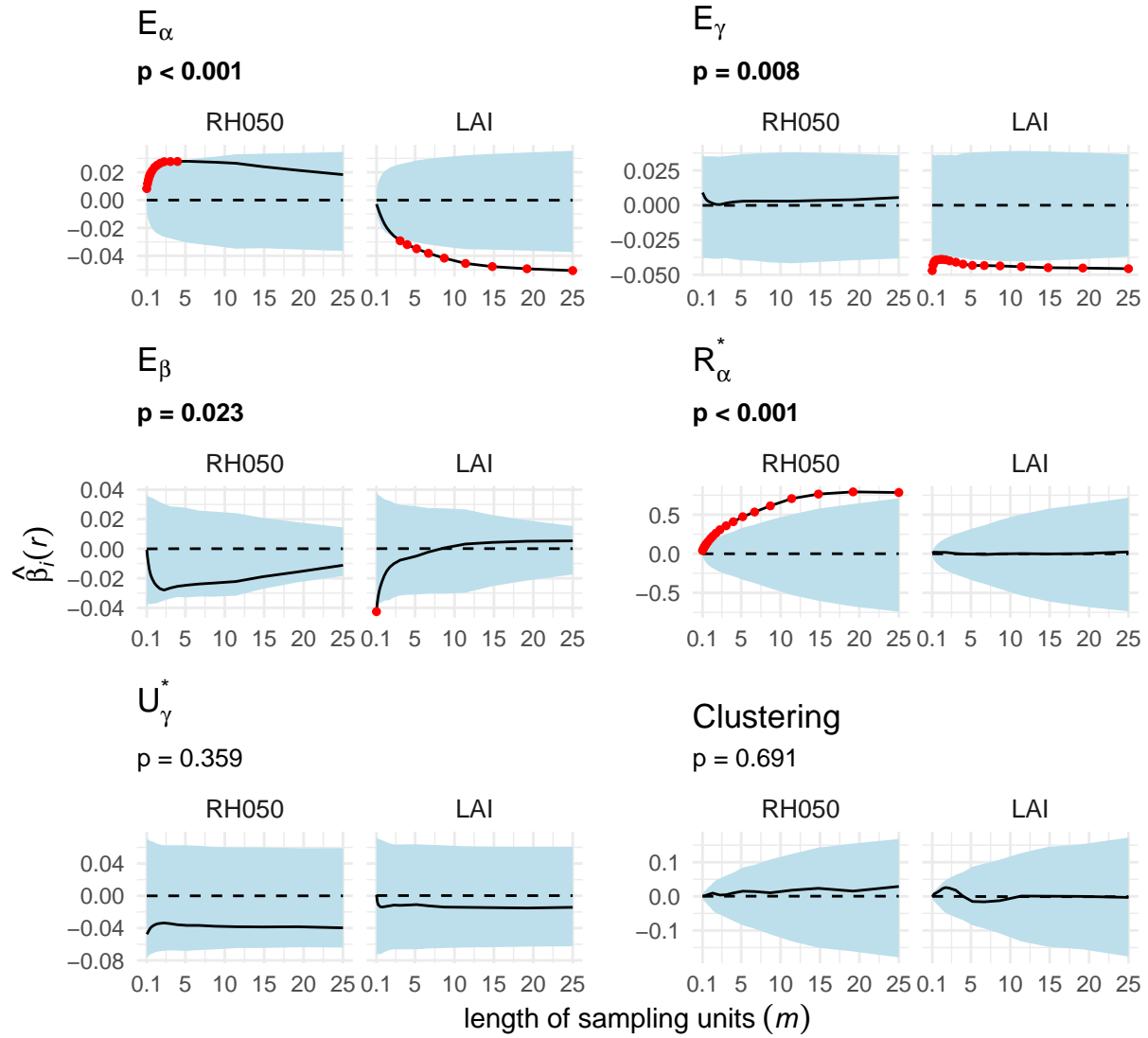
```
#plot
```

```
plots_axes <- lapply(flm_axes, function (x)
```

```
  imap(x, ~ plot_FLM(.x, title = labels[.y])))
```

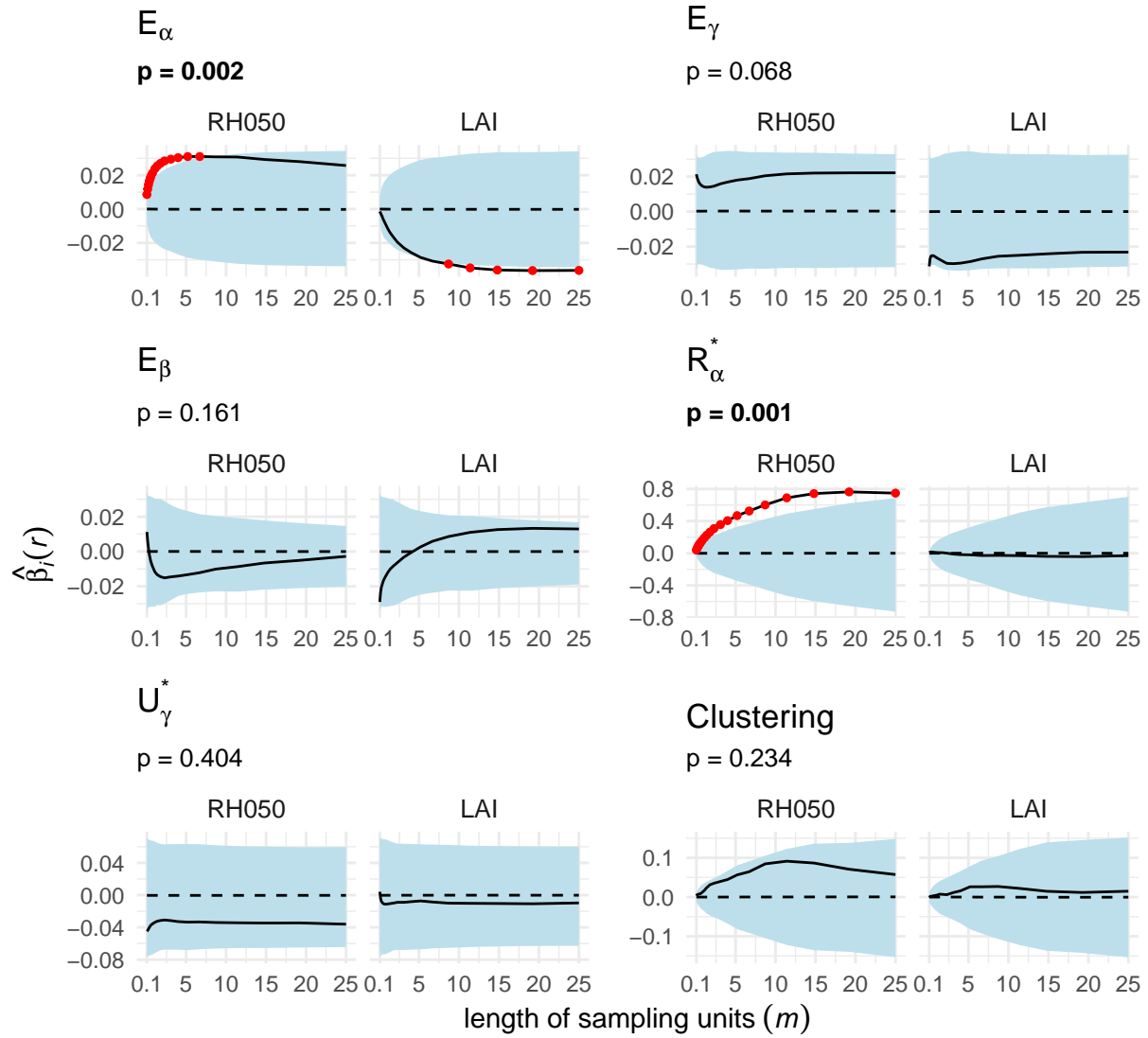
```
)
```

```
patch(plots_axes$above1)
```



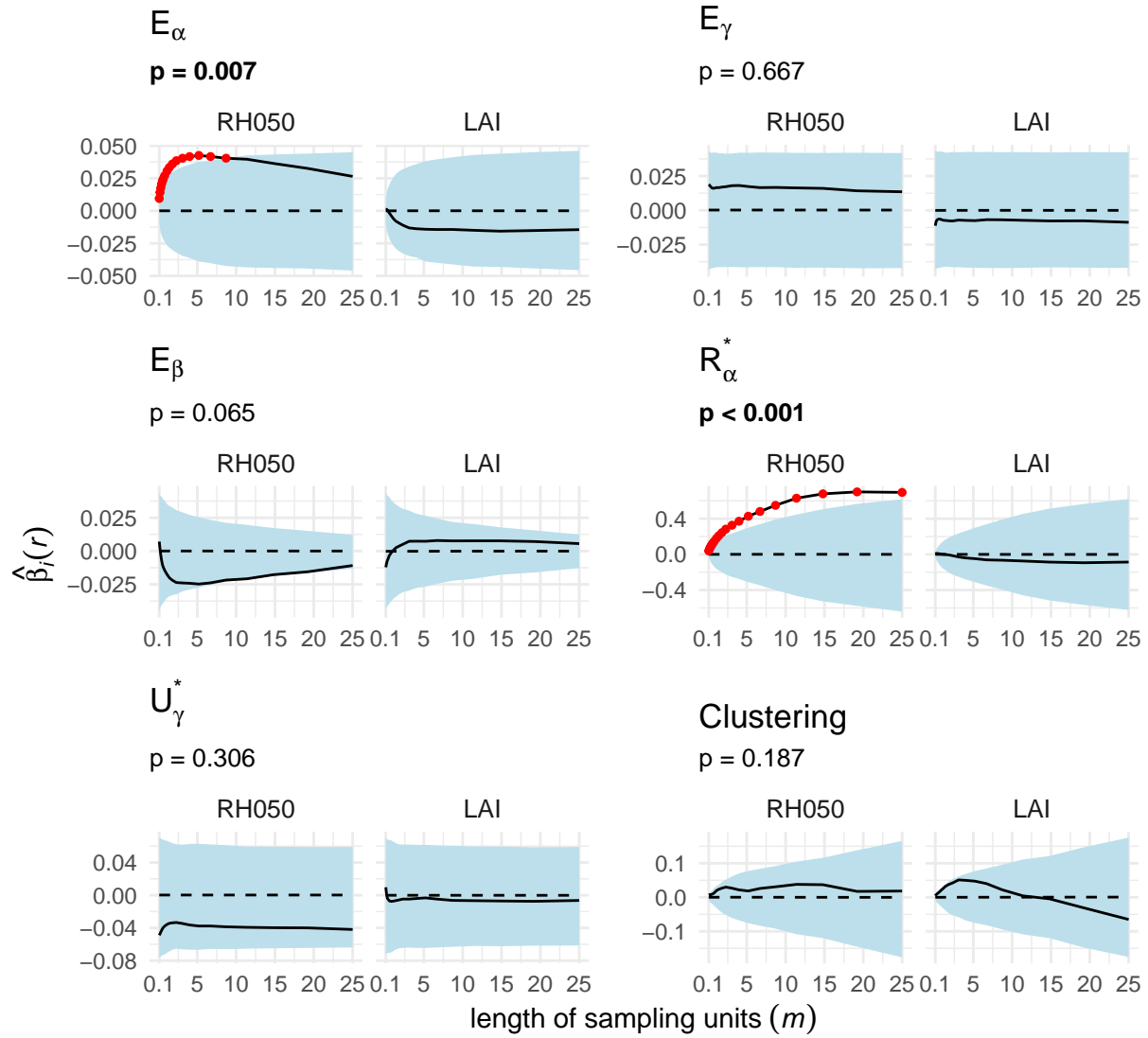
```
ggsave('plots/aboveRC1.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots_axes$above2)
```



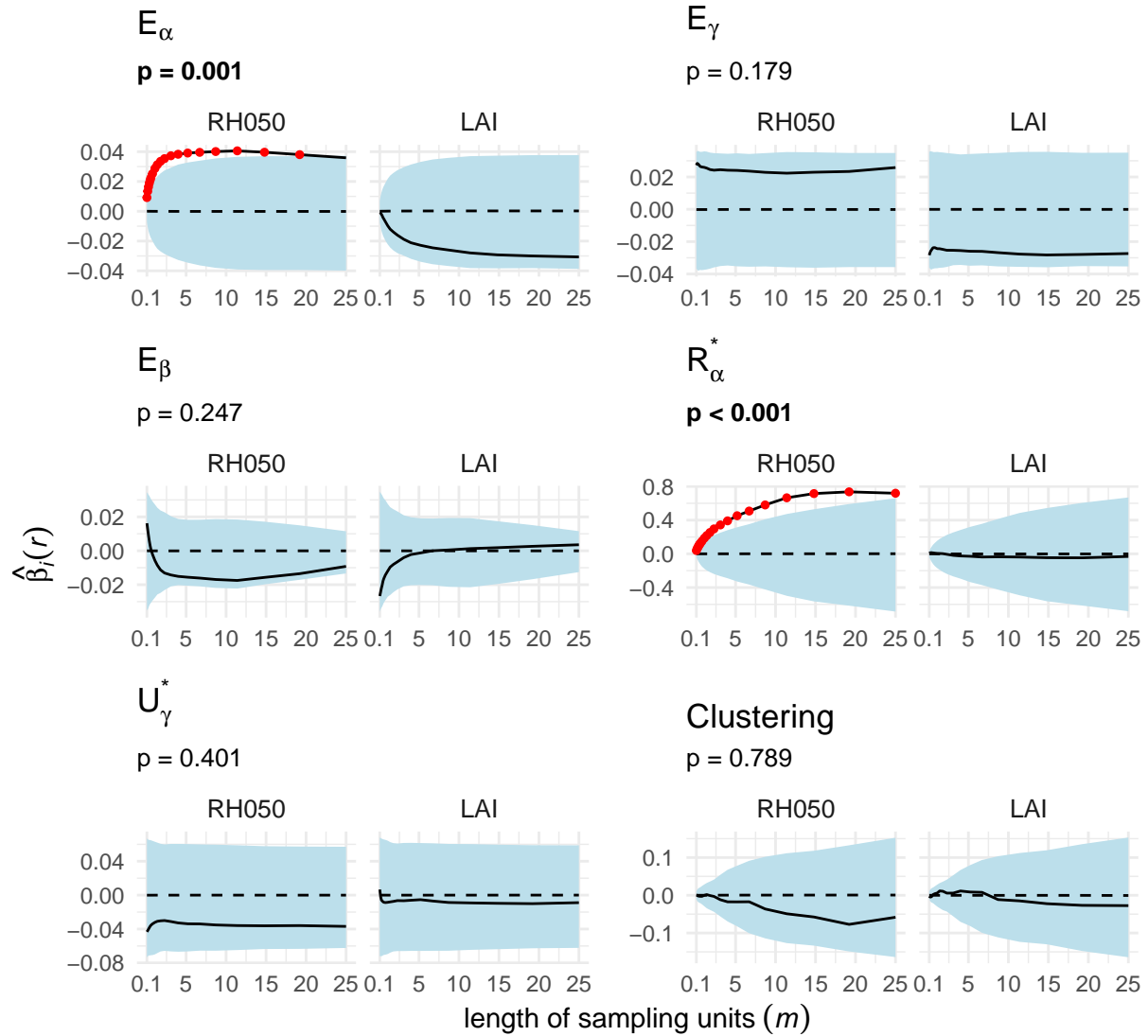
```
ggsave('plots/aboveRC2.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots_axes$clonal1)
```



```
ggsave('plots/clonalRC1.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots_axes$clonal2)
```



```
ggsave('plots/clonalRC2.png', width = 190, height = 117, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

# supplementary indices -----

#load data
supp <- readRDS("curves.rds")

#select indices

indices_supp <- c("alpha", "gamma", "beta_add", "E_alpha_sp", "E_gamma_sp", "E_beta_sp")

#nicer labels for plotting
labels_supp <- list(alpha = expression(Q[alpha]),
                    gamma = expression(Q[gamma]),
                    beta_add = expression(Q[beta]),
```

```

E_alpha_sp = expression(E[alpha~tax]),
E_gamma_sp = expression(E[gamma~tax]),
E_beta_sp = expression(E[beta~tax]))

supp <- lapply(supp, function (x) x[indices_supp])

#models

```

Models have been fit prior to making this report. To refit the models, uncomment the following lines and comment “load results”.

```

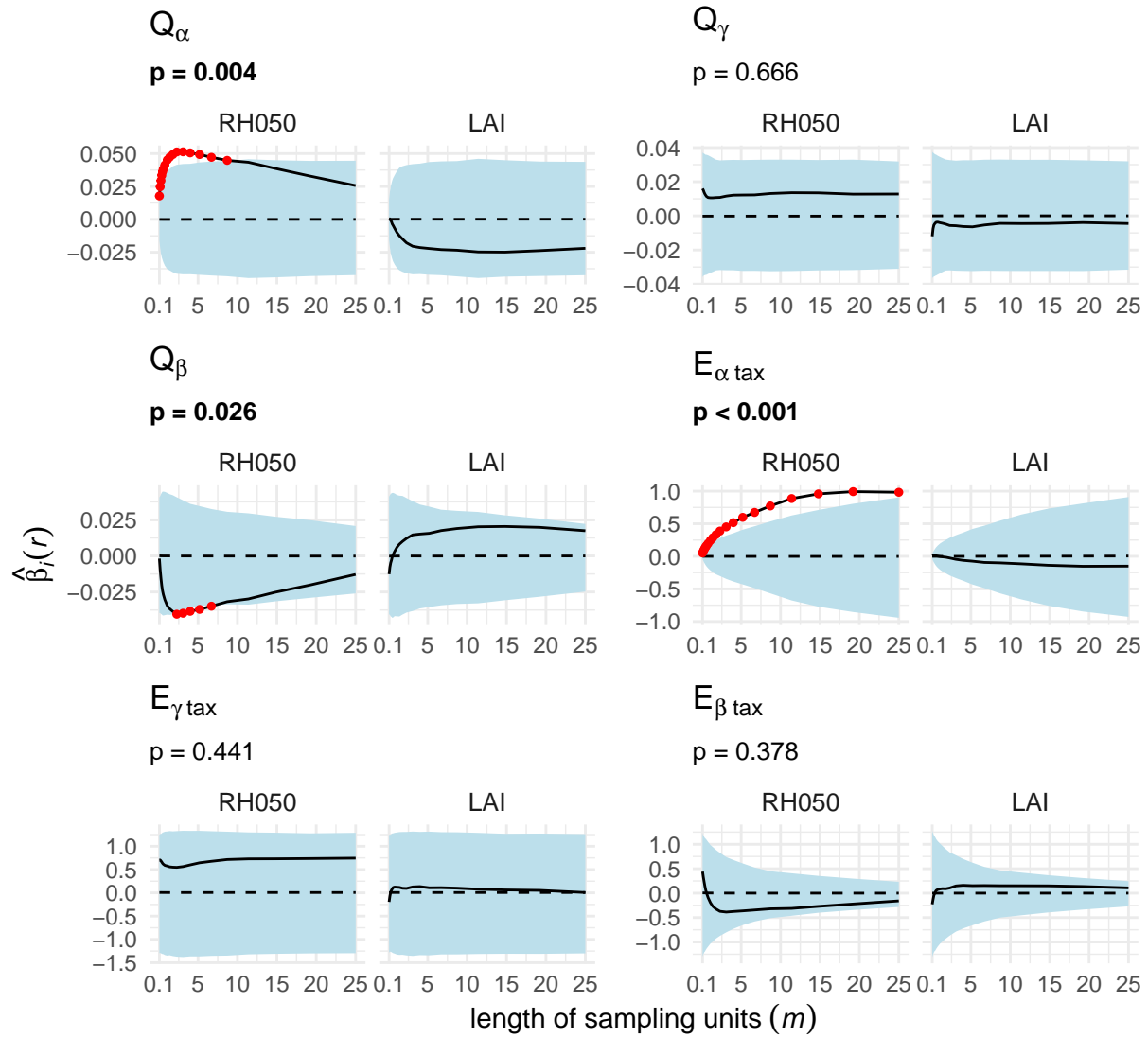
# flm_supp <- lapply(supp, function (x)
#   lapply(x, function (y) flmtest(y, formula = formula(Y ~ RH050 + LAI), cl = cl))
# )
#
# #save results
# saveRDS(flm_supp, "FLM_supp.rds")

#load results
flm_supp <- readRDS("FLM_supp.rds")

#plot
plots_supp <- lapply(flm_supp, function (x)
  imap(x, ~ plot_FLM(.x, title = labels_supp[[".y"]]))
)

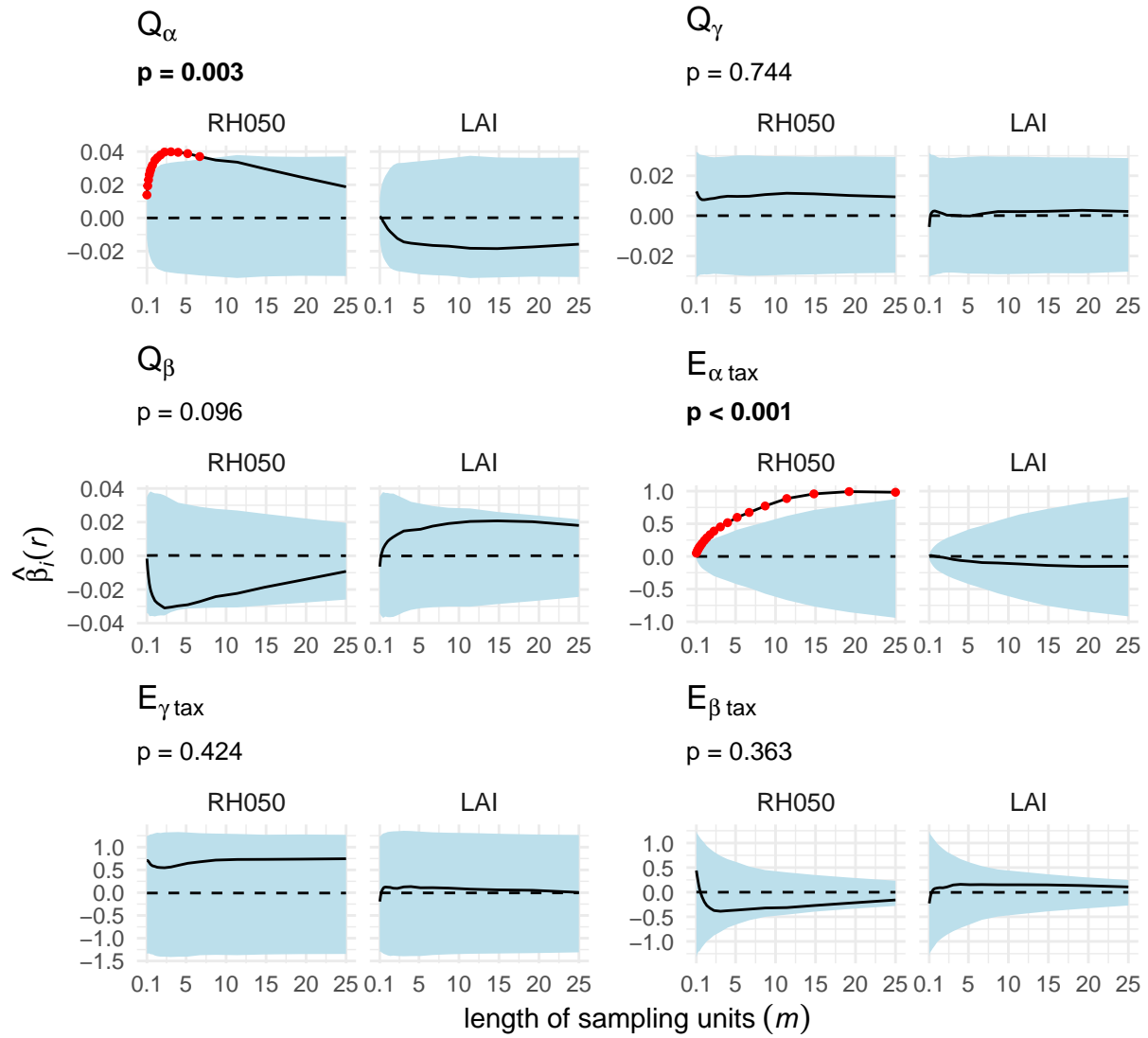
patch(plots_supp$overall)

```



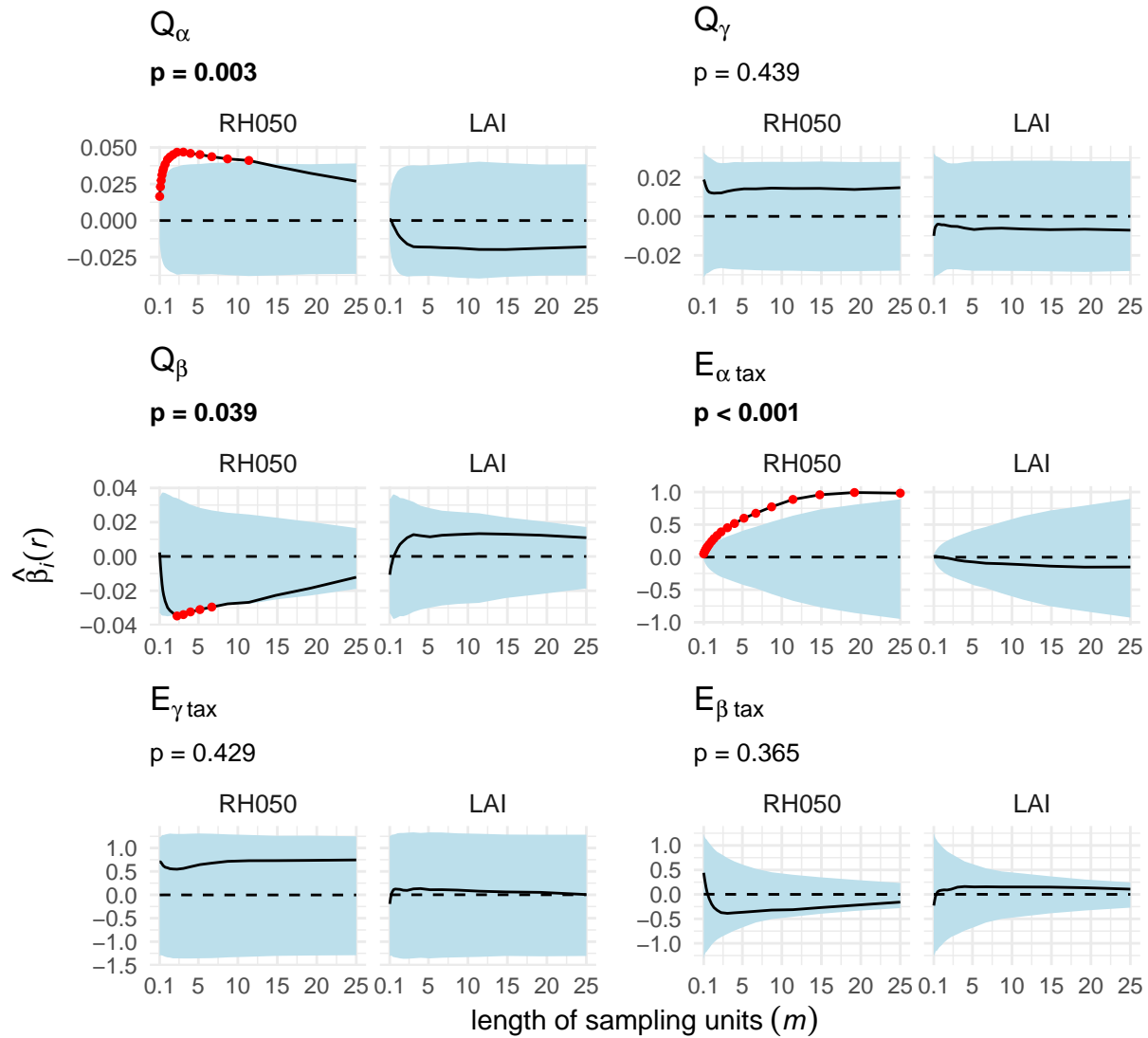
```
ggsave('plots/overall_supp.png', width = 190, height = 90, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots_supp$above)
```

```
ggsave('plots/above_supp.png', width = 190, height = 90, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)

patch(plots_supp$clonal)
```



```
ggsave('plots/clonal_supp.png', width = 190, height = 90, units = "mm",
       bg = 'white', scale = 1.5, dpi = 1000)
```

```
# session info -----
sessionInfo()
```

```
## R version 4.5.2 (2025-10-31)
## Platform: x86_64-redhat-linux-gnu
## Running under: Nobara Linux 43 (KDE Plasma Desktop Edition)
##
## Matrix products: default
## BLAS/LAPACK: FlexiBLAS OPENBLAS-OPENMP; LAPACK version 3.12.1
##
## Random number generation:
## RNG:      L'Ecuyer-CMRG
## Normal:   Inversion
```

```

## Sample: Rejection
##
## locale:
## [1] LC_CTYPE=it_IT.UTF-8      LC_NUMERIC=C
## [3] LC_TIME=it_IT.utf8       LC_COLLATE=it_IT.UTF-8
## [5] LC_MONETARY=it_IT.utf8   LC_MESSAGES=it_IT.UTF-8
## [7] LC_PAPER=it_IT.utf8      LC_NAME=C
## [9] LC_ADDRESS=C             LC_TELEPHONE=C
## [11] LC_MEASUREMENT=it_IT.utf8 LC_IDENTIFICATION=C
##
## time zone: Europe/Rome
## tzcode source: system (glibc)
##
## attached base packages:
## [1] parallel stats      graphics grDevices datasets utils      methods
## [8] base
##
## other attached packages:
## [1] patchwork_1.3.2 GET_1.0-7      lubridate_1.9.4 forcats_1.0.1
## [5] stringr_1.6.0  dplyr_1.1.4      purrr_1.2.0     readr_2.1.6
## [9] tidyr_1.3.2    tibble_3.3.0     ggplot2_4.0.1   tidyverse_2.0.0
##
## loaded via a namespace (and not attached):
## [1] generics_0.1.4  stringi_1.8.7    hms_1.1.4        digest_0.6.39
## [5] magrittr_2.0.4  evaluate_1.0.5   grid_4.5.2       timechange_0.3.0
## [9] RColorBrewer_1.1-3 fastmap_1.2.0    gridExtra_2.3     viridisLite_0.4.2
## [13] scales_1.4.0    CoprManager_0.5.7 textshaping_1.0.4 cli_3.6.5
## [17] rlang_1.1.6     crayon_1.5.3     withr_3.0.2       yaml_2.3.12
## [21] tools_4.5.2     tzdb_0.5.0       vctrs_0.6.5      R6_2.6.1
## [25] lifecycle_1.0.4 ragg_1.5.0       cluster_2.1.8.1   pkgconfig_2.0.3
## [29] pillar_1.11.1   gtable_0.3.6     glue_1.8.0        systemfonts_1.3.1
## [33] xfun_0.55       tidyselect_1.2.1 rstudioapi_0.17.1 knitr_1.50
## [37] farver_2.1.2    htmltools_0.5.9  rmarkdown_2.30    labeling_0.4.3
## [41] compiler_4.5.2  S7_0.2.1

```