

Create line plots - clean data

Abigail Seeger

January 8, 2021

Contents

Read in the data	2
Function to create line plots	2
Create Plots	4
December	4
January	31
Februray	58

This code will recreate the line plots created in the Summer. Except, I will use the data where the outliers are removed, and the data is quantile normalized.

These plots will be created for each month (December, January, and February), and each measurement (npq, phi2, and leafarea).

```
library(dplyr)
library(tidyverse)
library(ggplot2)
### Lemon is used in ggplot2 -
### facet_rep_grid modification
library(lemon)
library(data.table)
library(ggthemes)
library(extrafont)
### Routliers is used for outliersmad to
### find outliers
library(Routliers)
library(stringi)
library(wesanderson)
library(viridis)
library(reshape2)
library(sm)
library(lme4)
library(lmerTest)
library(lsmeans)
library(car)
library(ggcorrplot)
```

```
library(preprocessCore)
library(grid)
```

Read in the data

```
depi_data <- read.table("C:/Users/Owner/Documents/Research/Shiu_Lab/Shiu_Lab_R/Data/Clean_DEPI_Data.csv",
  sep = ",", header = TRUE)
```

Function to create line plots

```
add_number <- function(data_frame) {
  ### First, if the genotype is Col0 (only
### genotype with length 4), assign 0 as
### number Else, assign number as genotype
### with 'mpk' removed Example: mpk1 will
### be 1, mpk1-17 will be 1-17
  data_frame <- data_frame %>% mutate(number = ifelse(genotype !=
    "Col0", (stri_sub(genotype, 4, length(genotype))),
    0))
  ### Next, for all double mutants, replace
### '-' with '0' Example: 1-17 becomes 1017
  data_frame$number <- as.numeric(gsub("-",
    "0", data_frame$number))
  ### Almost there! There's a problem with
### two single digit double mutants We need
### a four digit number to sort correctly
### Example: mpk1-3 -> 1-3 -> 103, but we
### need it to be 1003 to sort correctly
  data_frame$number[data_frame$number ==
    "103"] <- "1003"
  data_frame$number[data_frame$number ==
    "506"] <- "5006"
  data_frame$number[data_frame$number ==
    "608"] <- "6008"
  data_frame$number[data_frame$number ==
    "609"] <- "6009"
  ### Convert number to a numeric in order
### to sort
  data_frame$number <- as.numeric(data_frame$number)
  data_frame <- data_frame %>% arrange(number)
  data_frame <- data_frame %>% mutate(number_2 = number)
  data_frame$number_2[nchar(data_frame$number_2) ==
    4] <- 0
  data_frame$number_2[nchar(data_frame$number_2) ==
    5] <- 0
  return(data_frame)
}
```

Note that in this function, I changed measured_value to normalized_value.

```
cell_370_data <- function(data_frame) {
  npq_phi2 <- data_frame %>% filter(measurement %in%
    c("npq", "phi2")) %>% group_by(genotype,
    time_point, measurement, month) %>%
    summarize(med = median(normalized_value))
  ### For each day, we want the minimum and
  ### maximum time point for leaf area

  ### Previously included the midpoint -
  ### leave code in case we want to use in
  ### the future, just commented out

  ### If there are an odd number of time
  ### points, use the median time point If
  ### there are an even number of time
  ### points, instead of finding the average
  ### of the two center values, choose the
  ### larger value start_mid_end <-
  ### unique((data_frame%>%group_by(day)%>%filter(time_point
  ### %in% c(min(time_point),
  ### max(time_point),
  ### ifelse(length(time_point %% 2 == 0),
  ### median(time_point[-1]),
  ### median(time_point))))$time_point)

  start_end <- unique((data_frame %>% group_by(day) %>%
    filter(time_point %in% c(min(time_point),
    max(time_point))))$time_point)

  leaf_area <- data_frame %>% filter(measurement ==
    "leafarea") %>% filter(time_point %in%
    start_end) %>% group_by(genotype,
    time_point, measurement) %>% summarize(med = median(normalized_value))

  out <- rbind(npq_phi2, leaf_area) %>%
    group_by(genotype, time_point, measurement,
    month)

  return(as.data.frame(out))
}
```

```
add_day_col <- function(data_frame) {
  unique_time <- sort(unique(data_frame$time_point))
  diff <- c()
  for (i in 1:length(unique_time)) {
    if (i == 1) {
      diff[1] <- 0
    } else {
      diff[i] <- unique_time[i] - unique_time[i -
        1]
    }
  }
}
```

```

breaks <- c(0)
for (i in 1:length(diff)) {
  if (diff[i] > 5)
    breaks <- append(breaks, unique_time[i])
}
out <- data.frame()
for (i in 1:length(breaks)) {
  if (i == length(breaks)) {
    indiv <- data_frame %>% filter(time_point >=
      breaks[i]) %>% mutate(day = i)
  } else {
    indiv <- data_frame %>% filter(time_point >=
      breaks[i] & time_point <
      breaks[i + 1]) %>% mutate(day = i)
  }

  out <- rbind(as.data.frame(indiv),
    out)
}
return(out)
}

```

```

genotype_combinations <- list(c("mpk1-17",
  "mpk1", "mpk17"), c("mpk1-16", "mpk1",
  "mpk16"), c("mpk6-9", "mpk6", "mpk9"),
c("mpk17-20", "mpk17", "mpk20"), c("mpk14-17",
  "mpk14", "mpk17"), c("mpk8-17", "mpk8",
  "mpk17"), c("mpk8-20", "mpk8", "mpk20"),
c("mpk6-18", "mpk6", "mpk18"), c("mpk1-13",
  "mpk1", "mpk13"), c("mpk17-20", "mpk17",
  "mpk20"), c("mpk13-20", "mpk13",
  "mpk20"), c("mpk6-8", "mpk6", "mpk8"),
c("mpk9-18", "mpk9", "mpk18"), c("mpk6-20",
  "mpk6", "mpk20"), c("mpk14-16", "mpk14",
  "mpk16"), c("mpk18-20", "mpk18",
  "mpk20"), c("mpk5-6", "mpk5", "mpk6"),
c("mpk14-18", "mpk14", "mpk18"), c("mpk5-6",
  "mpk5", "mpk6"), c("mpk14-18", "mpk14",
  "mpk18"), c("mpk5-17", "mpk5", "mpk17"),
c("mpk1-3", "mpk1", "mpk3"), c("mpk1-17",
  "mpk1", "mpk17"), c("mpk3-16", "mpk3",
  "mpk16"), c("mpk9-16", "mpk9", "mpk16"),
c("mpk14-20", "mpk14", "mpk20"))

```

Create Plots

December

```

plot_data <- filter(depi_data, depi_data$month ==
  "Dec")

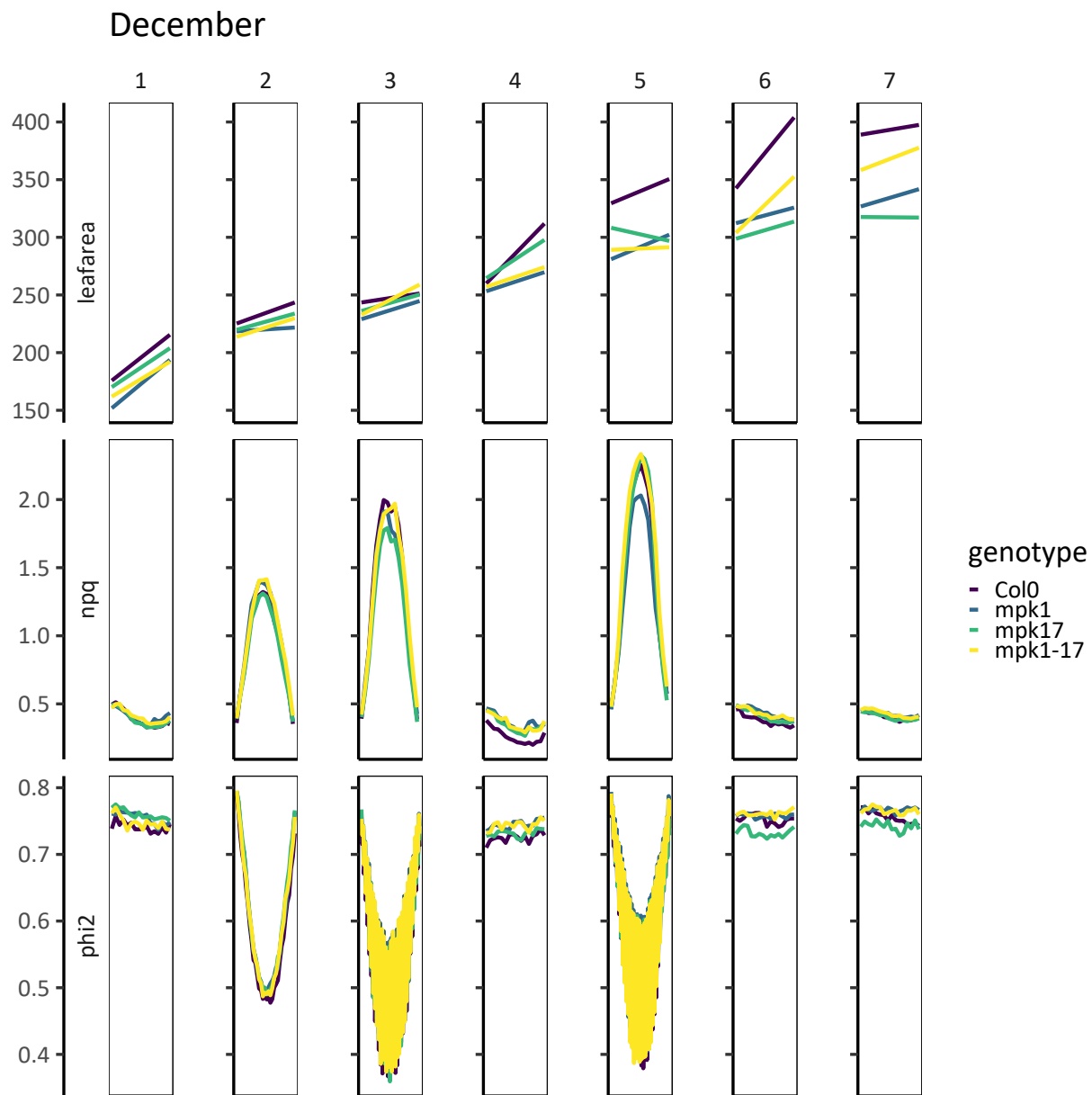
```

```

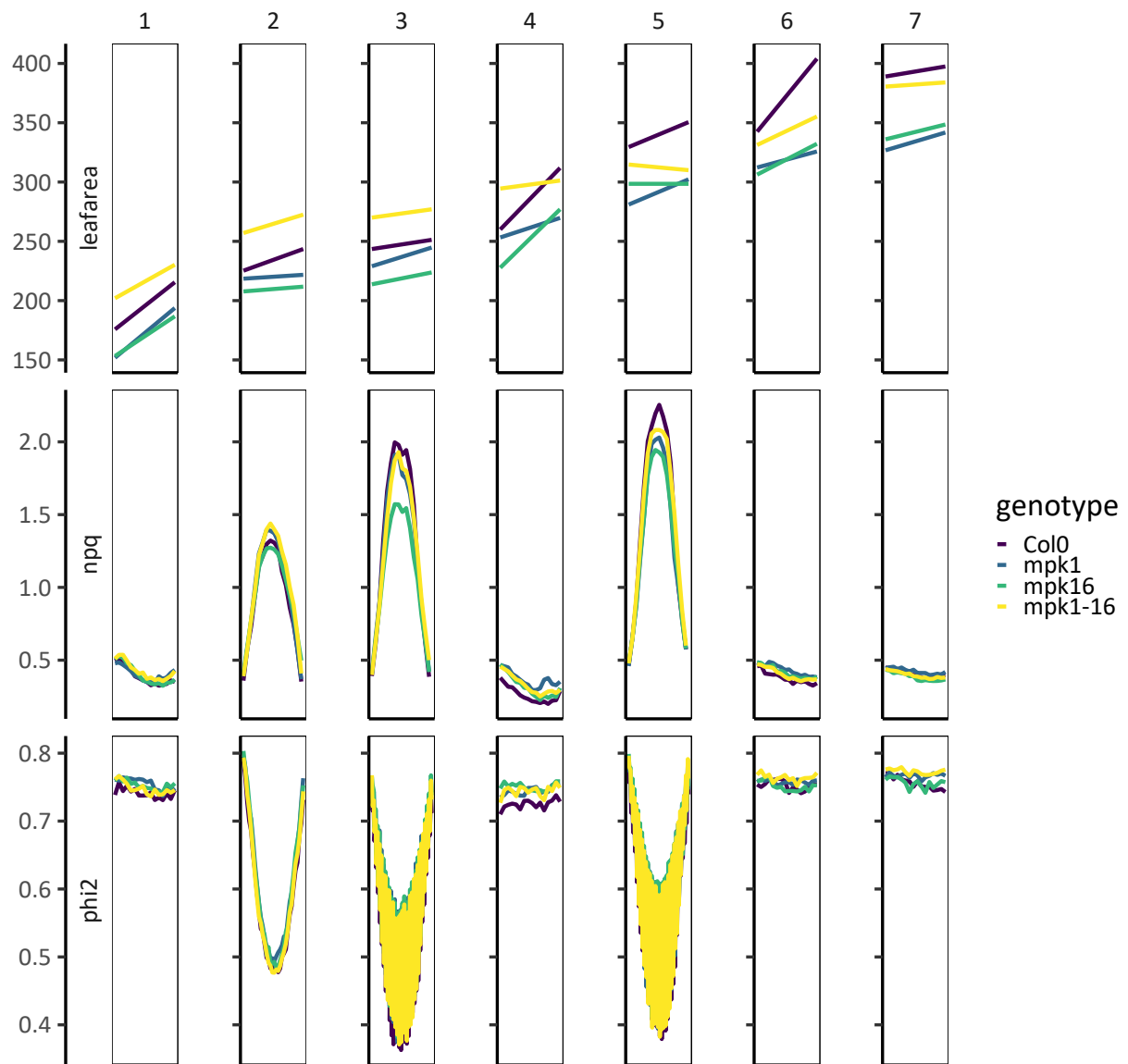
plot_data <- cell_370_data(plot_data)
plot_data <- add_number(plot_data)
plot_data <- add_day_col(plot_data)
plot_data$genotype <- reorder(plot_data$genotype,
  plot_data$number)

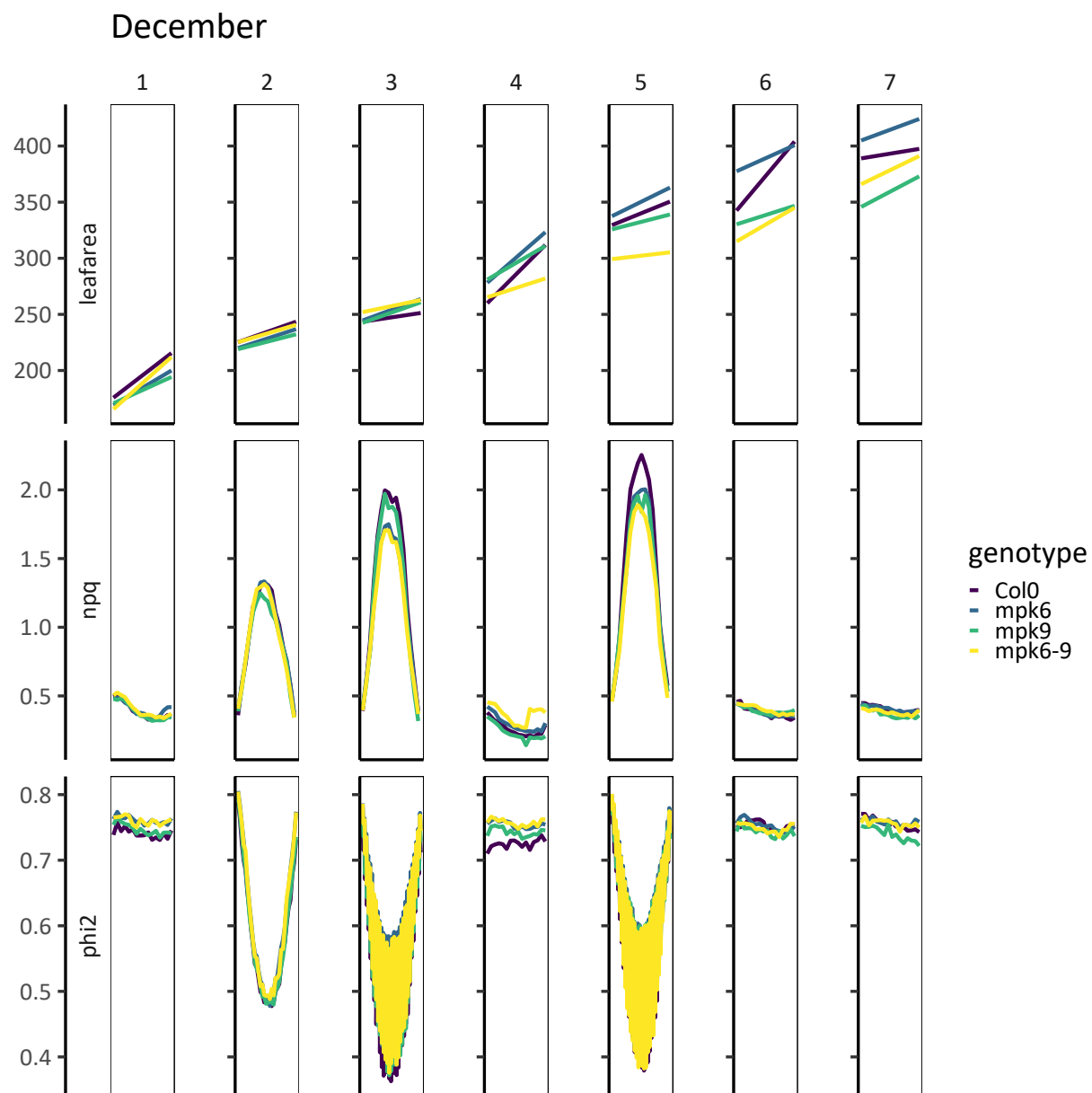
for (element in genotype_combinations) {
  data <- filter(plot_data, genotype %in%
    c(element, "Col0"))
  plot <- ggplot(data = data, aes(x = time_point,
    y = med)) + geom_line(aes(color = genotype),
    size = 3) + facet_rep_grid(measurement ~
    day, scales = "free", switch = "y",
    repeat.tick.labels = FALSE) + labs(x = "Hours",
    y = NULL, title = "December") + theme_tufte(base_family = "Calibri",
    base_size = 50) + theme(strip.background.x = element_blank(),
    axis.title.x = element_blank(), axis.text.x = element_blank(),
    axis.ticks.x = element_blank(), panel.border = element_rect(color = "black",
    fill = NA, size = 1), axis.line = element_line(),
    panel.spacing = unit(1, "lines")) +
    scale_color_viridis_d(begin = 0,
    end = 1, option = "viridis",
    aesthetics = c("colour", "fill"))
  print(plot)
}

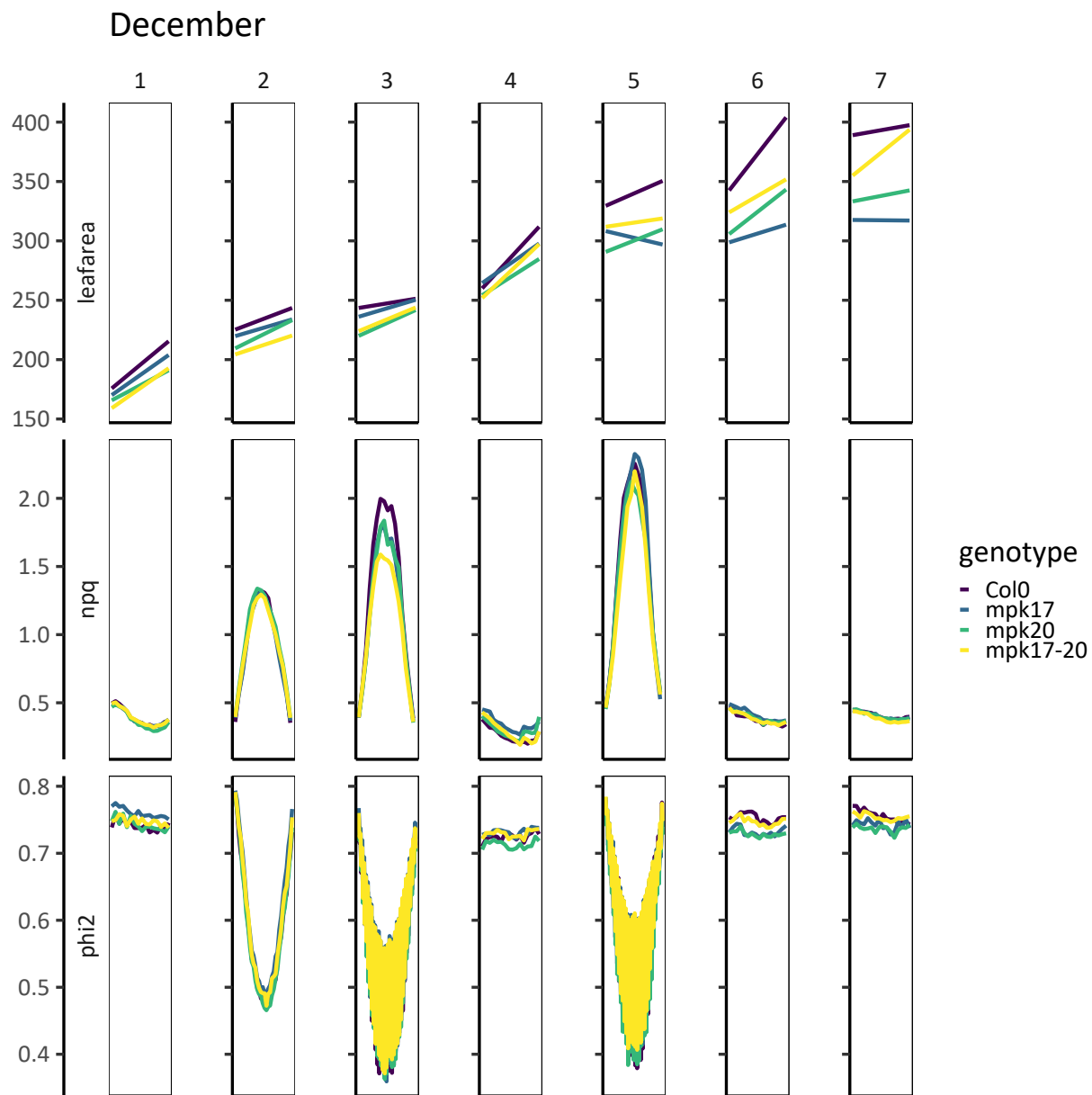
```

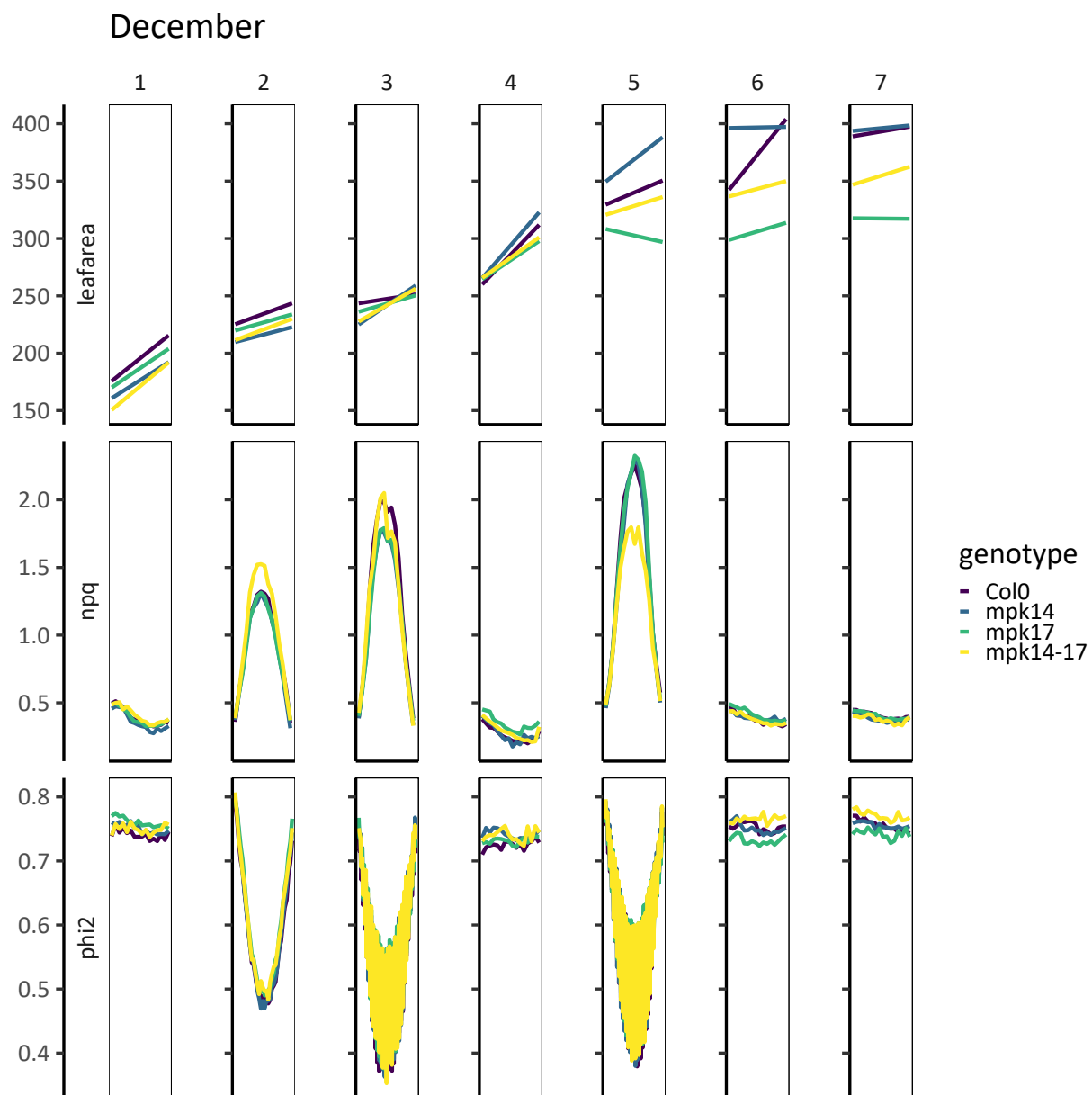


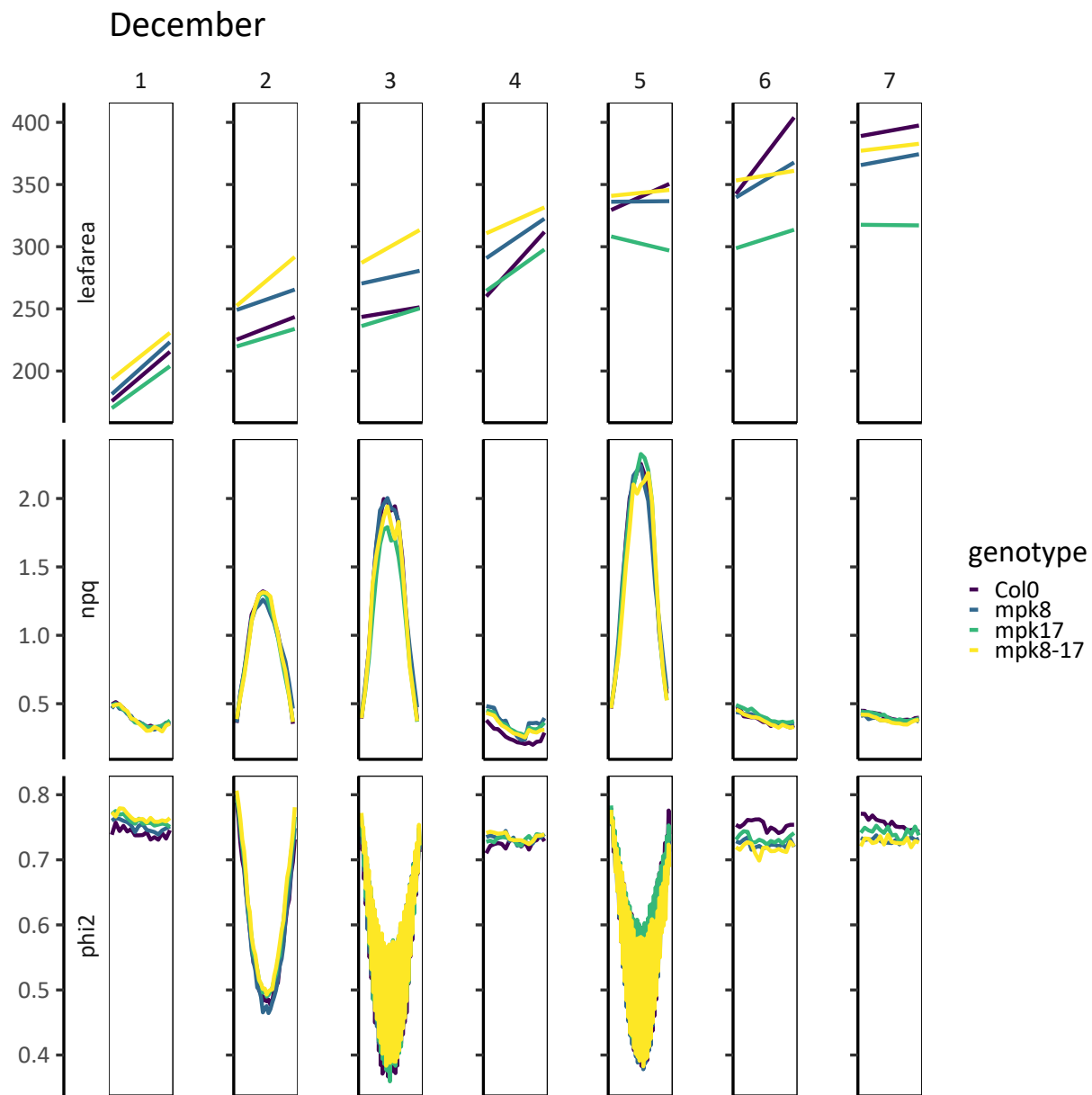
December

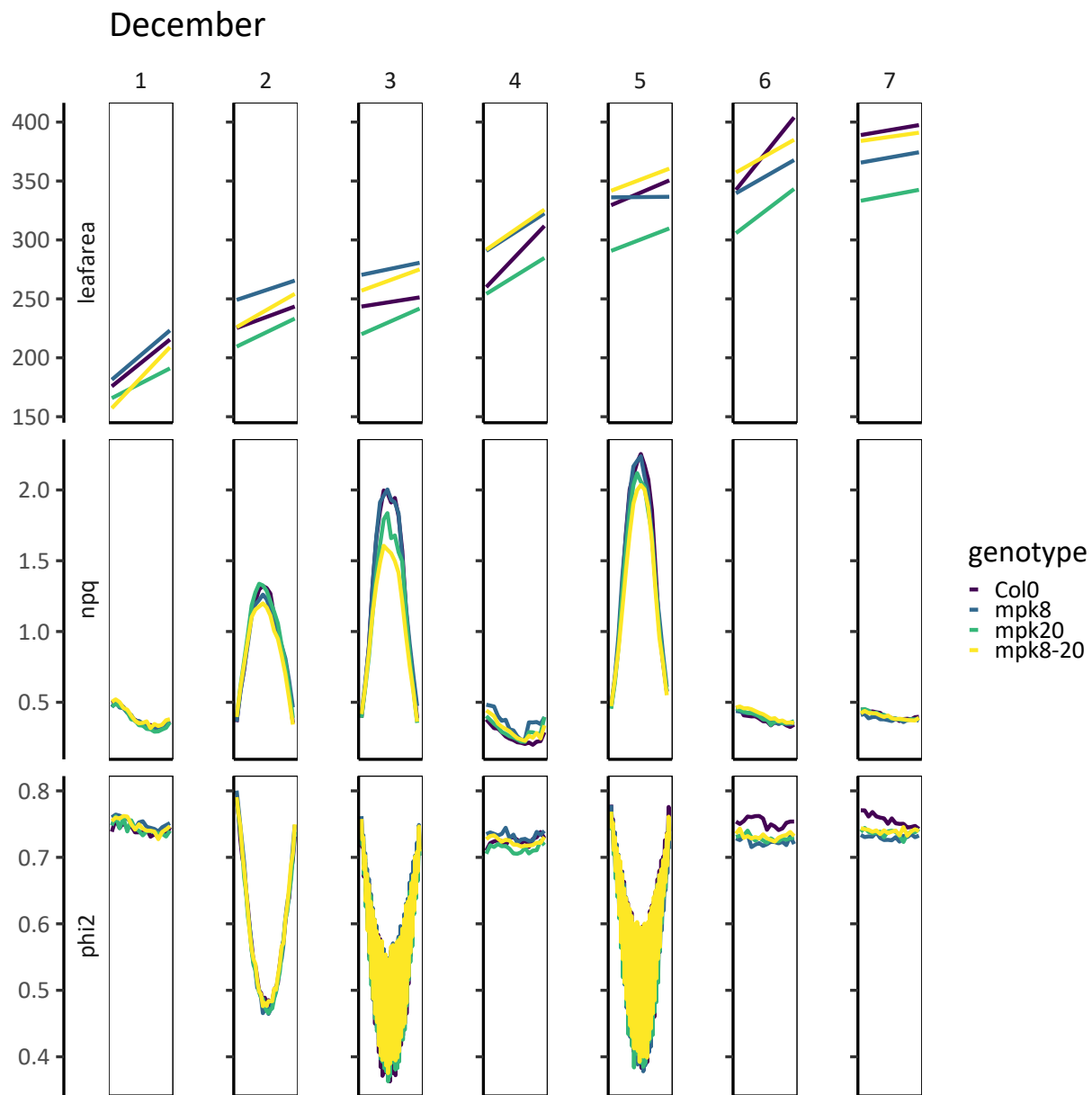


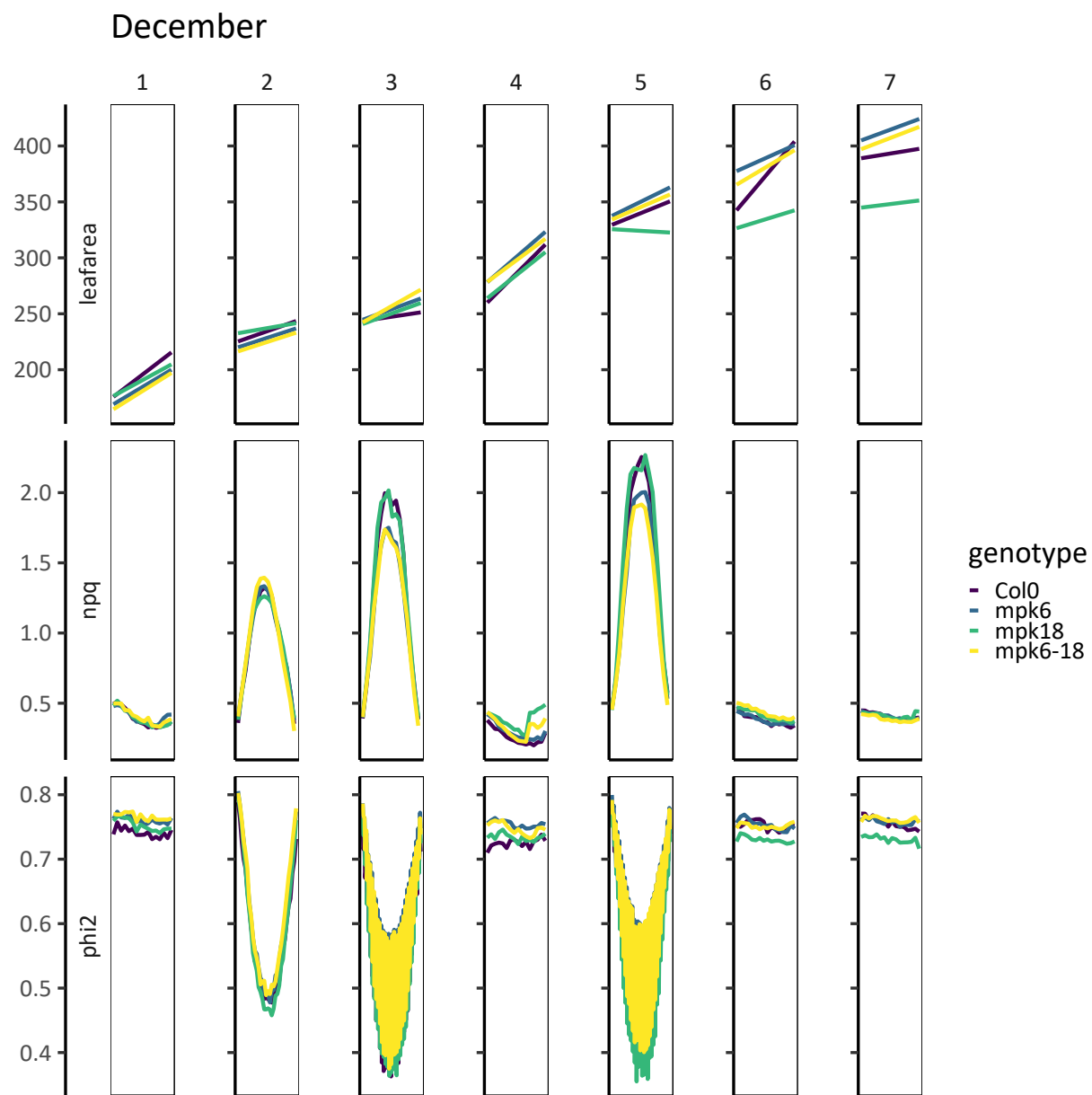


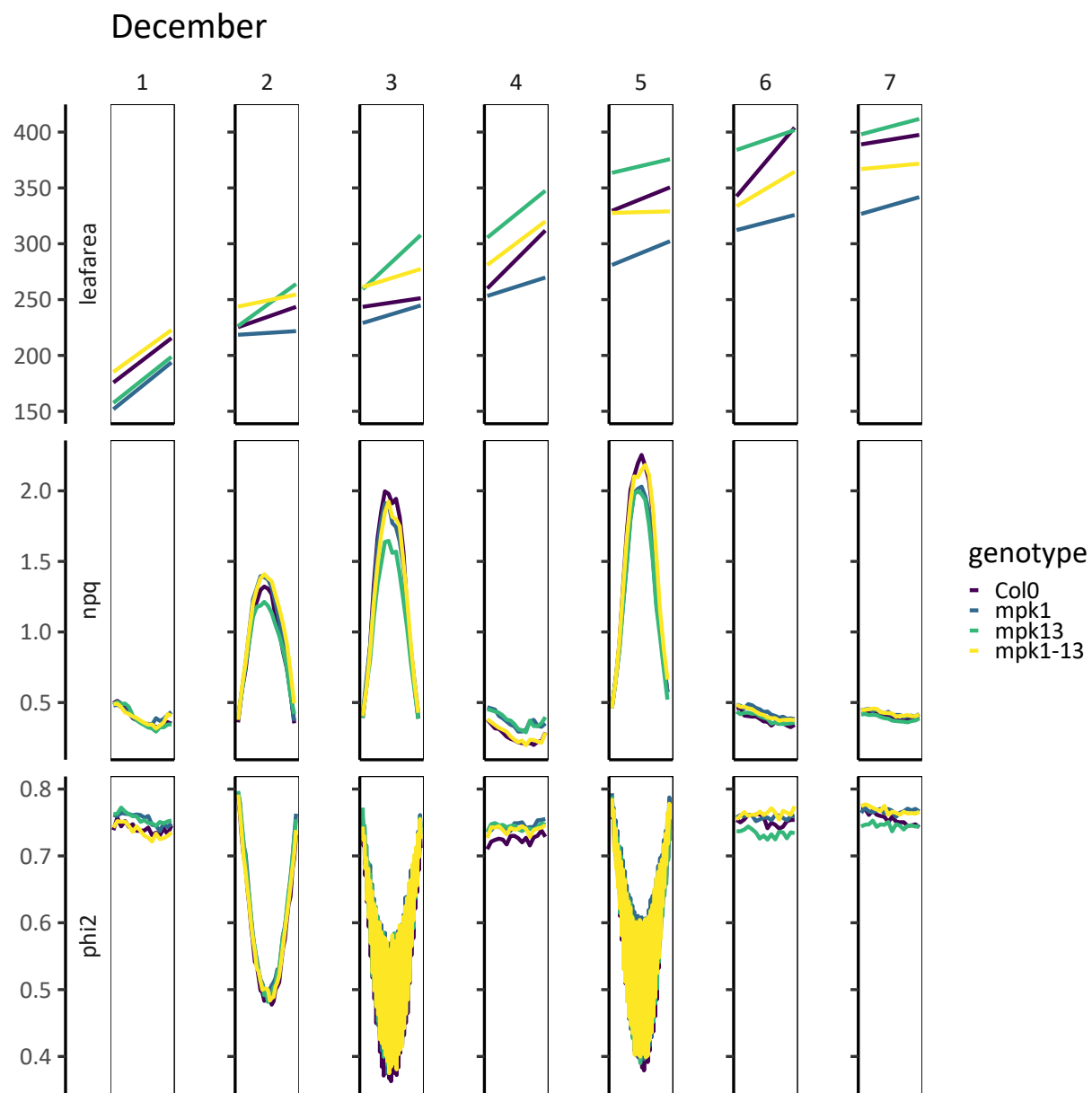


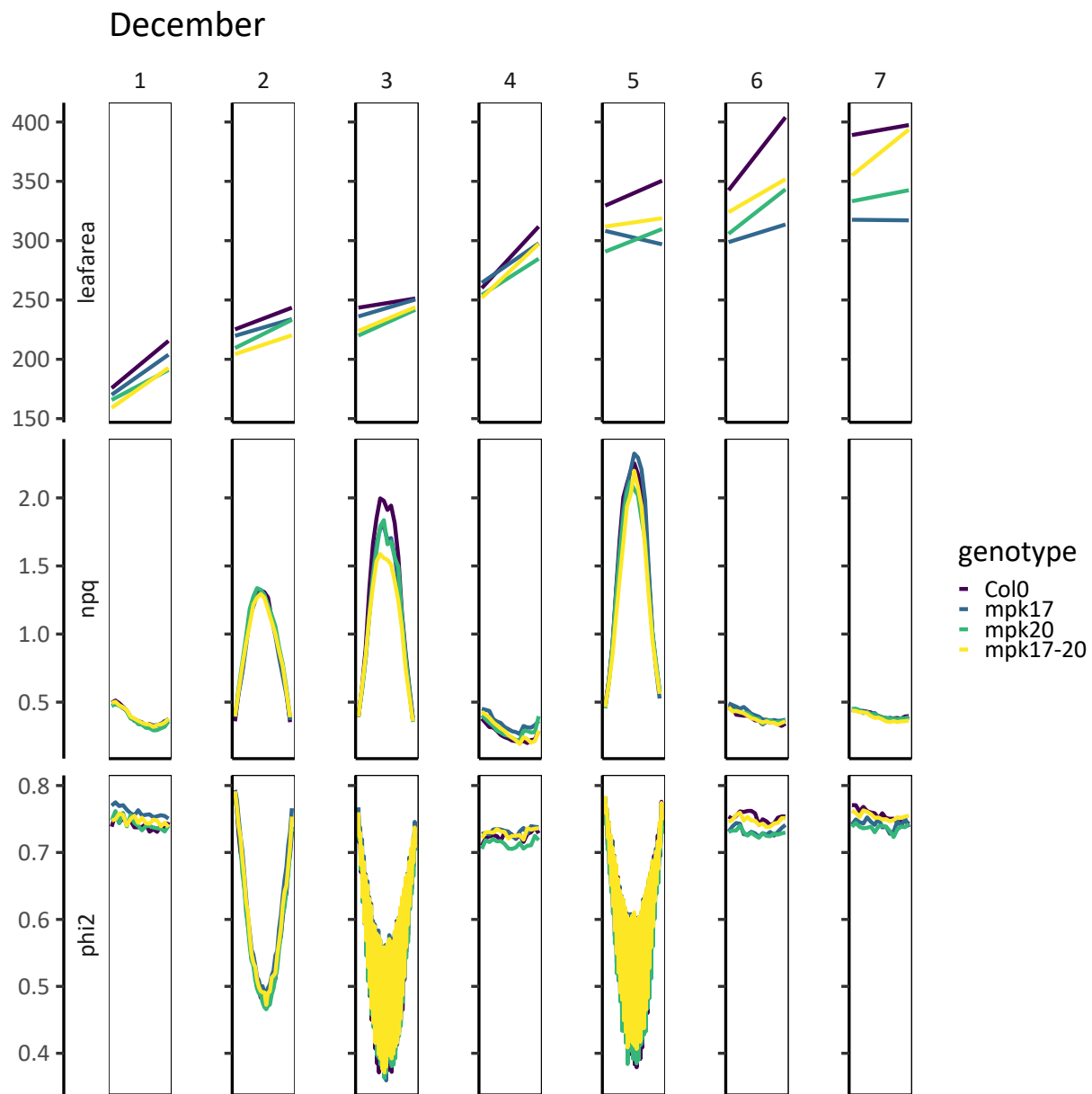


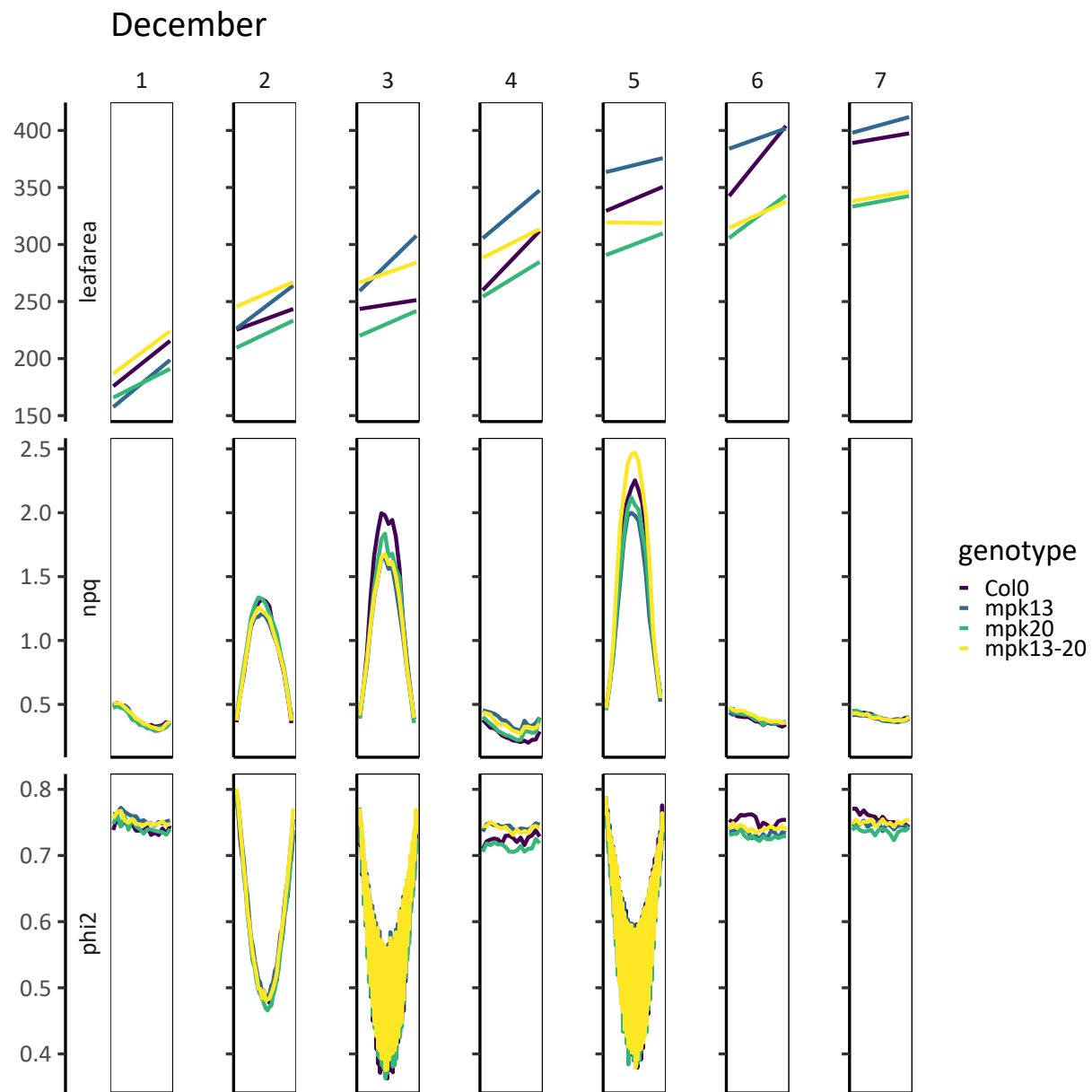


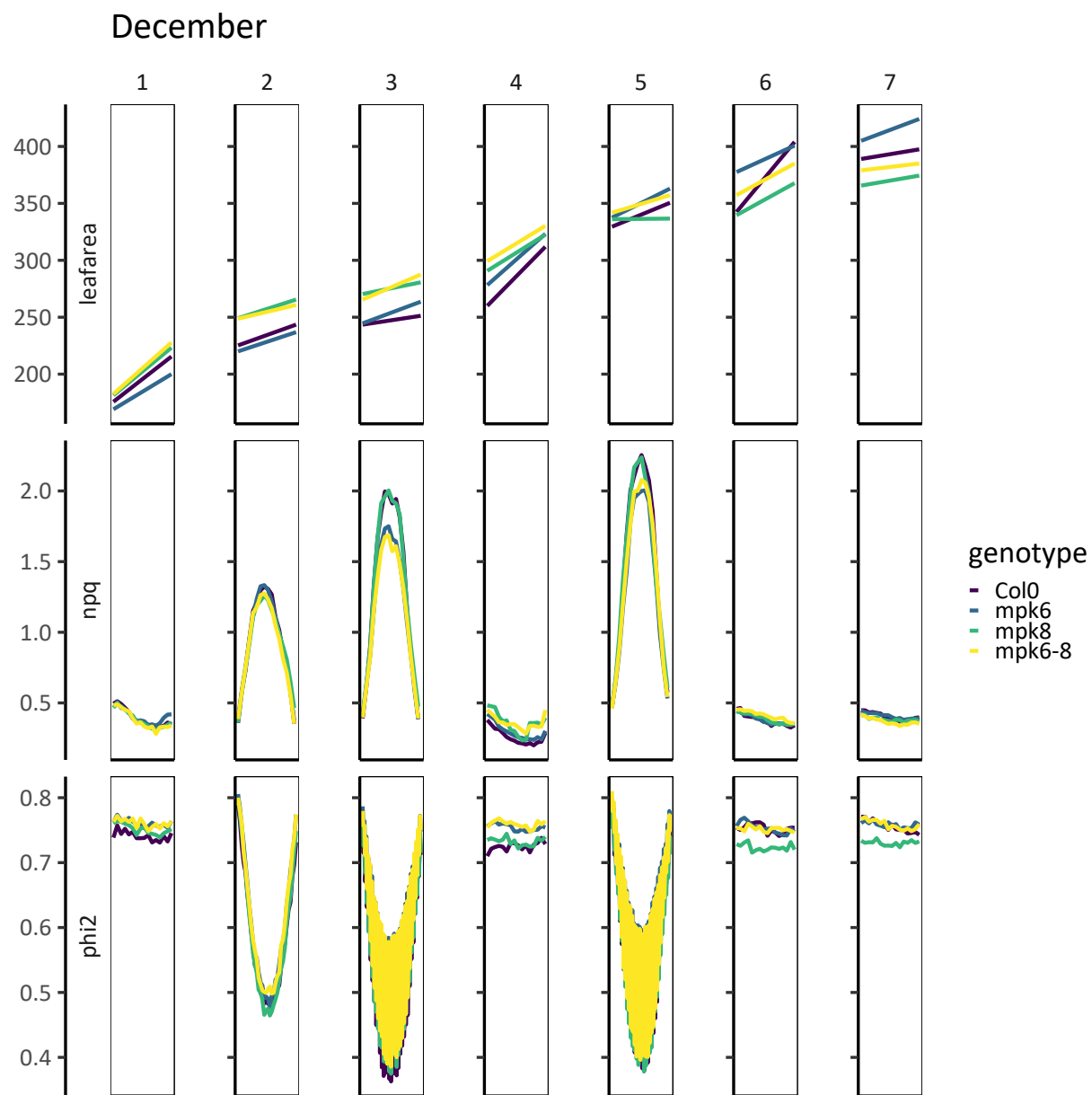


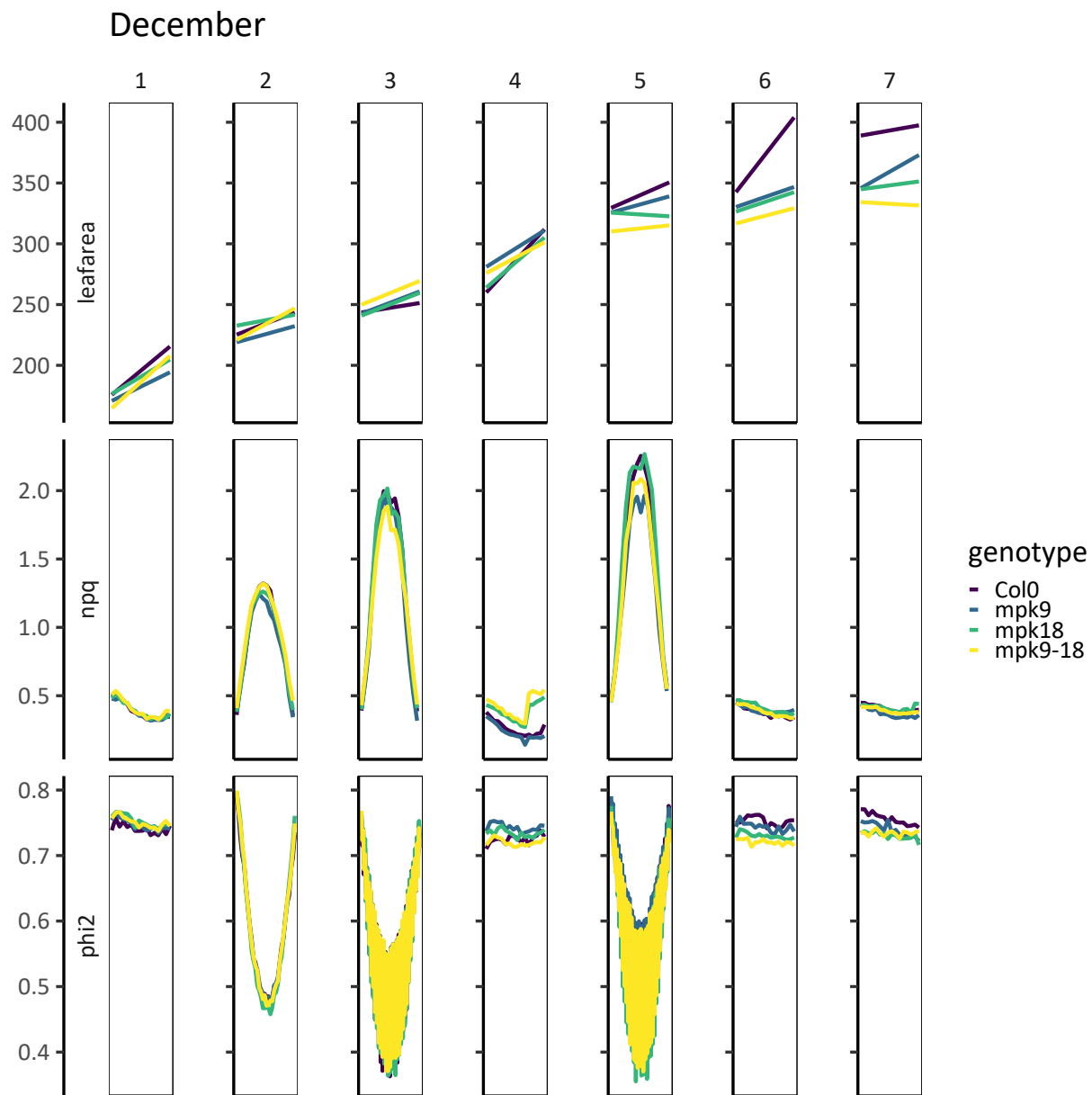


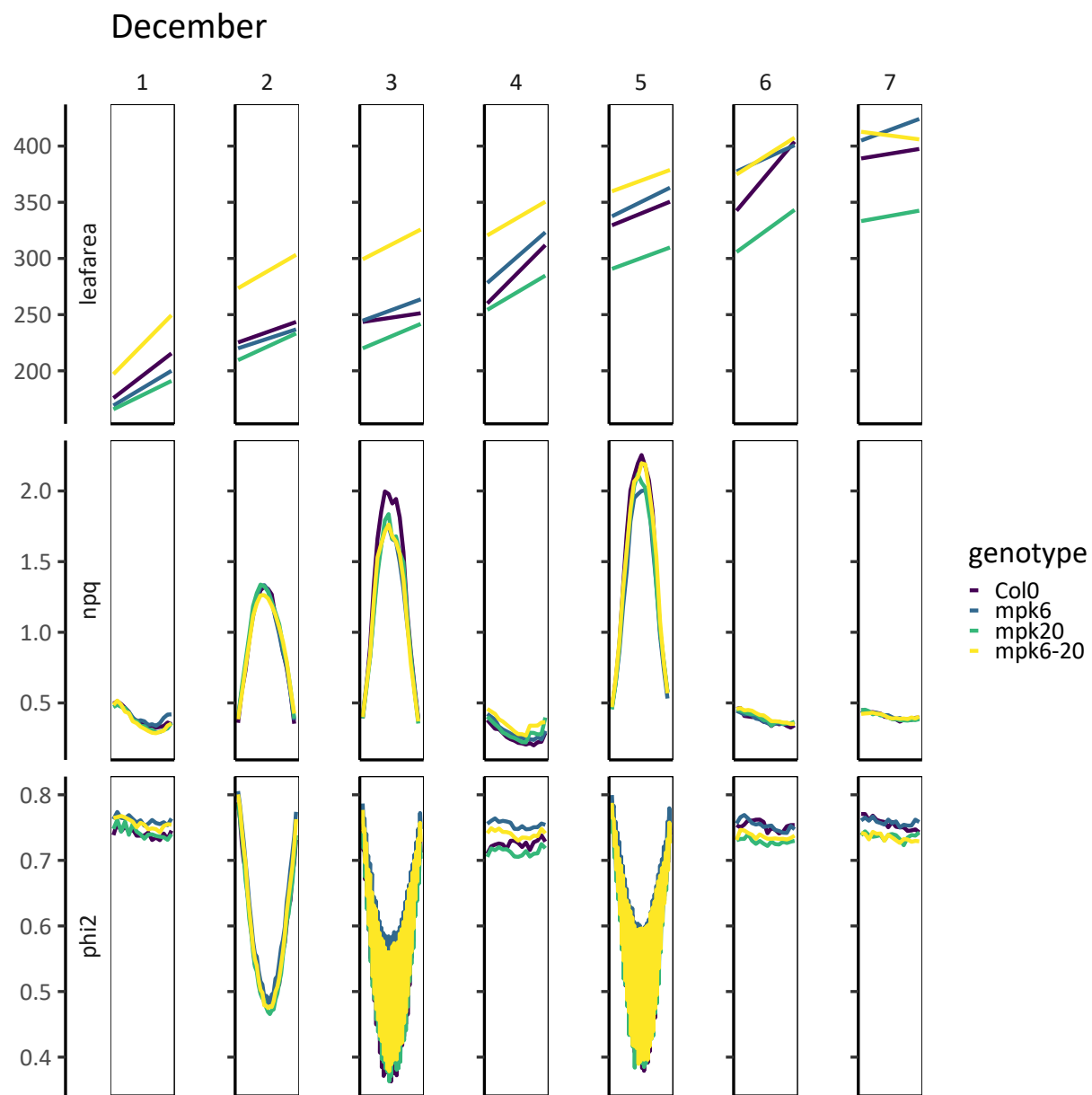


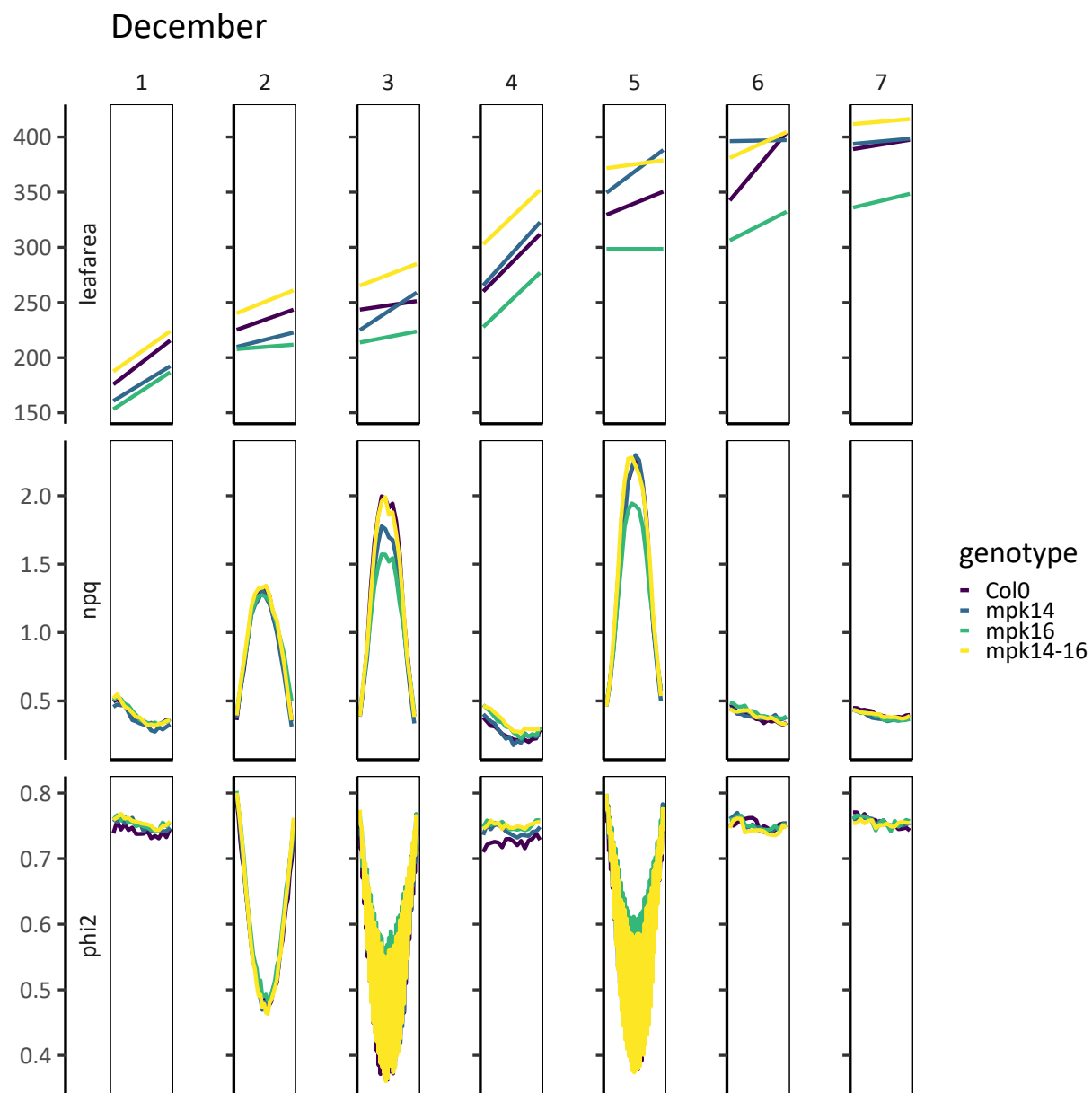


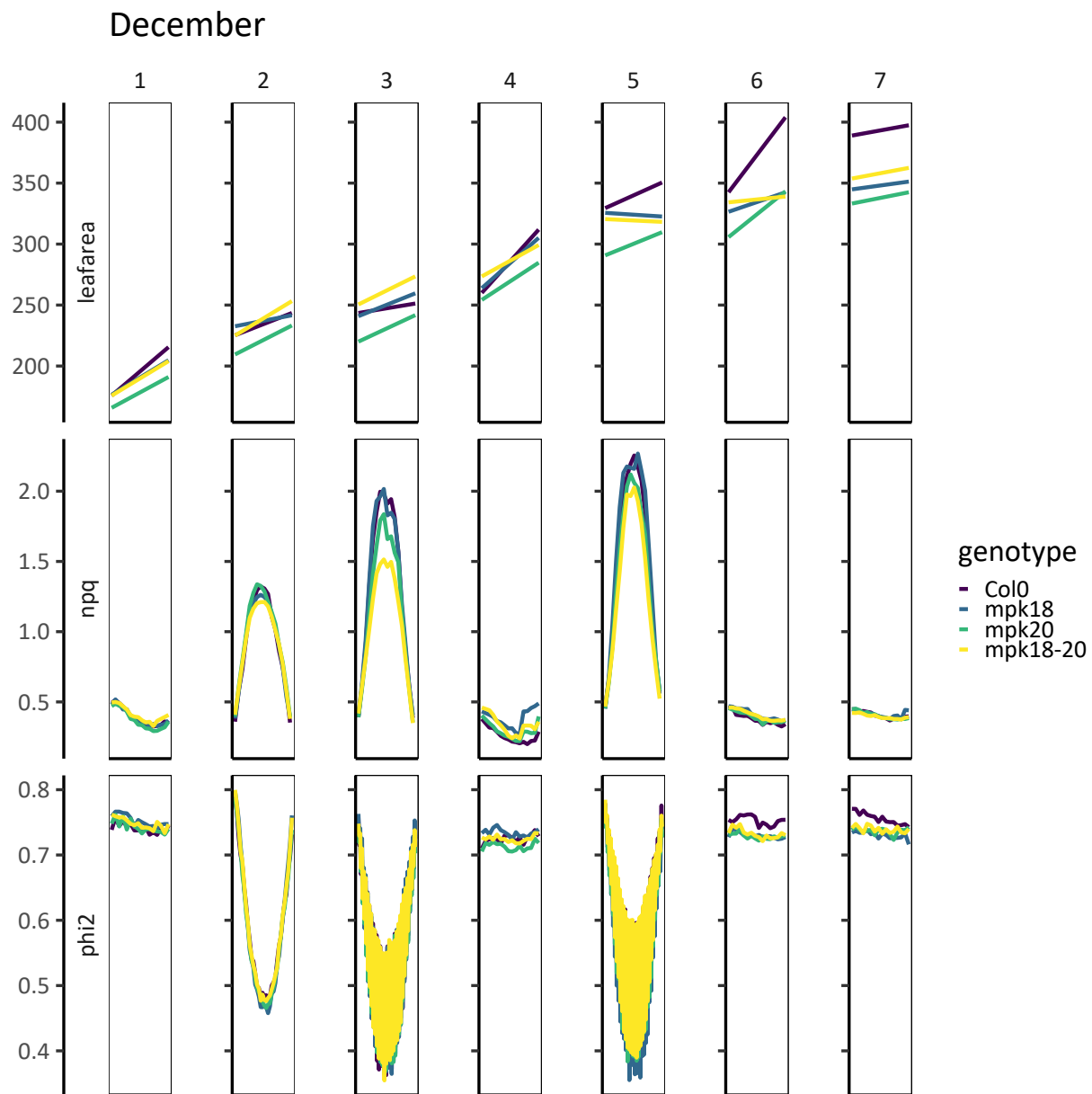


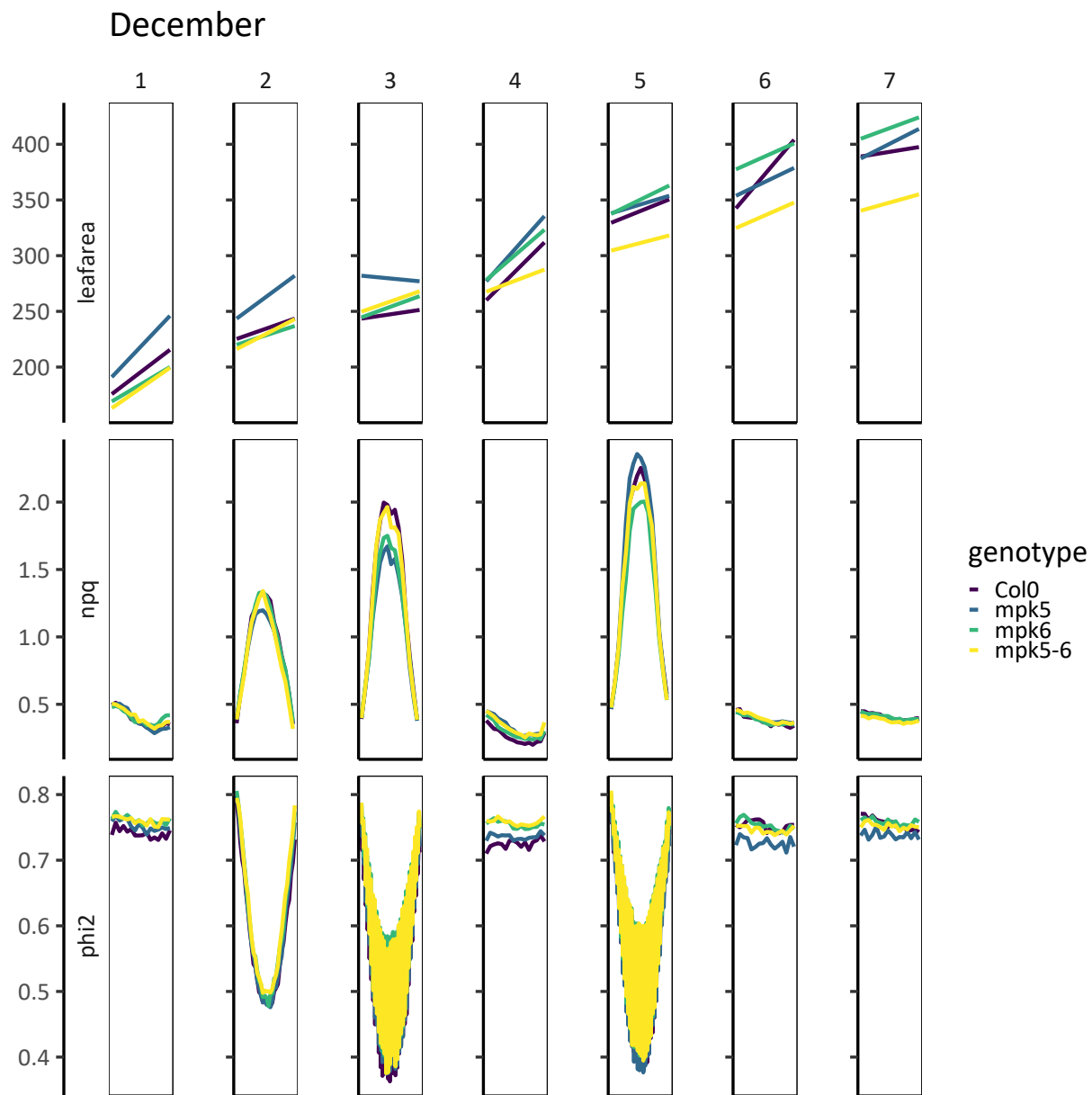


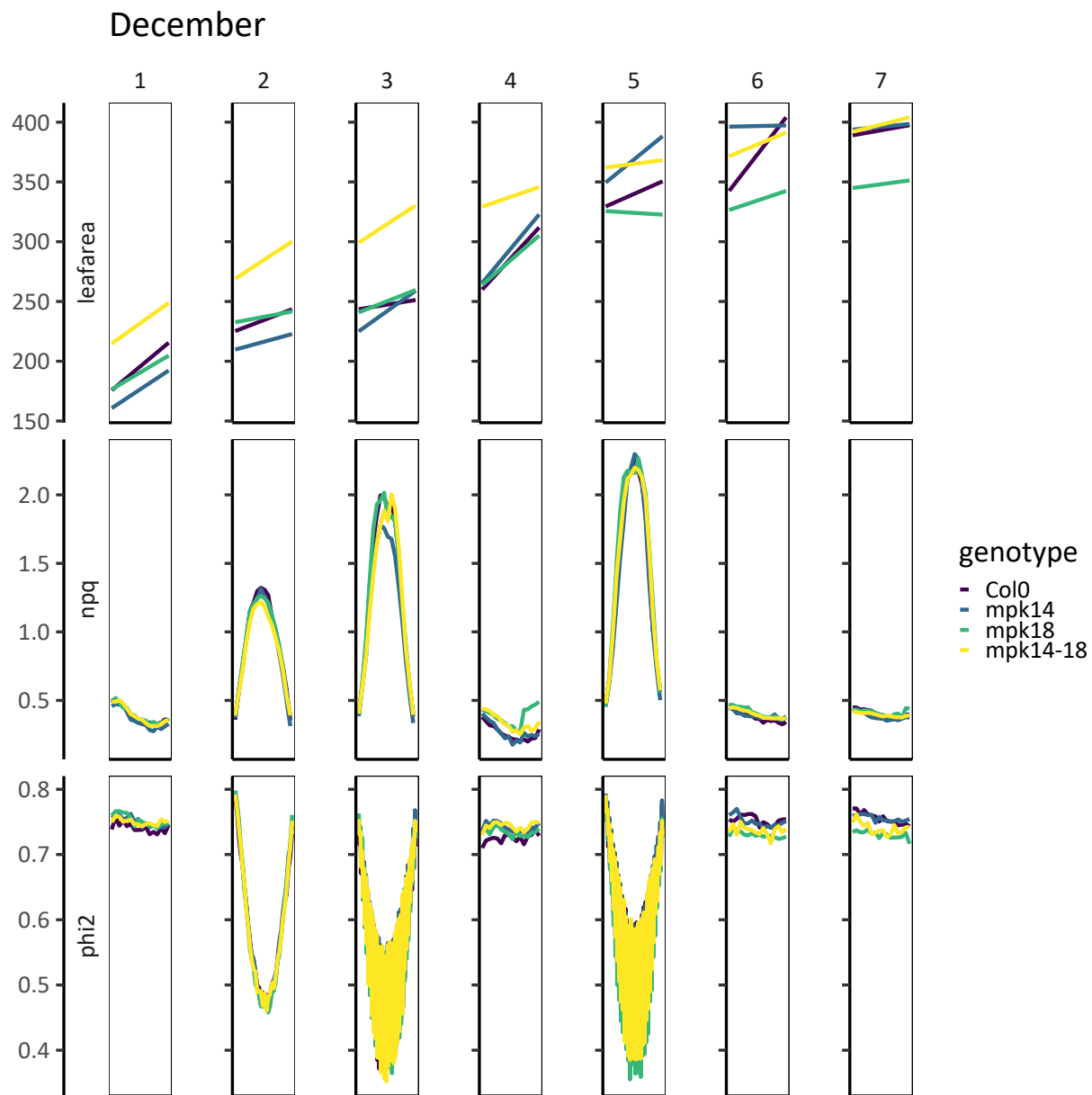


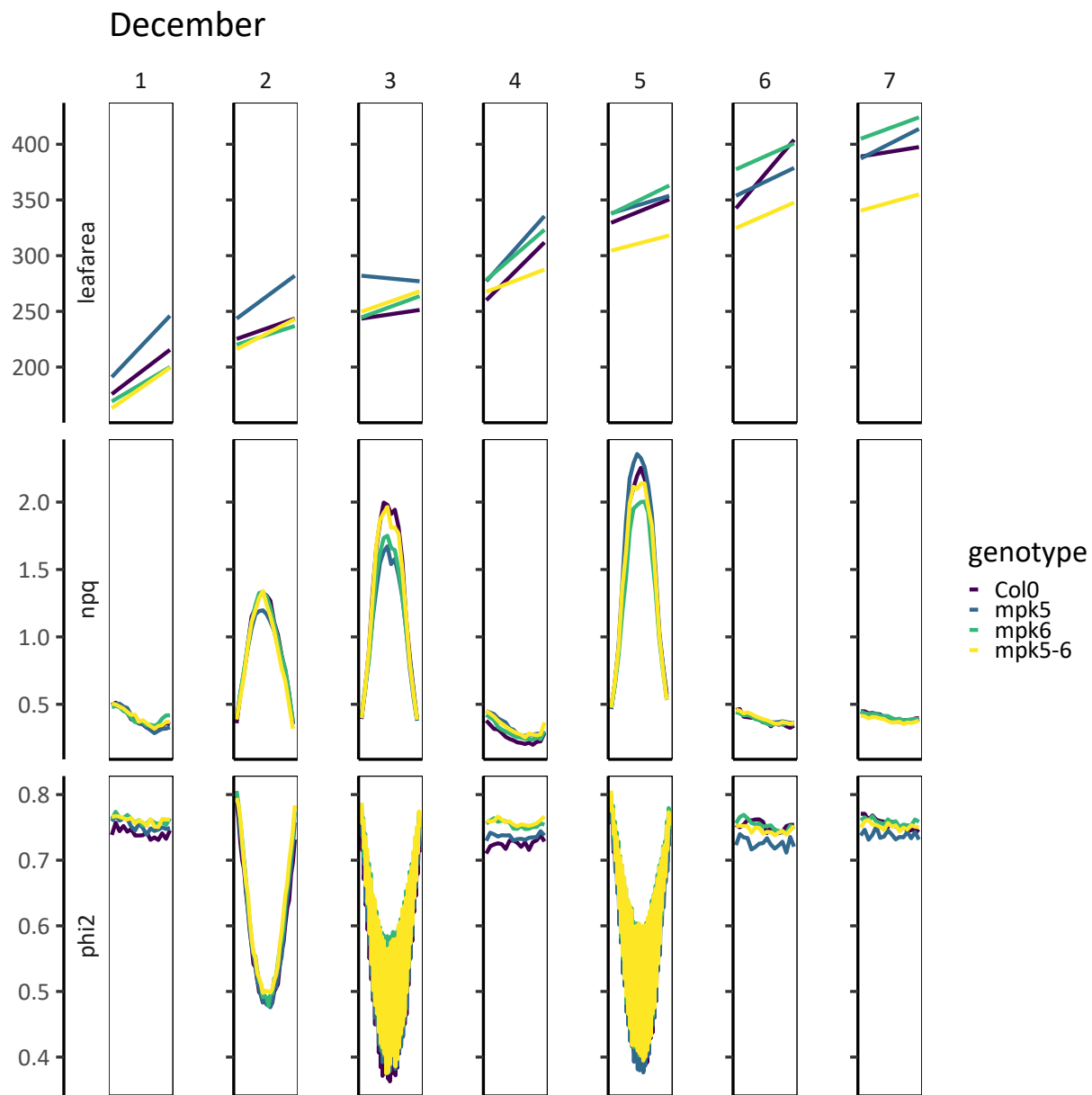


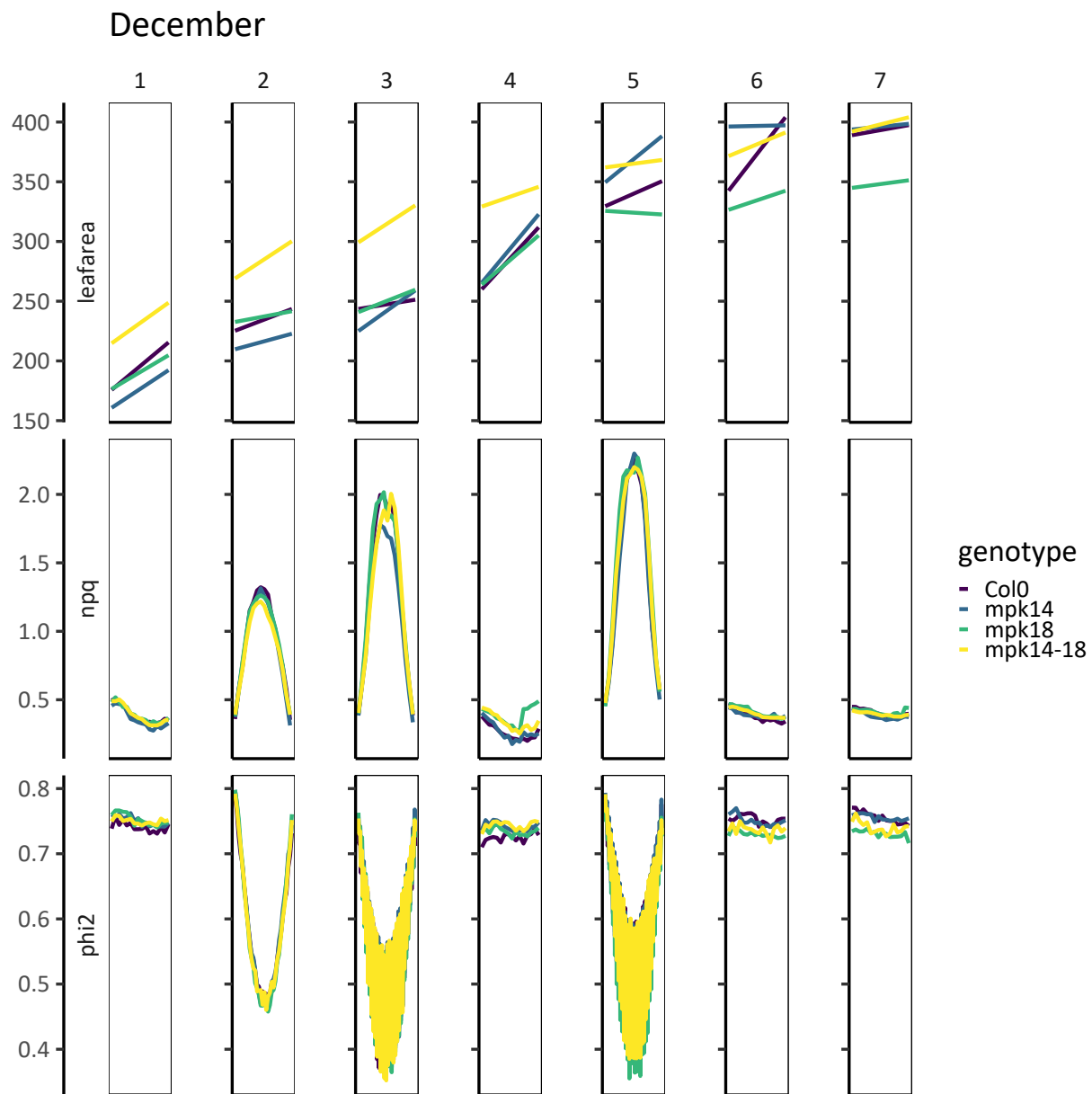


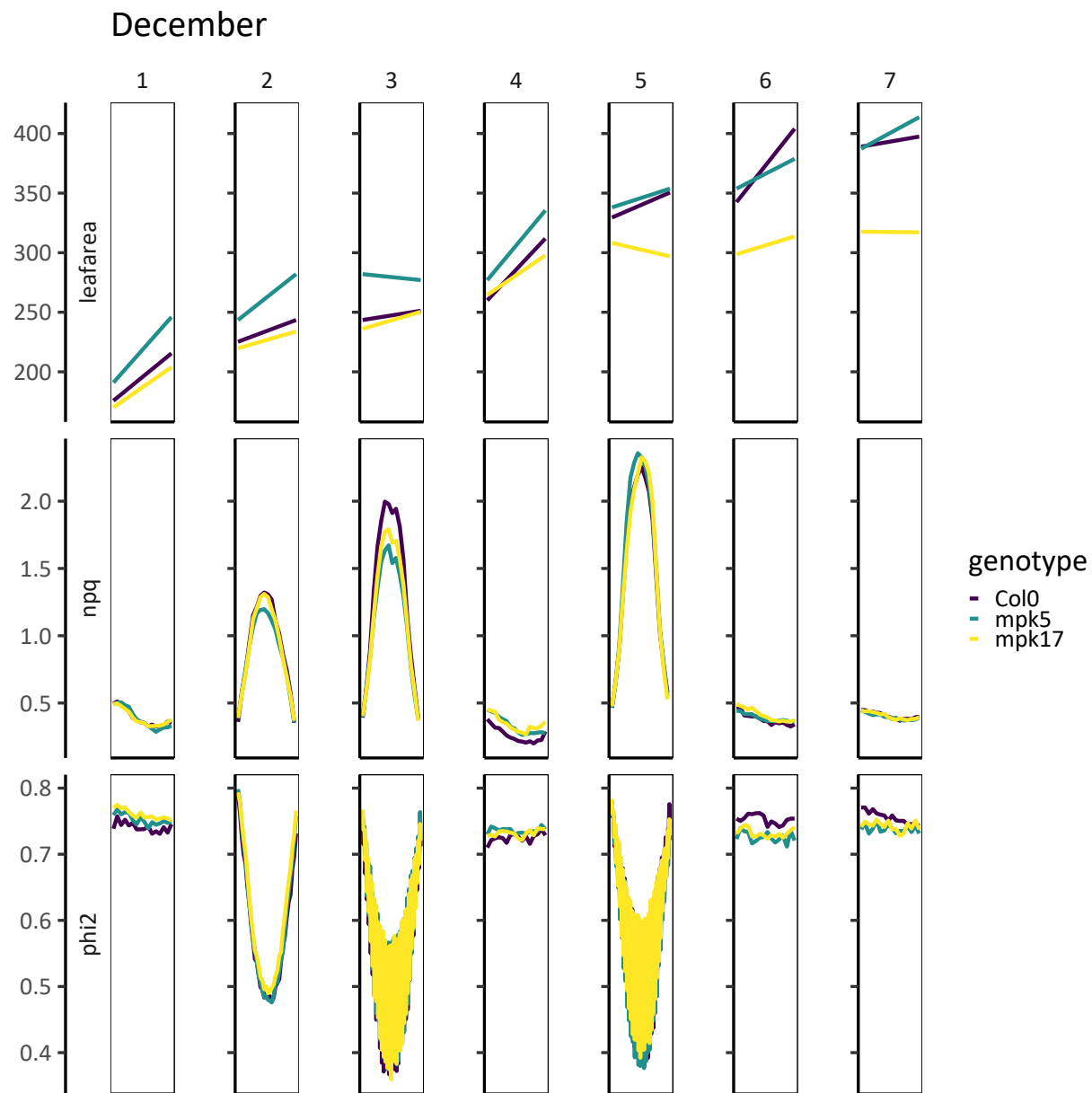


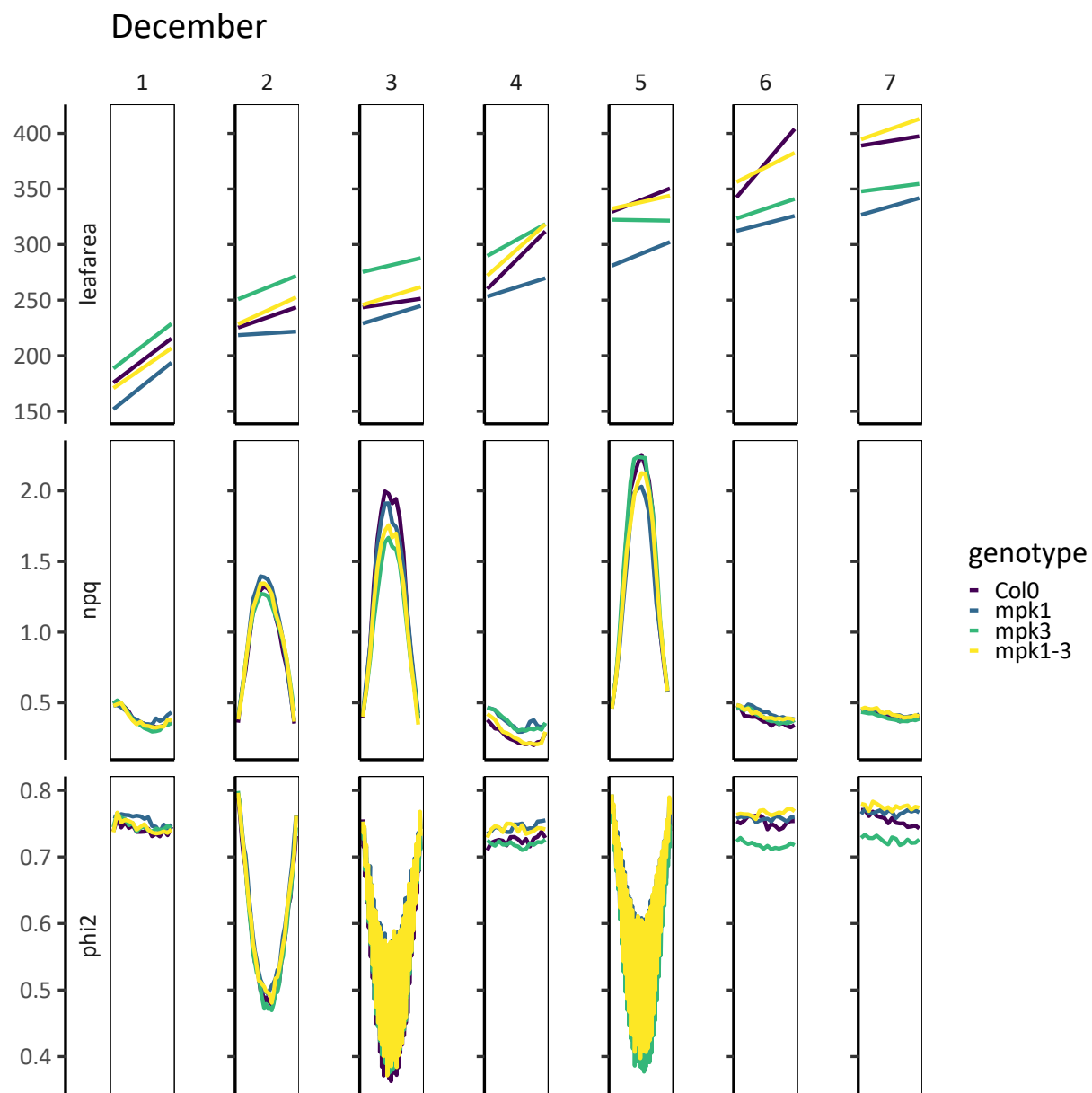


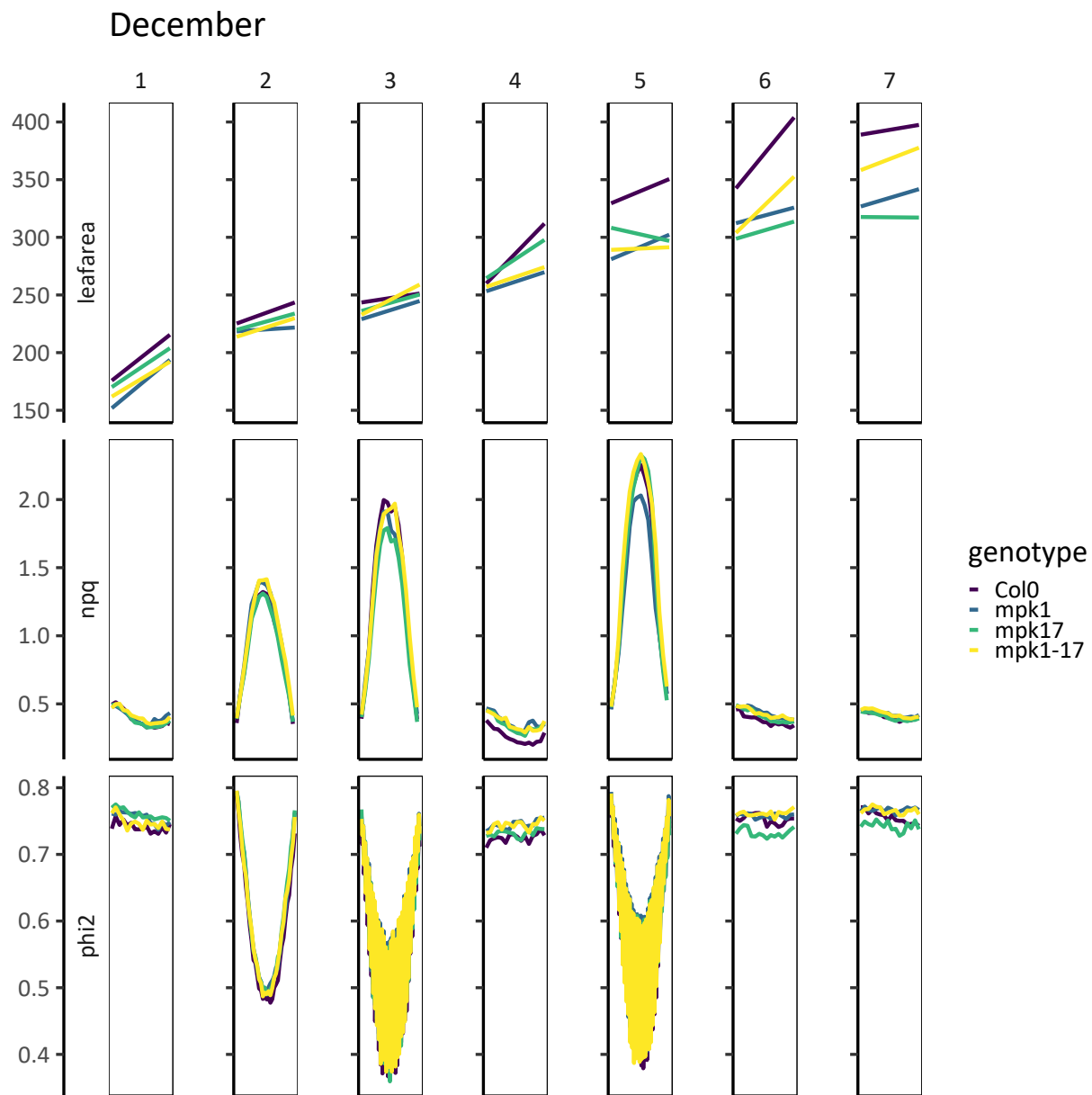


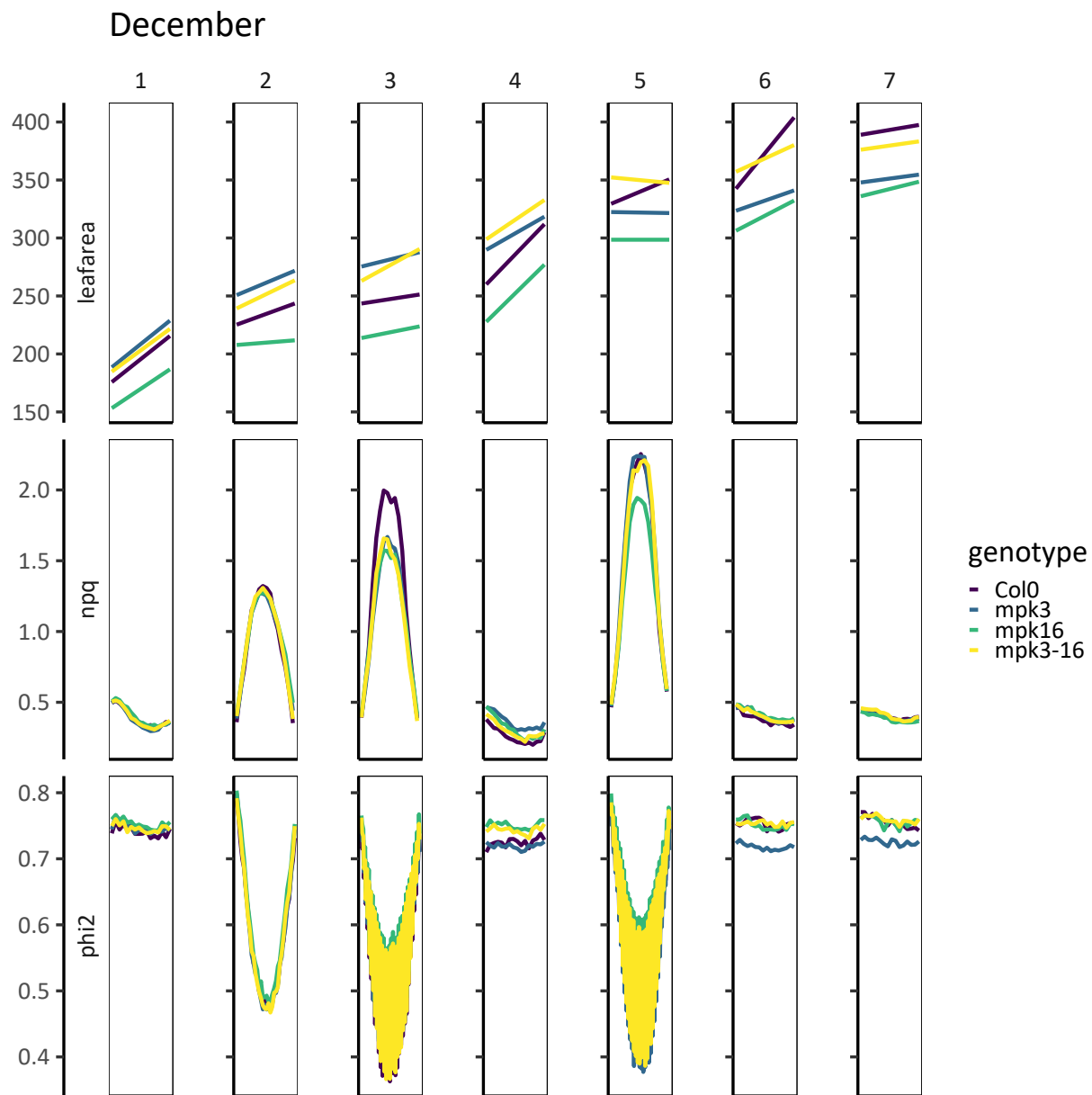


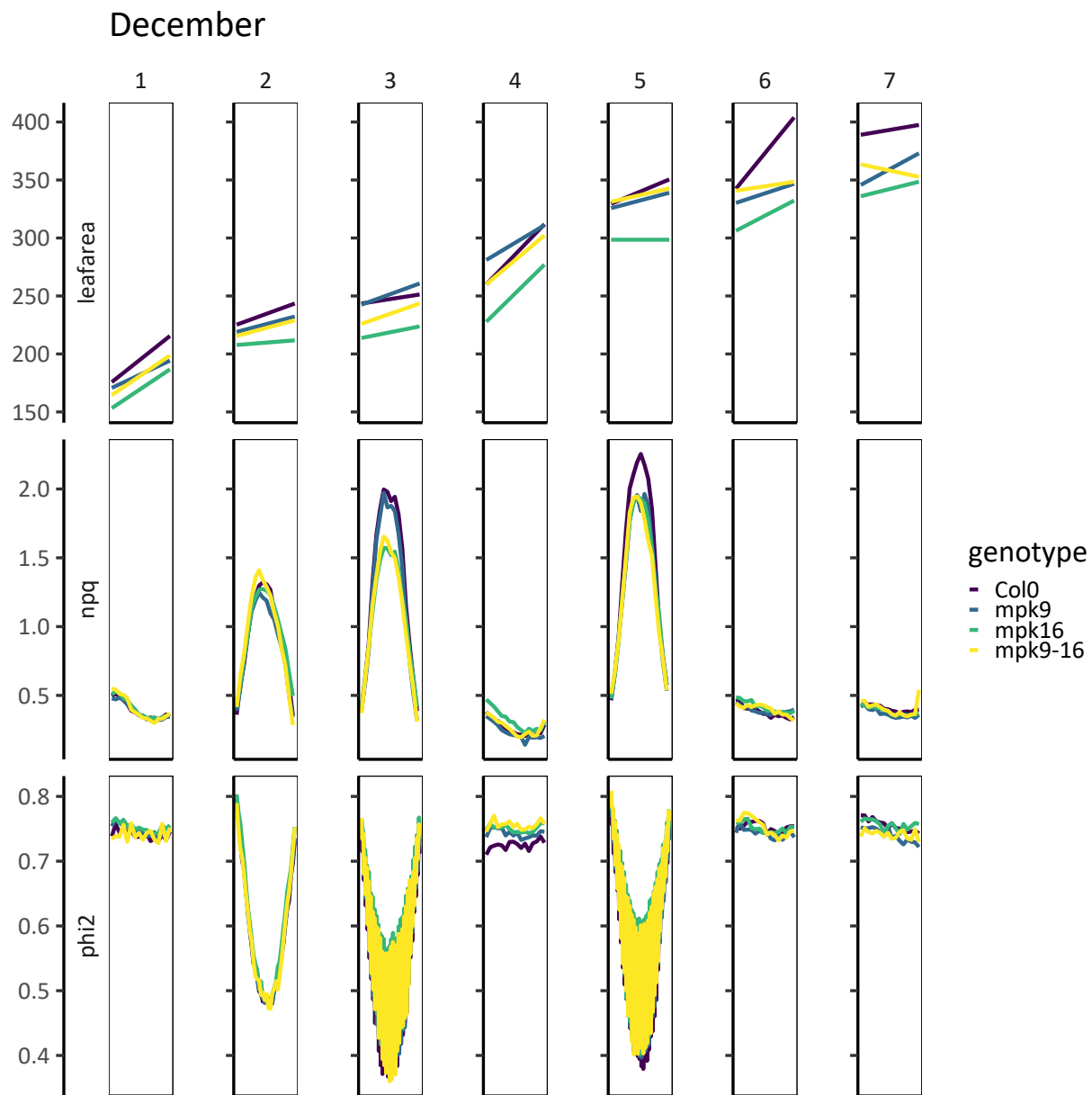




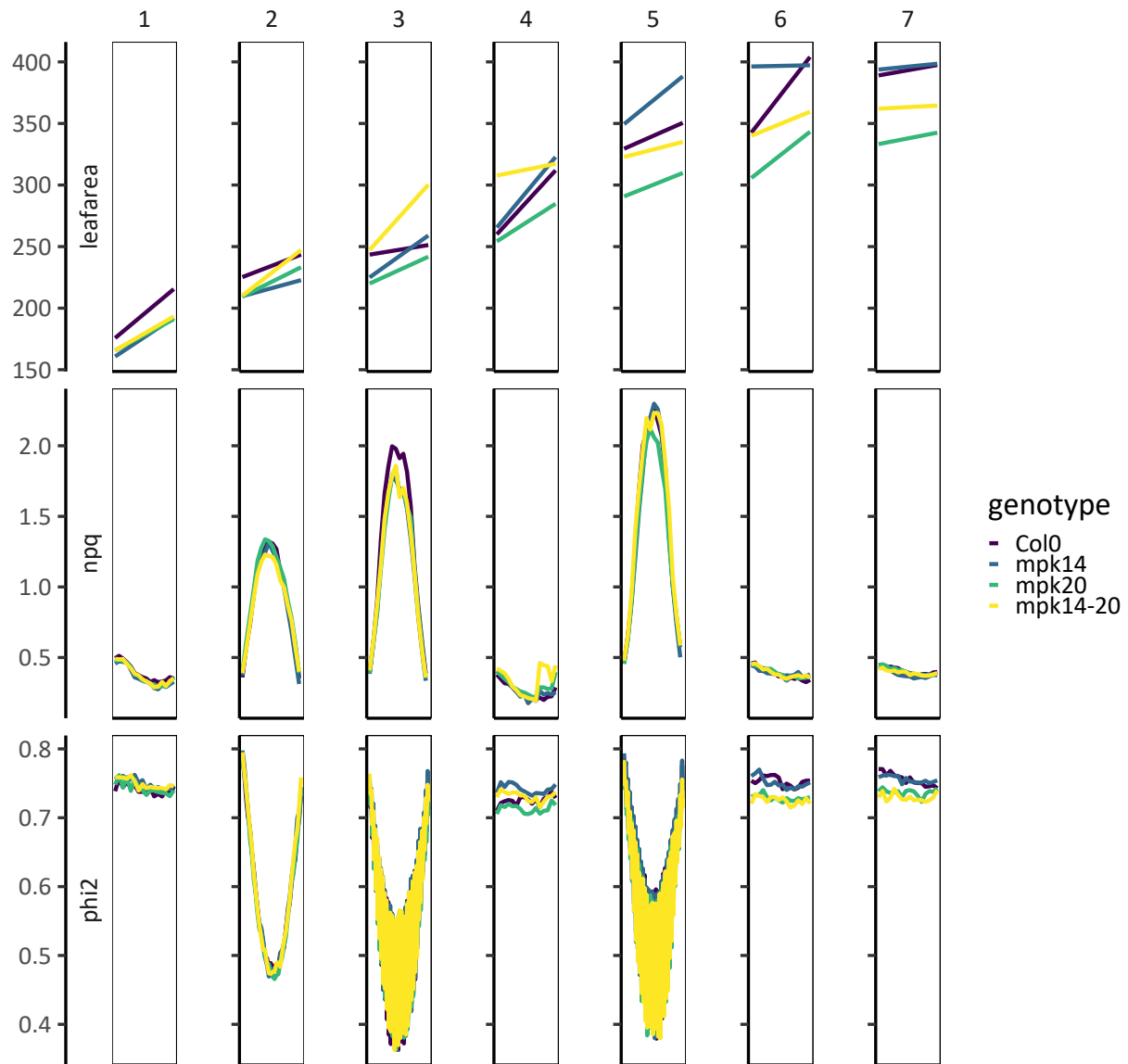








December



January

```
plot_data <- filter(depi_data, depi_data$month ==
  "Jan")

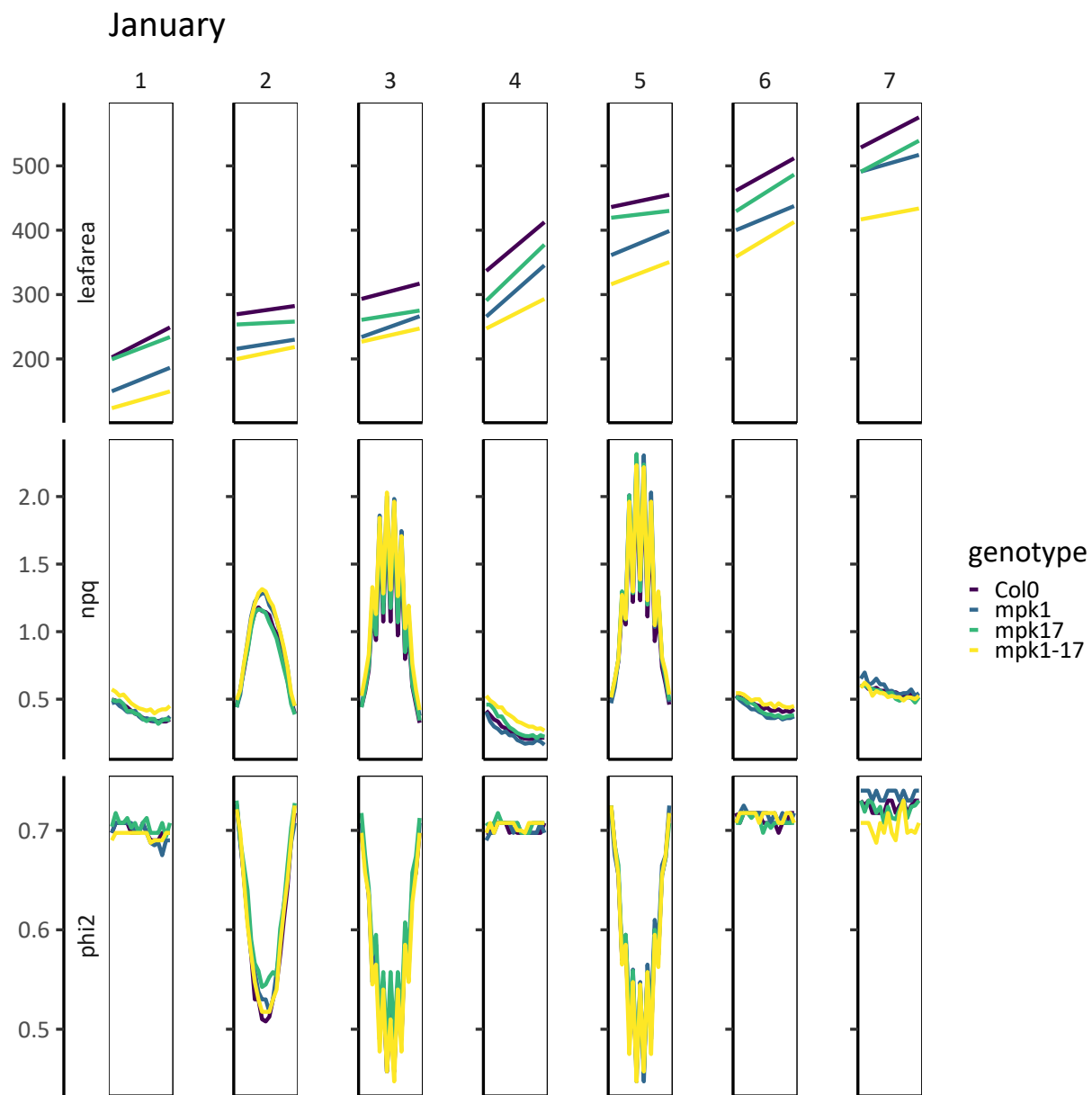
plot_data <- cell_370_data(plot_data)
plot_data <- add_number(plot_data)
plot_data <- add_day_col(plot_data)
plot_data$genotype <- reorder(plot_data$genotype,
  plot_data$number)

for (element in genotype_combinations) {
```

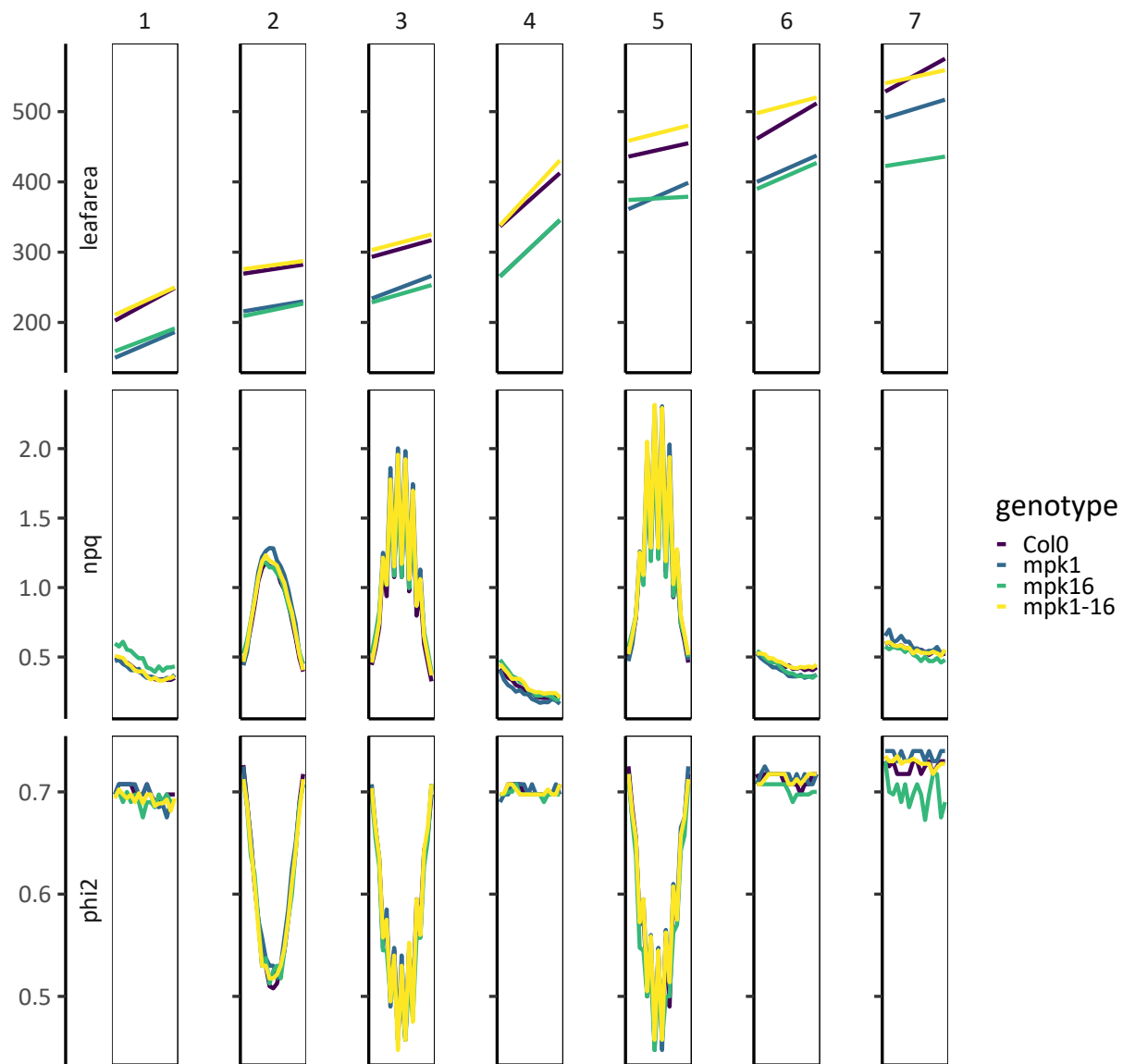
```

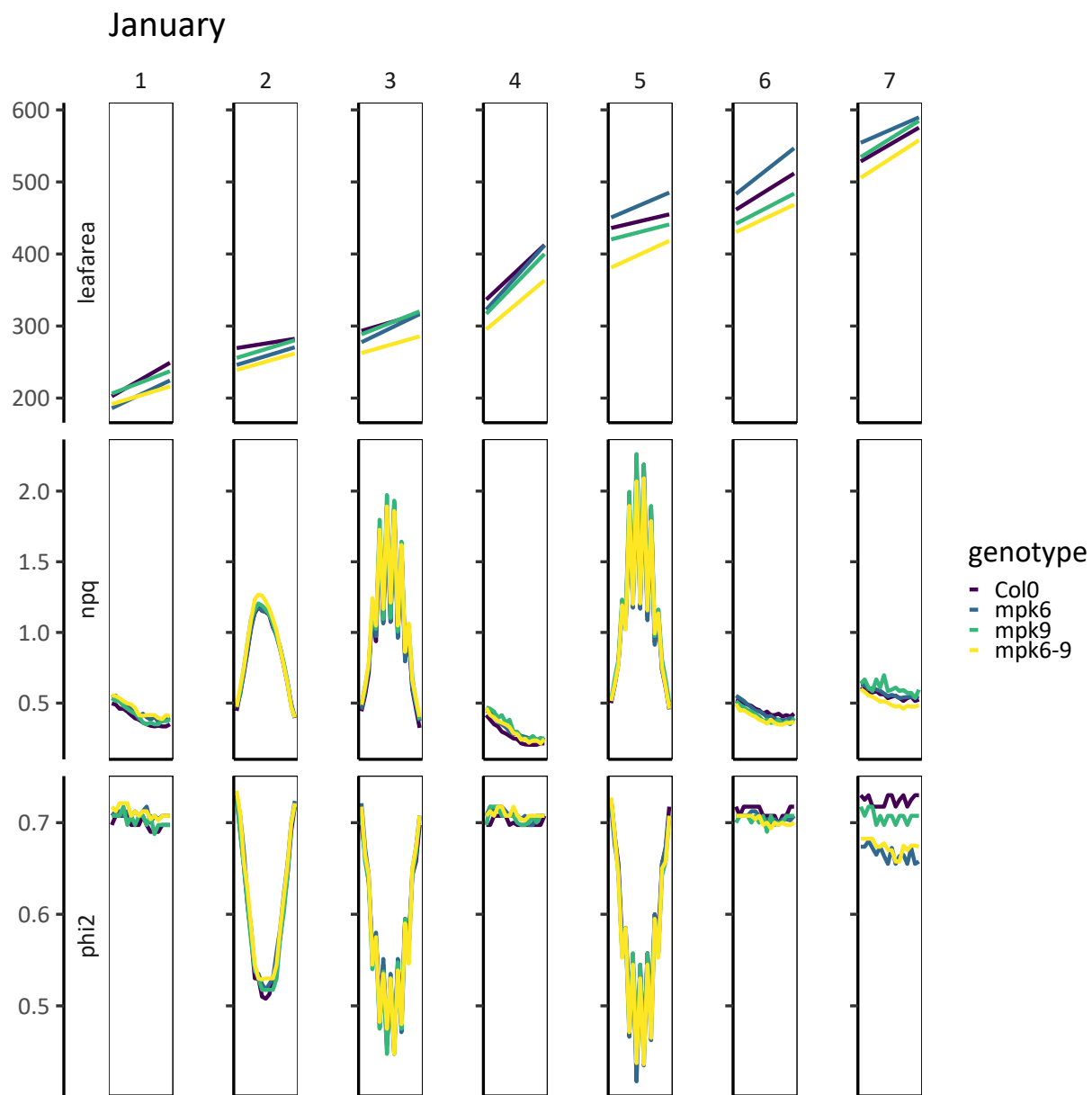
data <- filter(plot_data, genotype %in%
  c(element, "Col0"))
plot <- ggplot(data = data, aes(x = time_point,
  y = med)) + geom_line(aes(color = genotype),
  size = 3) + facet_rep_grid(measurement ~
  day, scales = "free", switch = "y",
  repeat.tick.labels = FALSE) + labs(x = "Hours",
  y = NULL, title = "January") + theme_tufte(base_family = "Calibri",
  base_size = 50) + theme(strip.background.x = element_blank(),
  axis.title.x = element_blank(), axis.text.x = element_blank(),
  axis.ticks.x = element_blank(), panel.border = element_rect(color = "black",
  fill = NA, size = 1), axis.line = element_line(),
  panel.spacing = unit(1, "lines")) +
  scale_color_viridis_d(begin = 0,
  end = 1, option = "viridis",
  aesthetics = c("colour", "fill"))
print(plot)
}

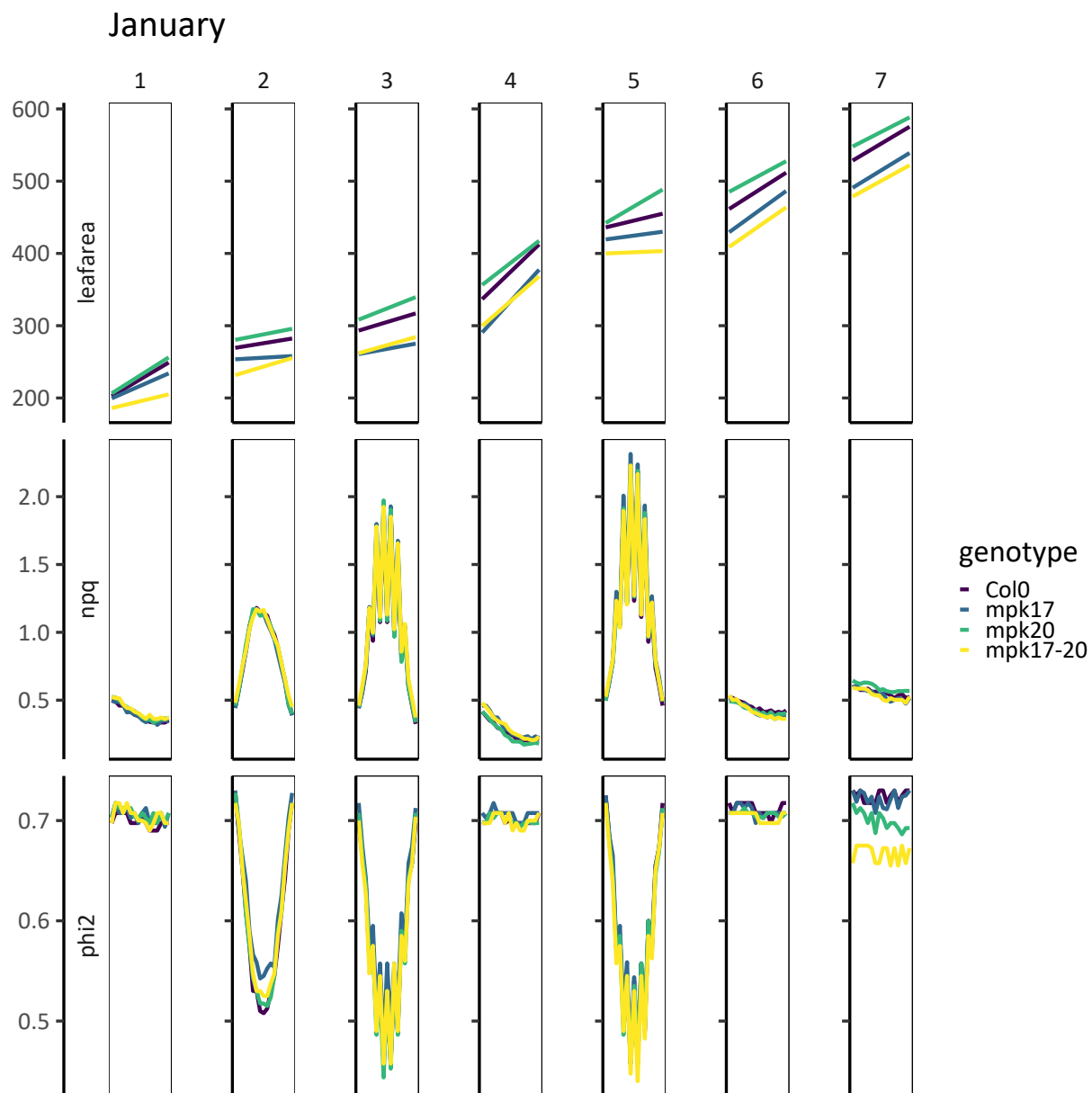
```

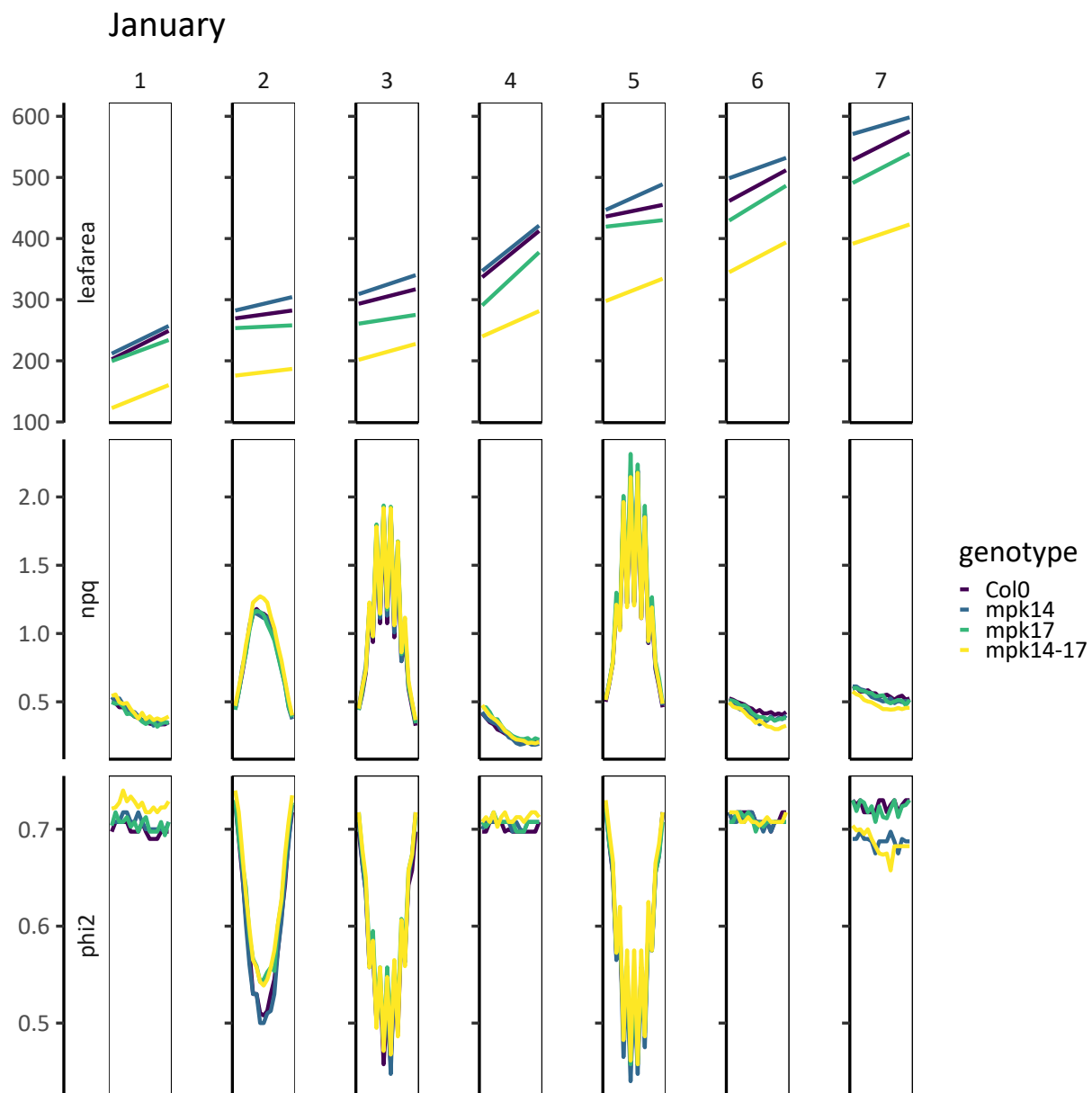



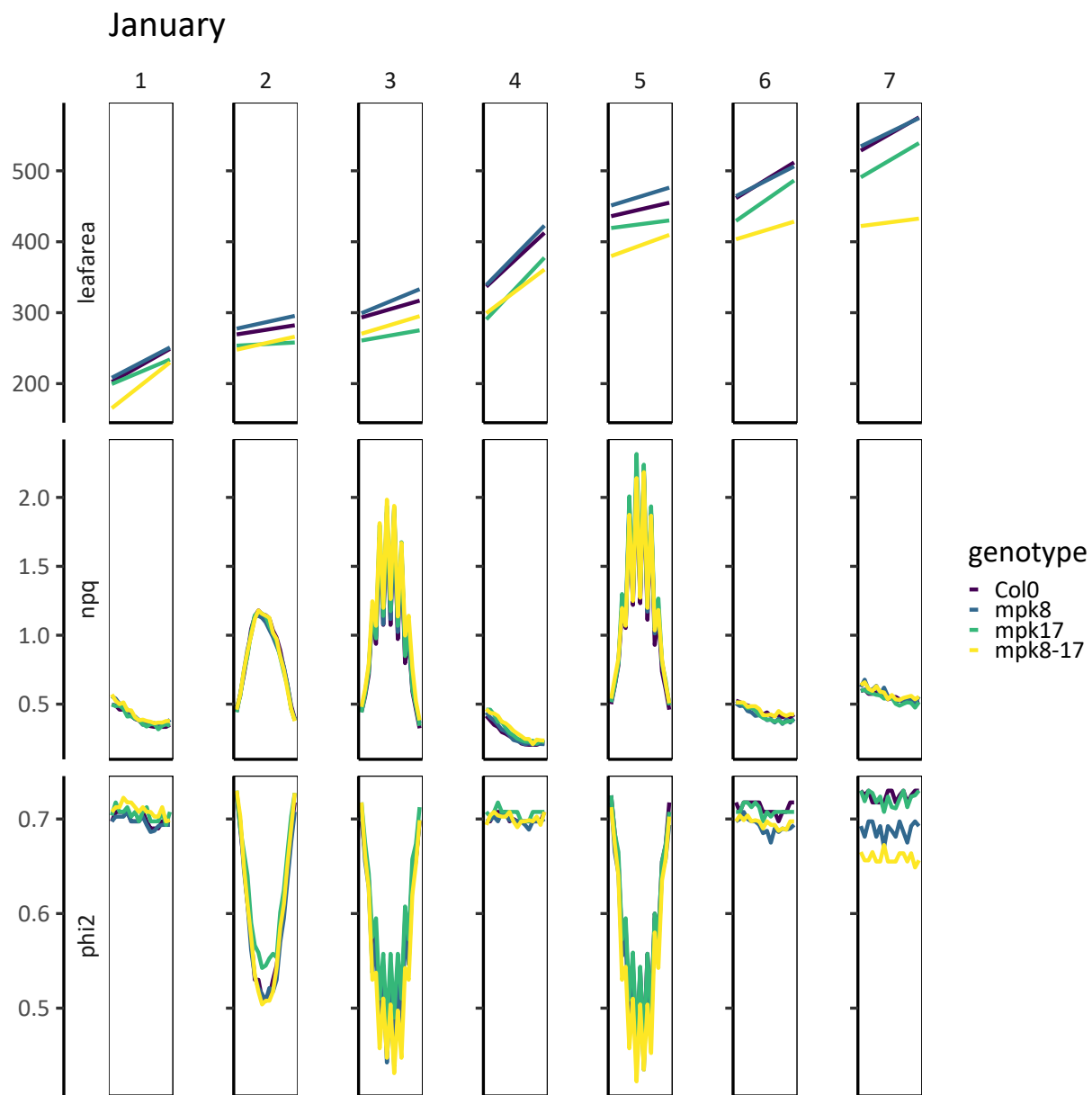
January

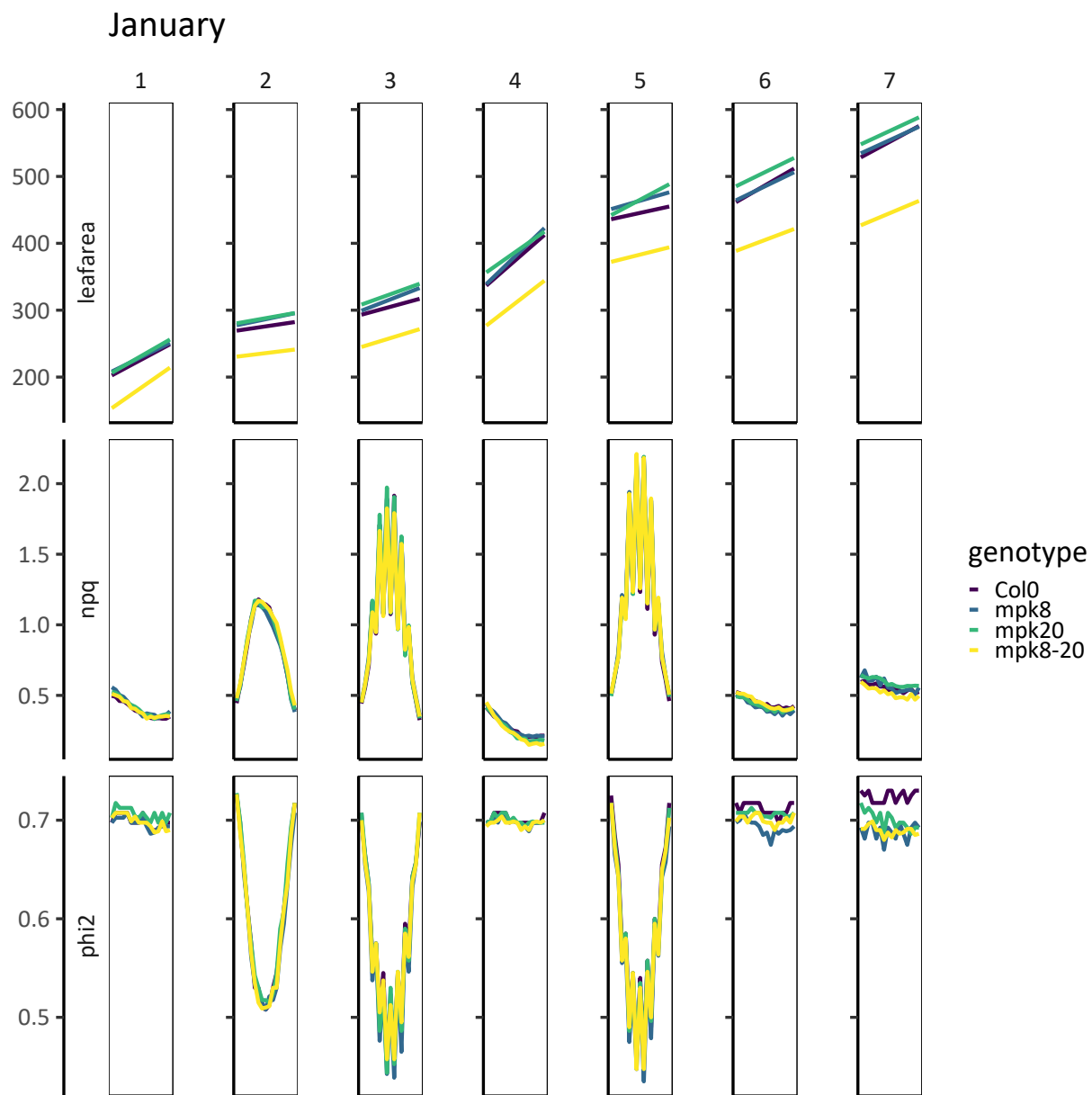


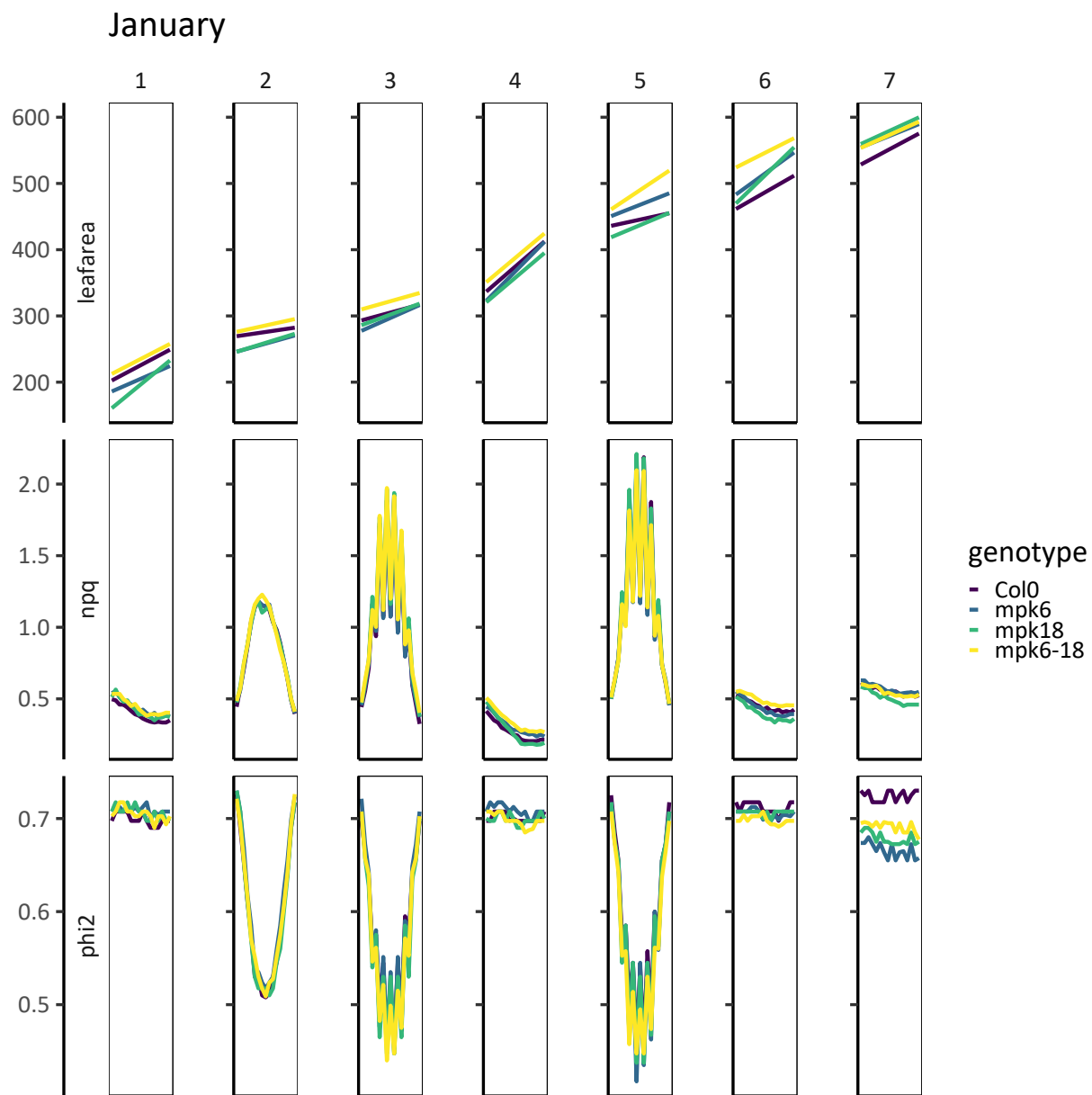


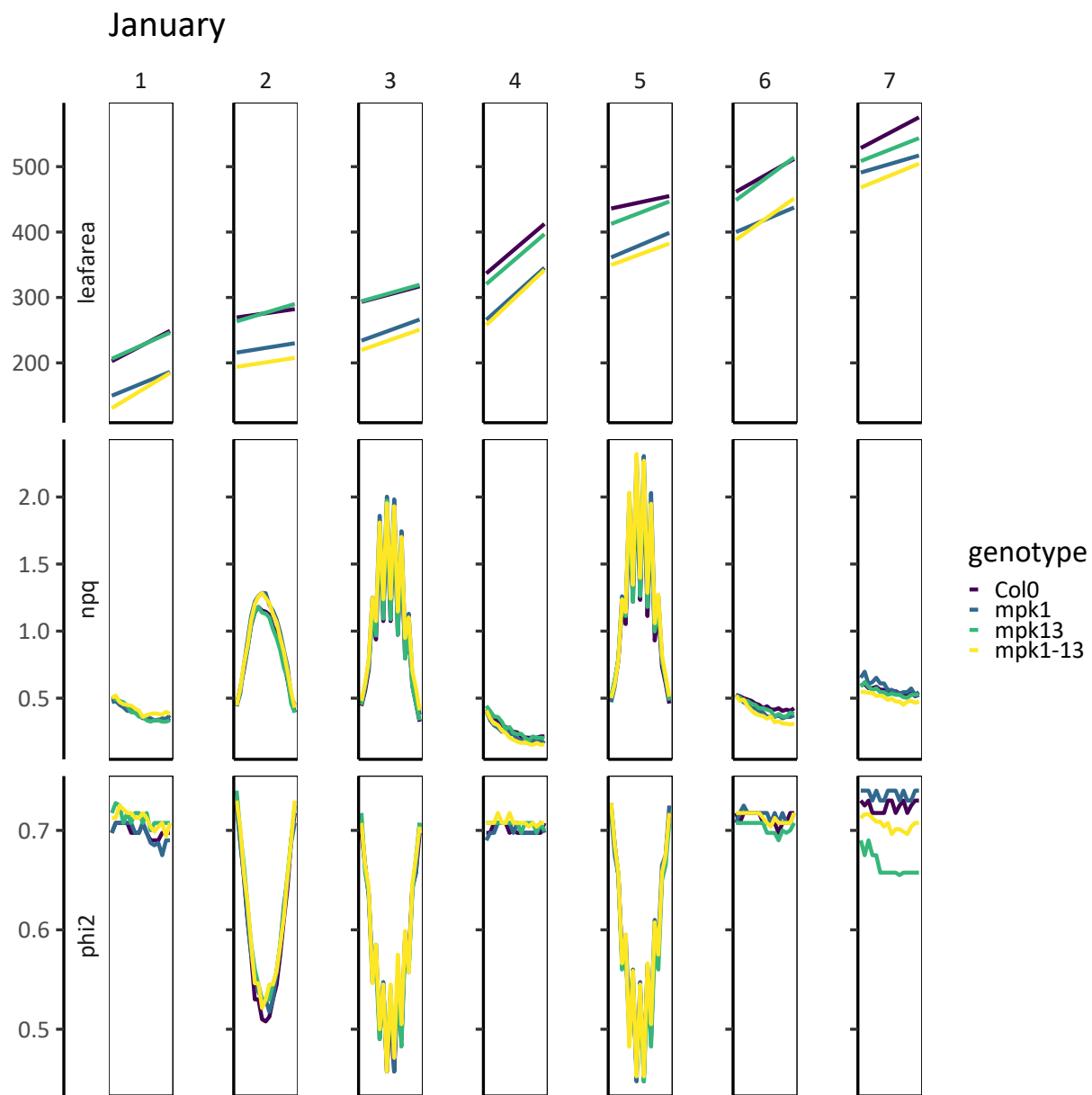


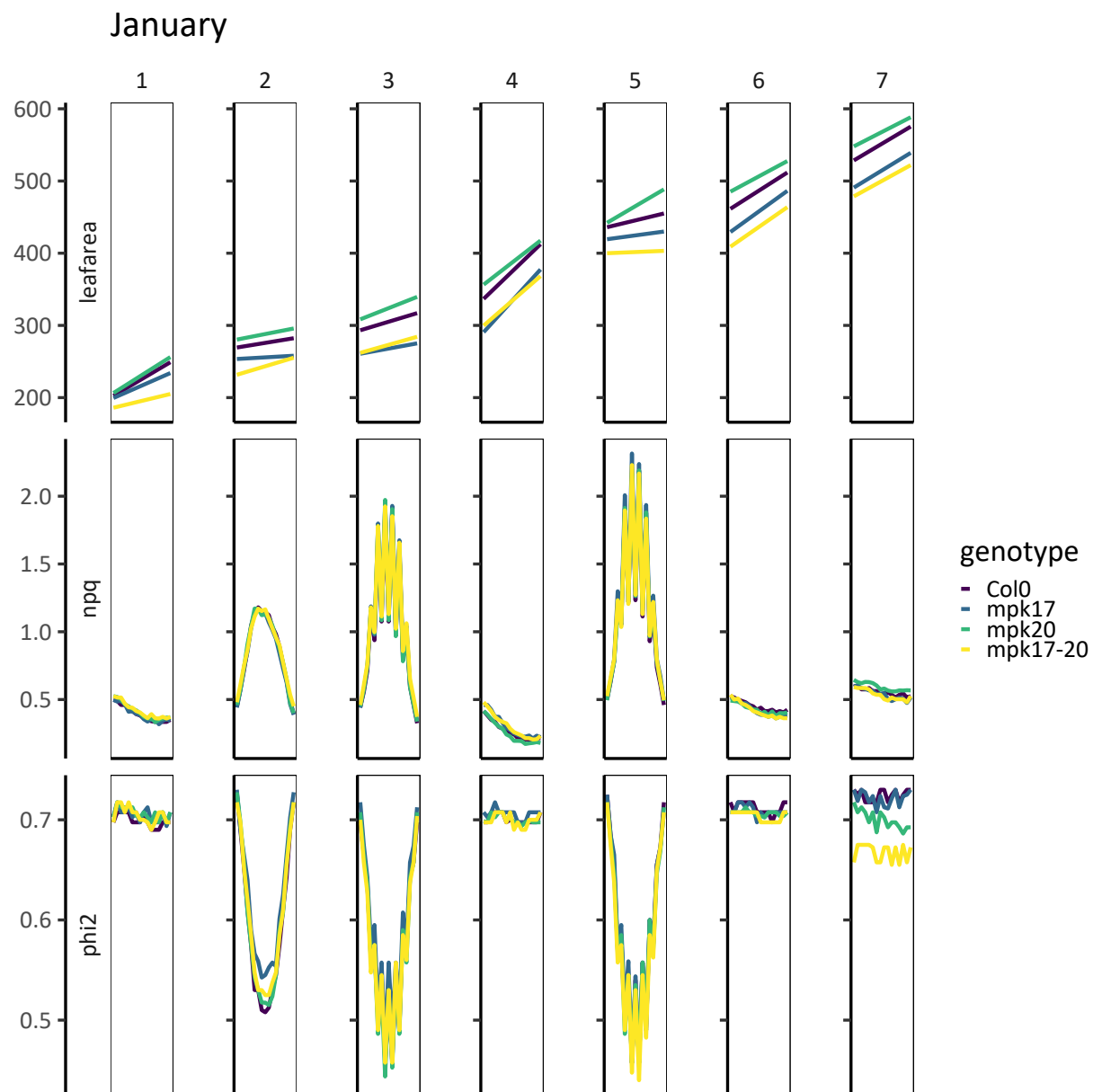


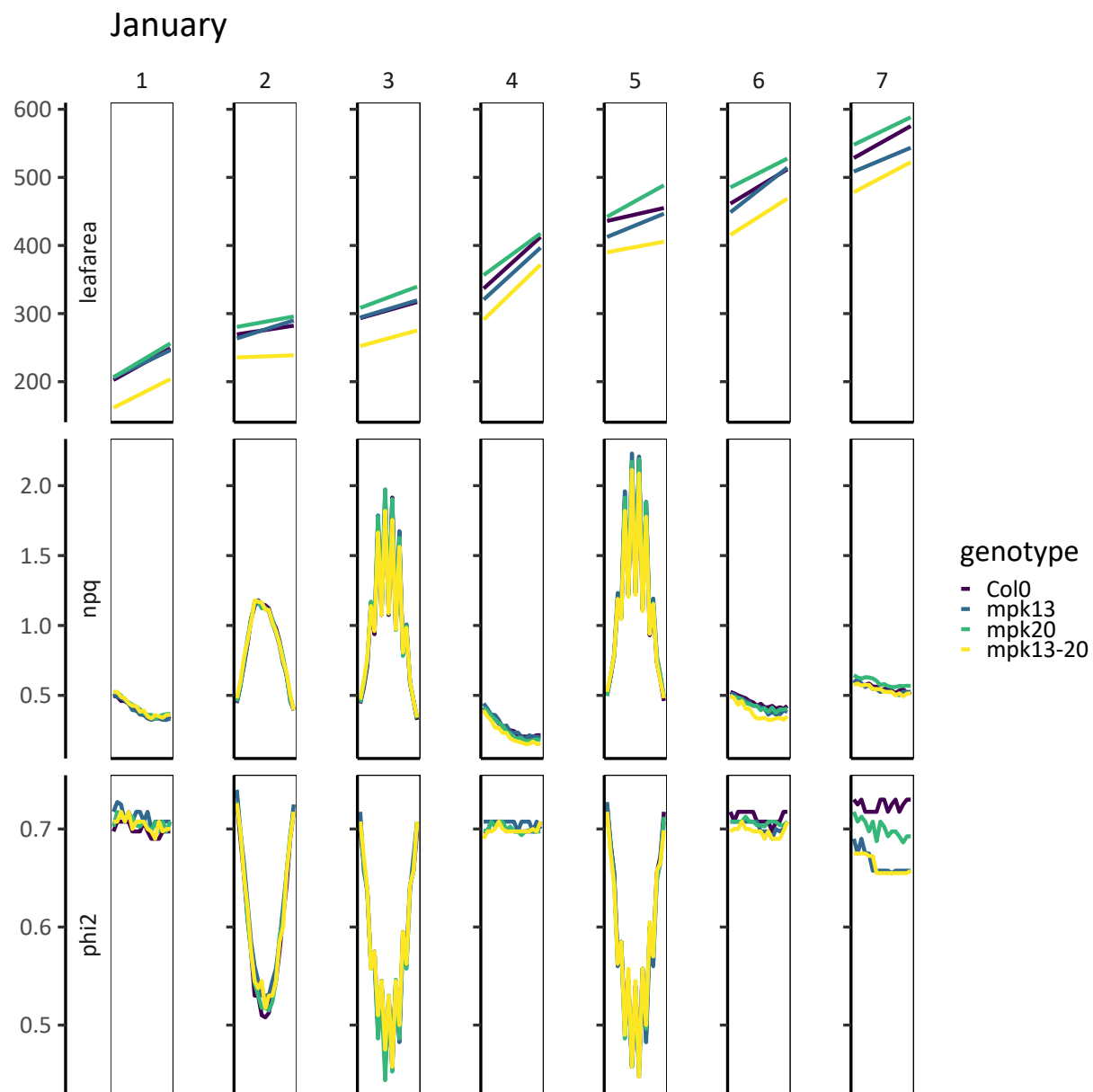


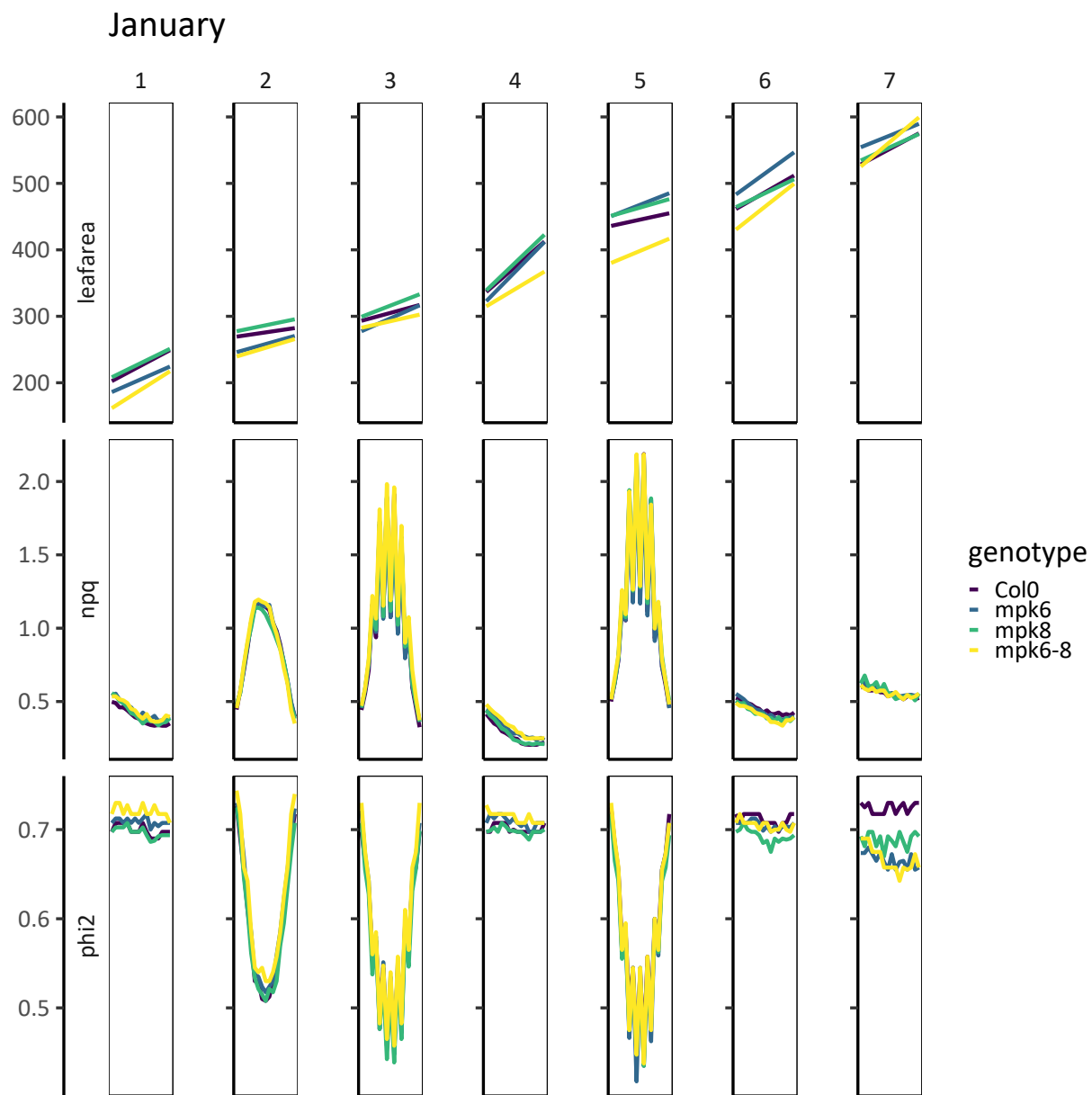


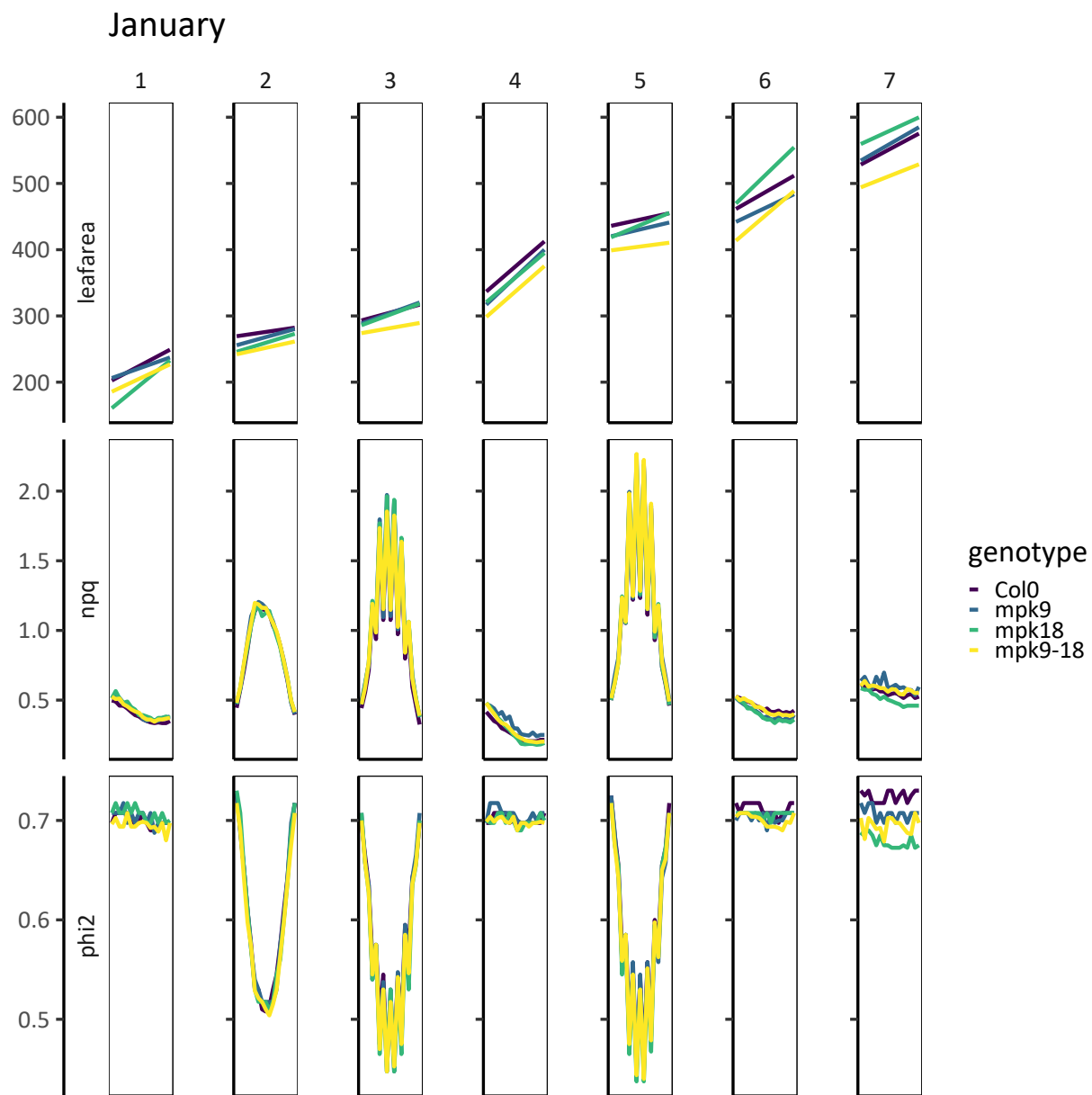


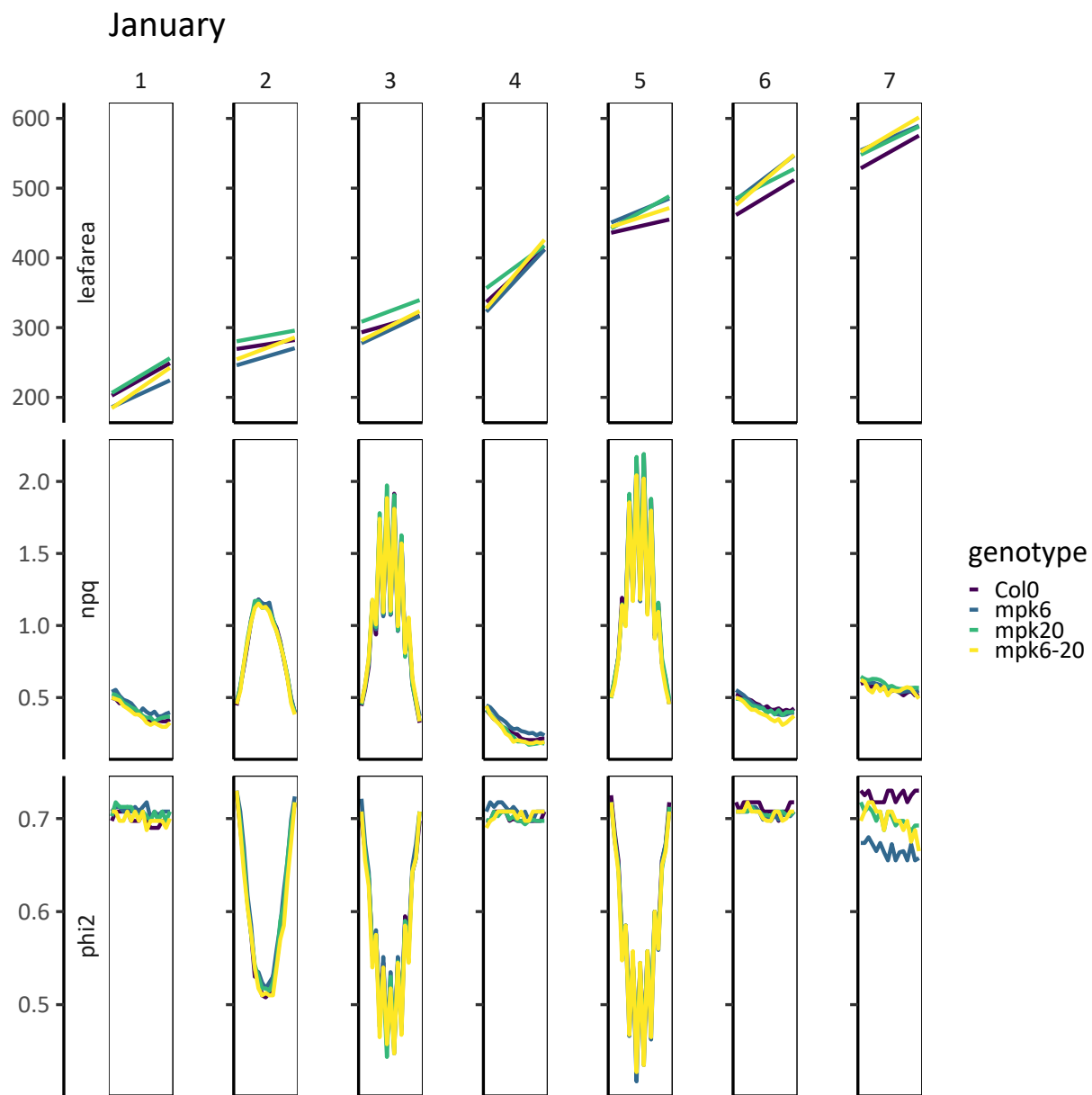




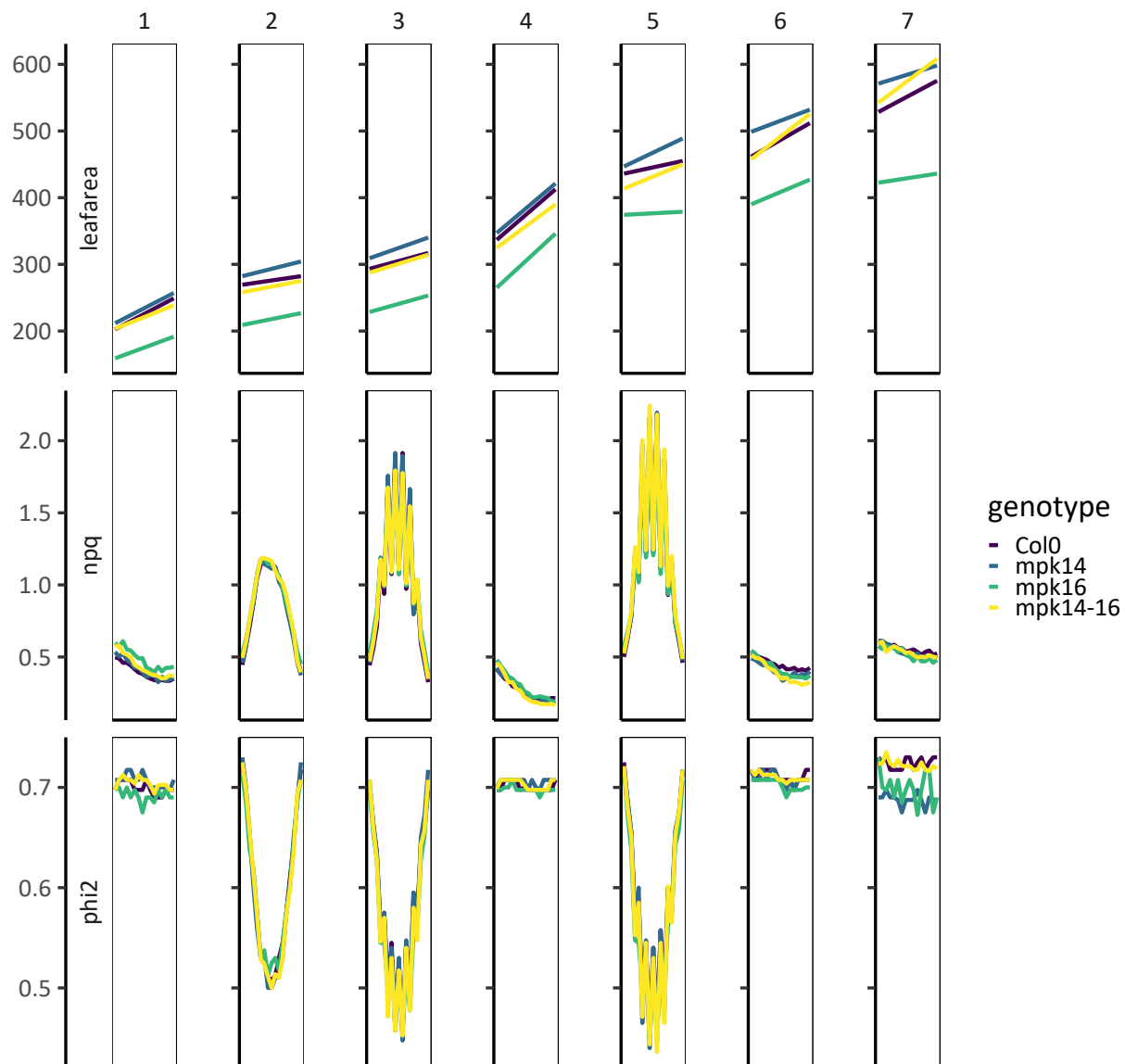


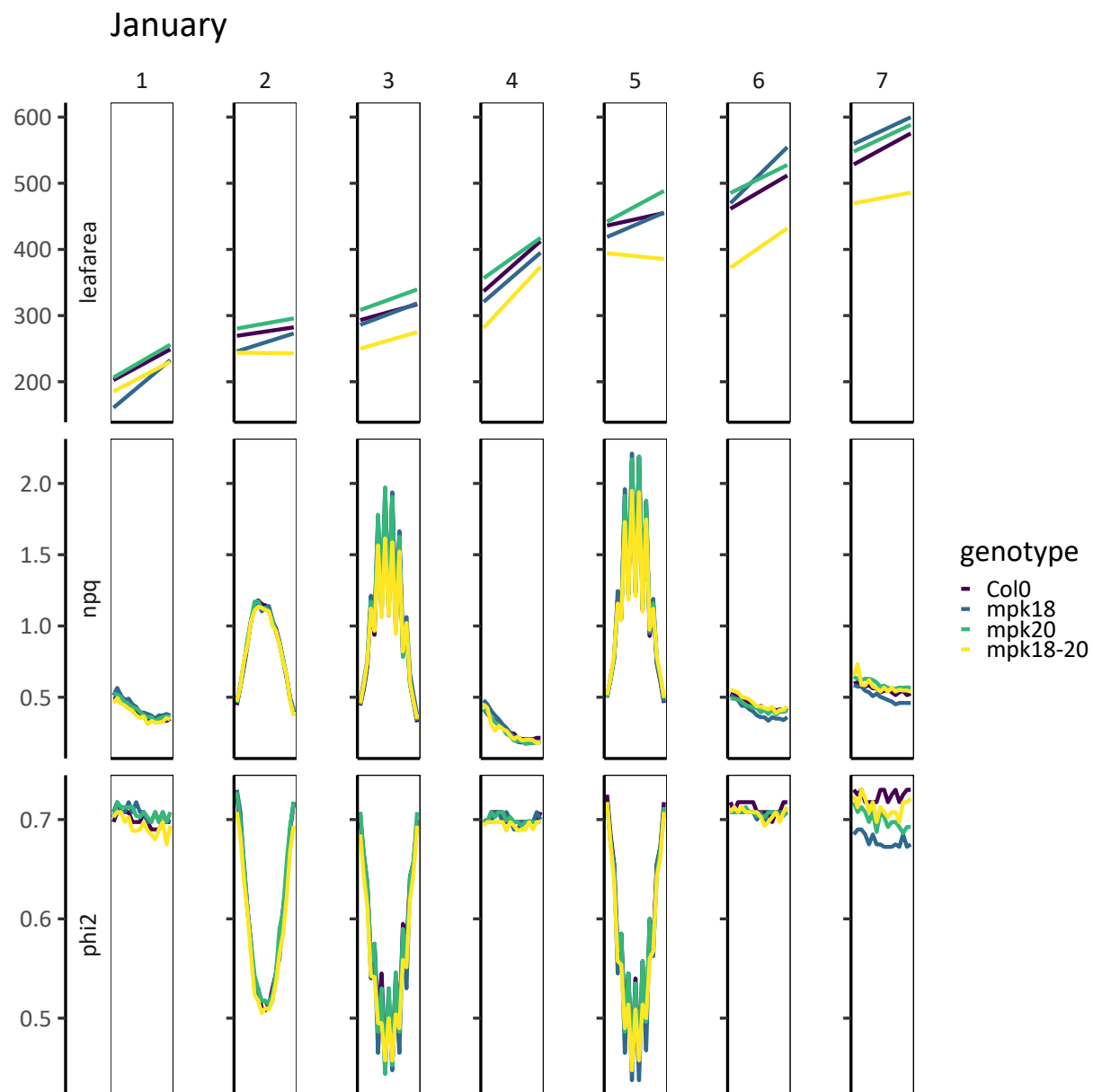


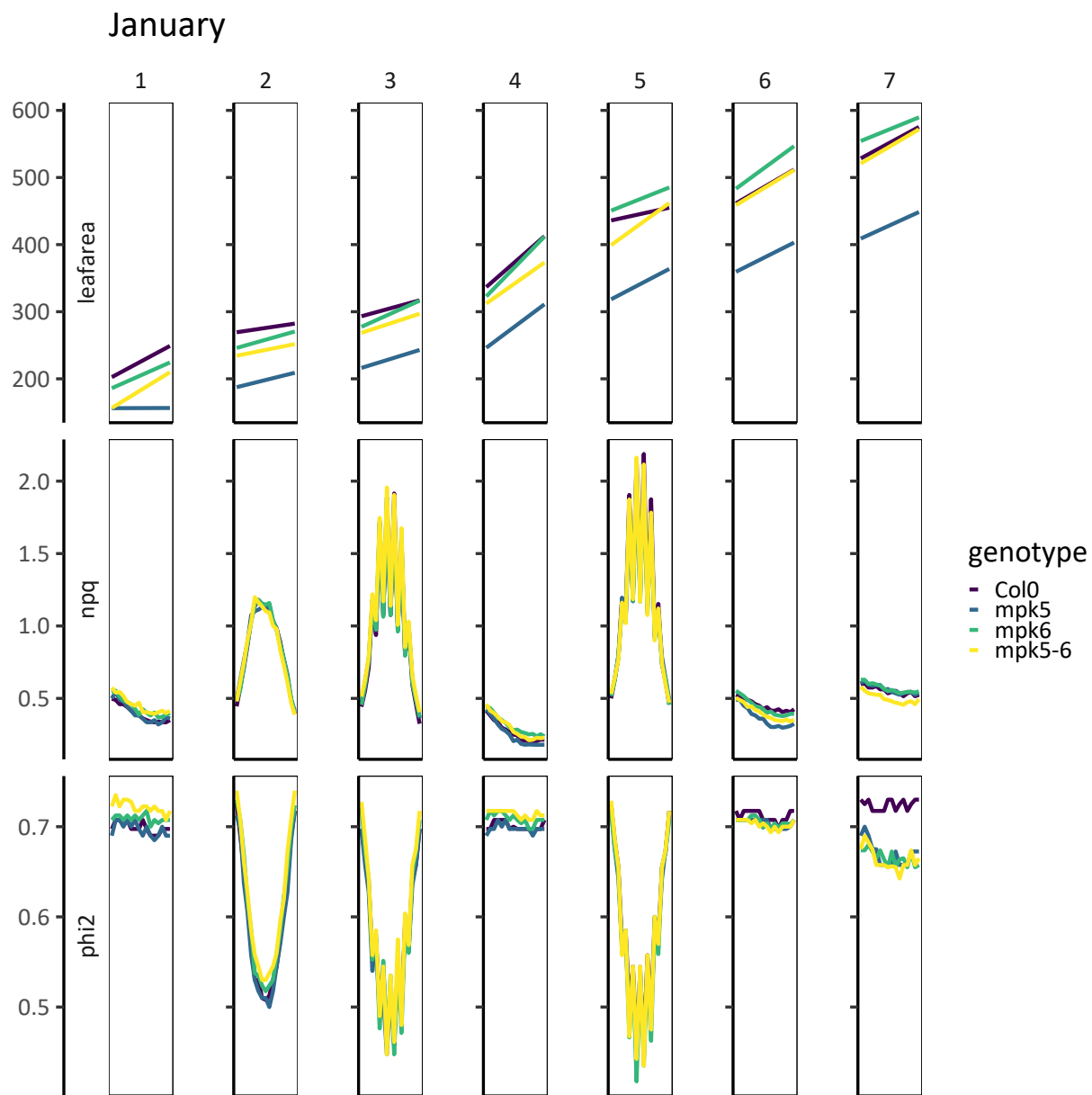


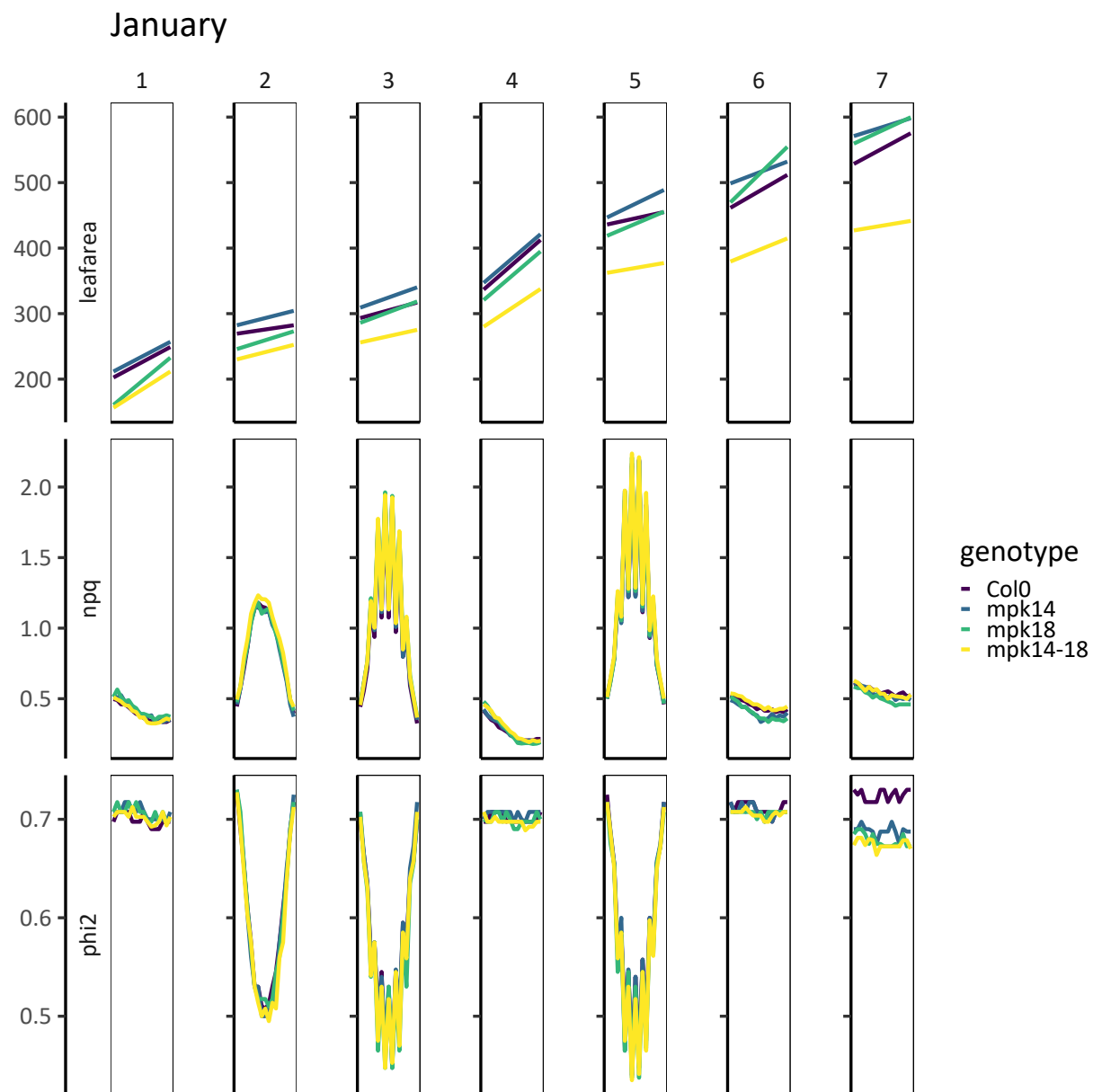


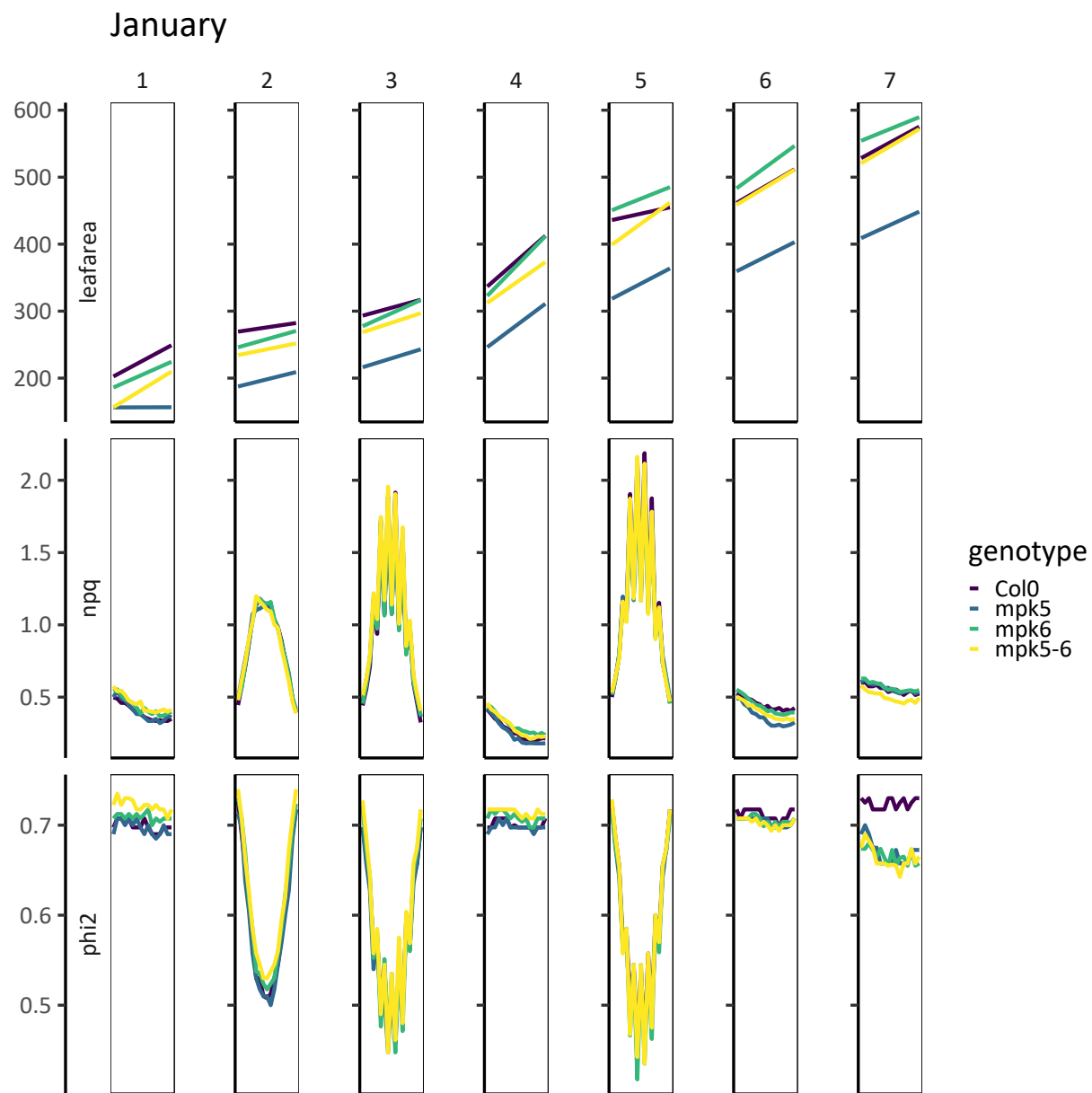
January

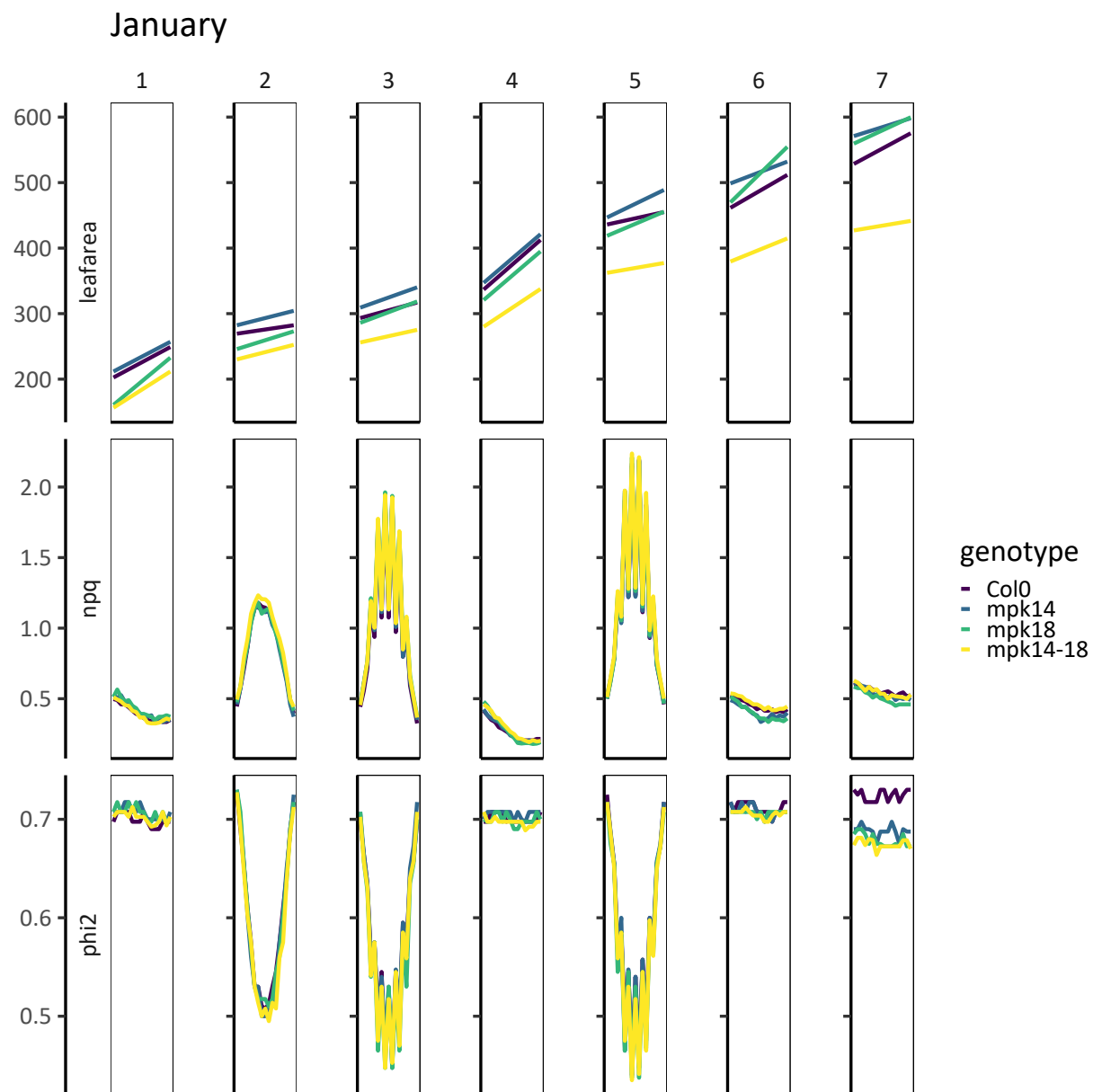




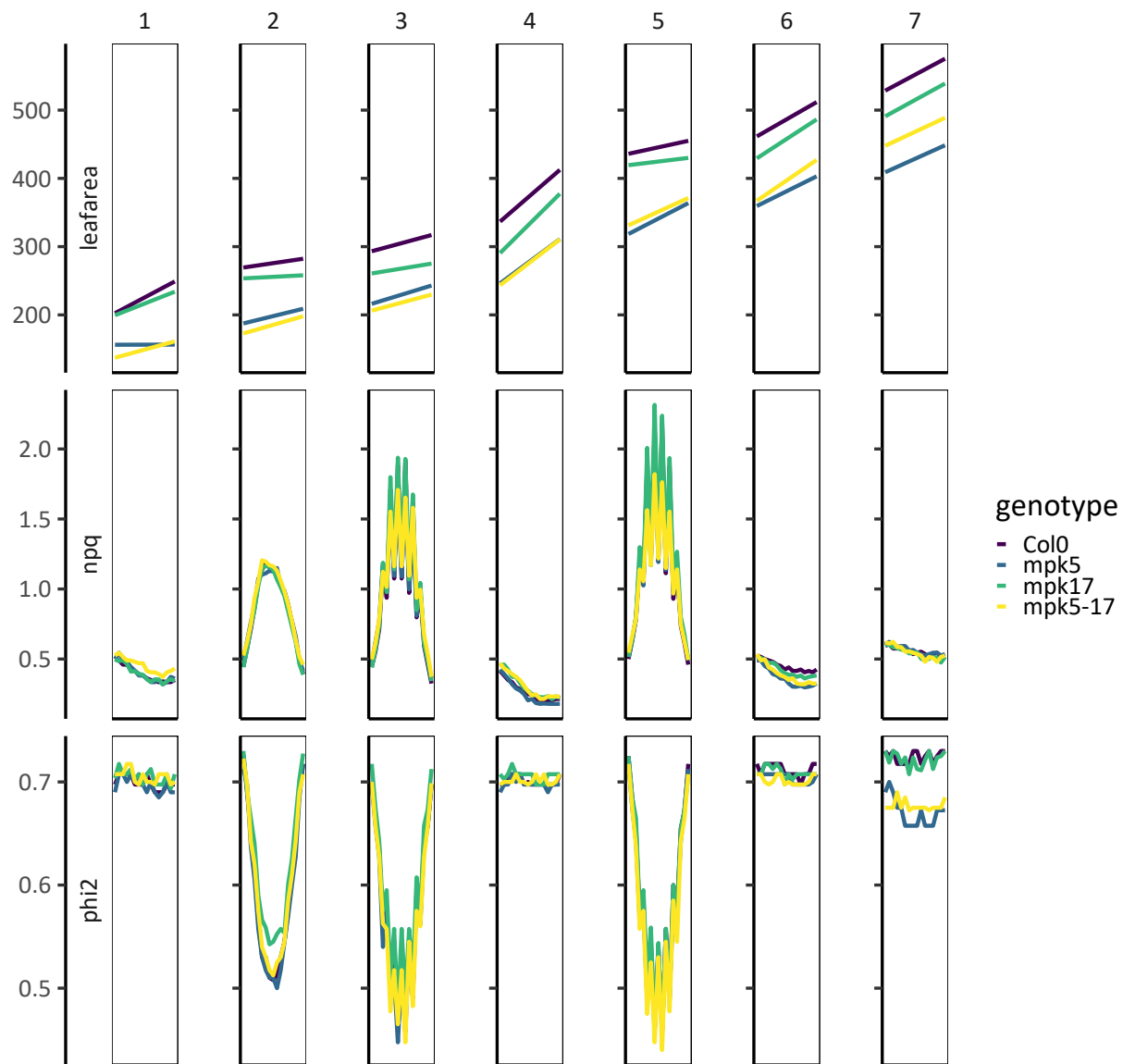


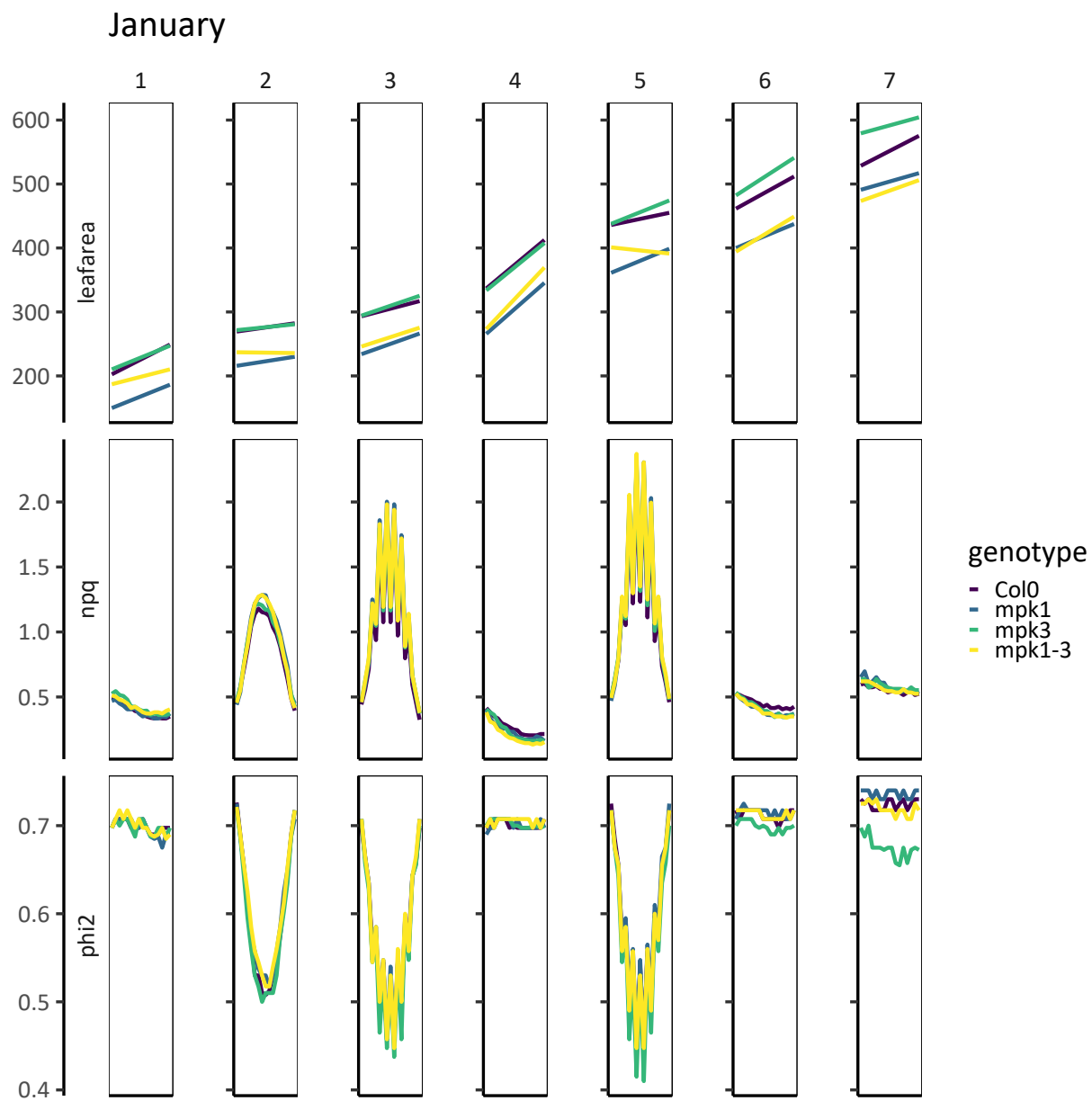


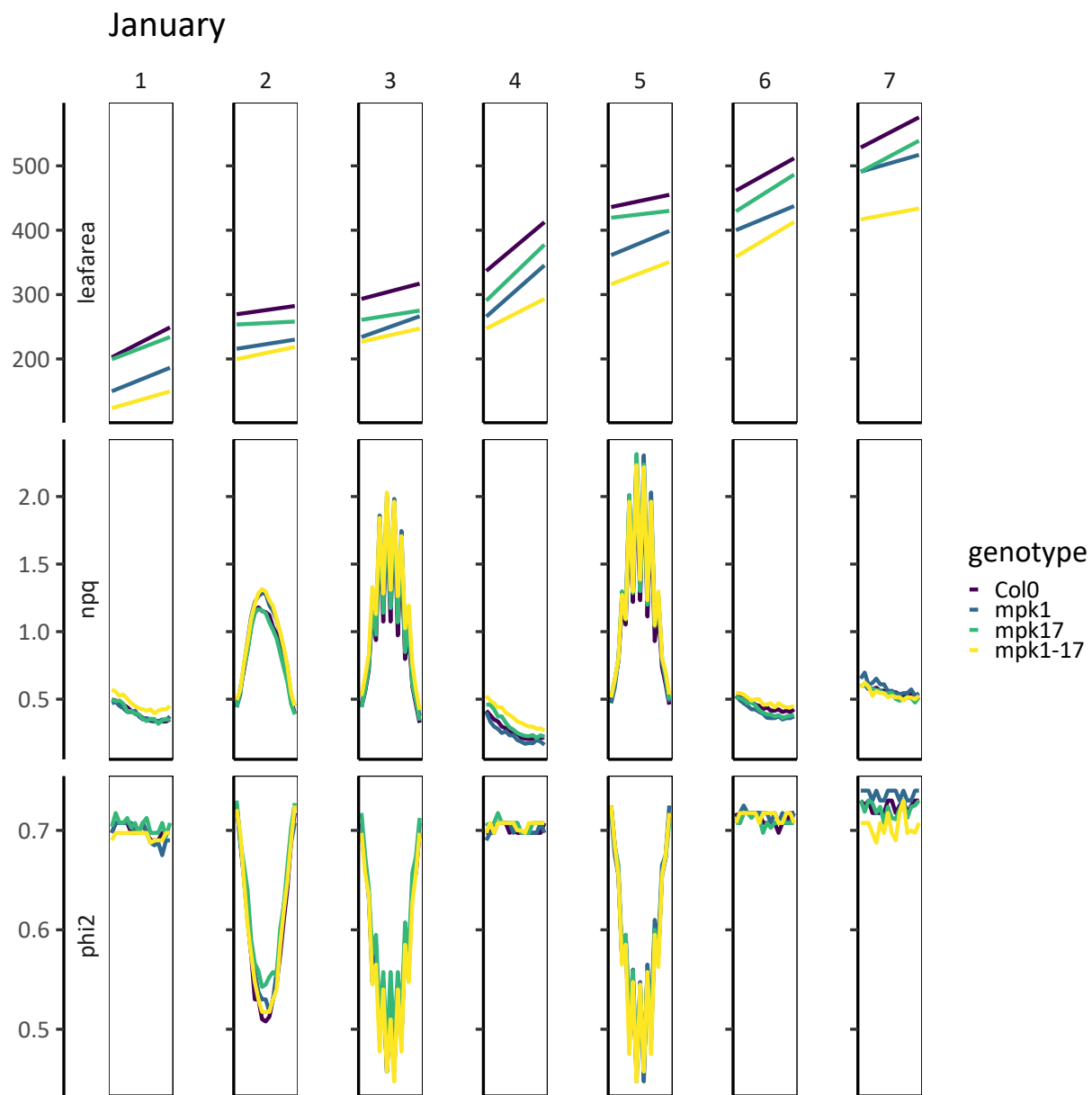




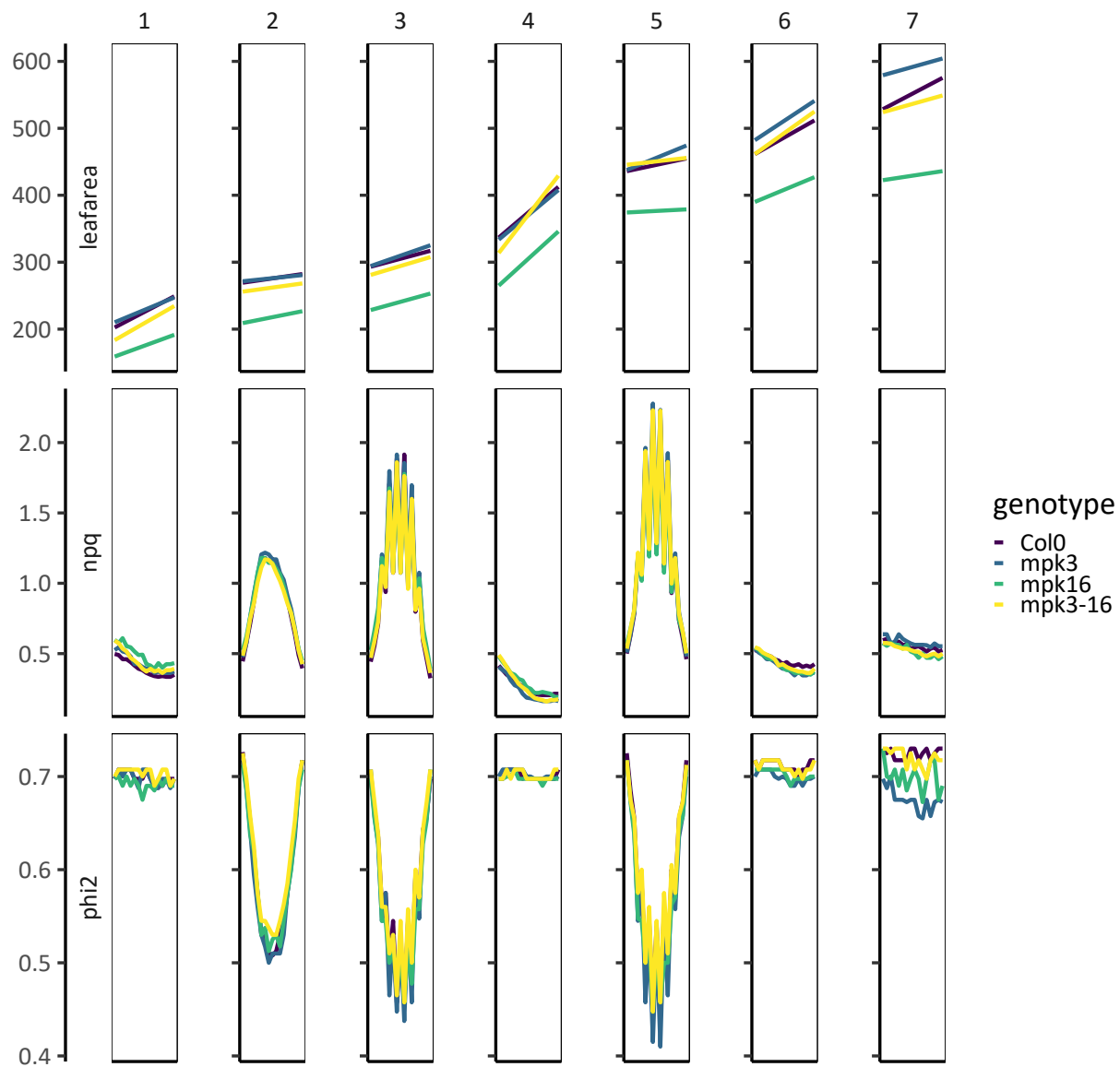
January

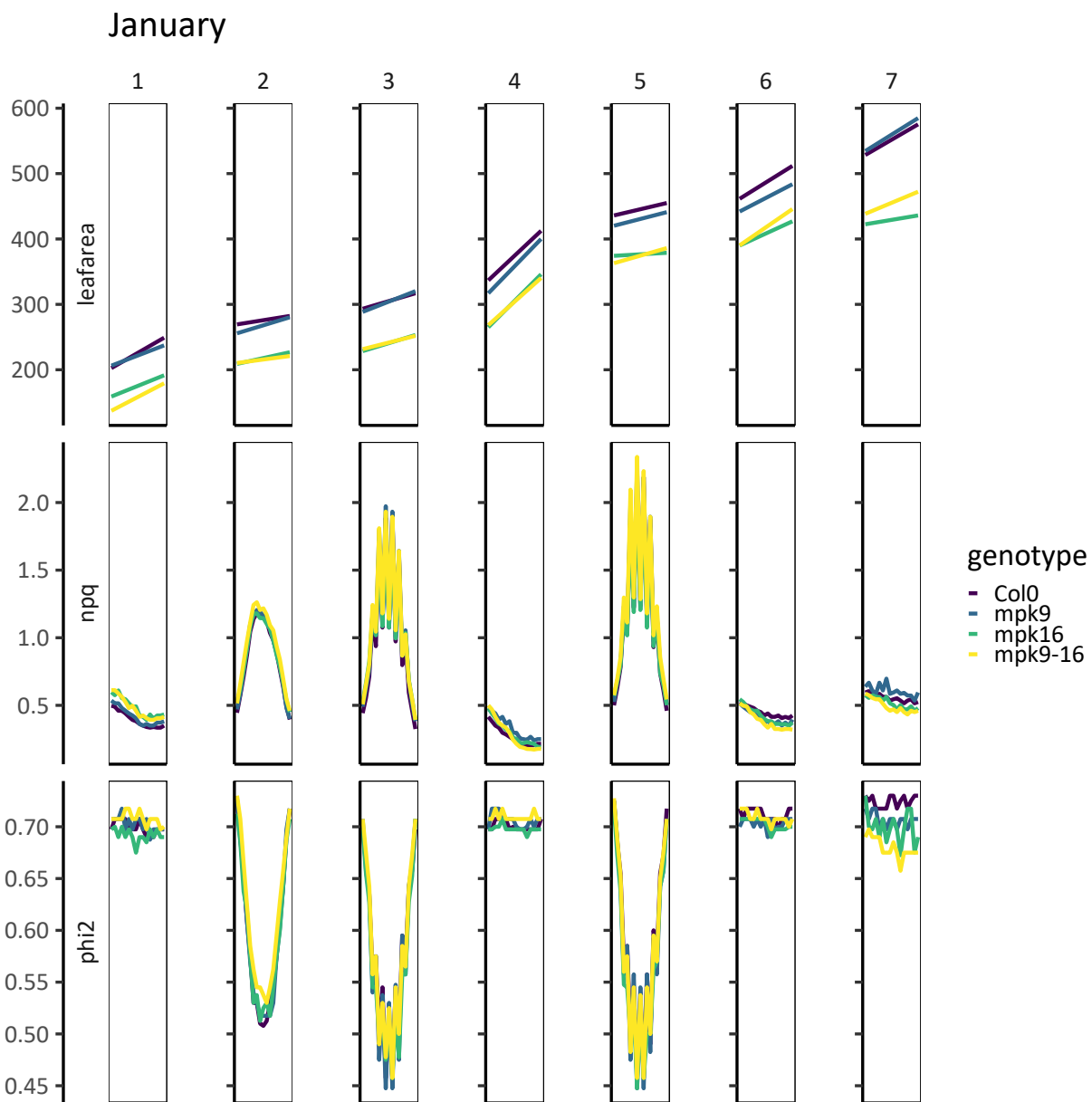




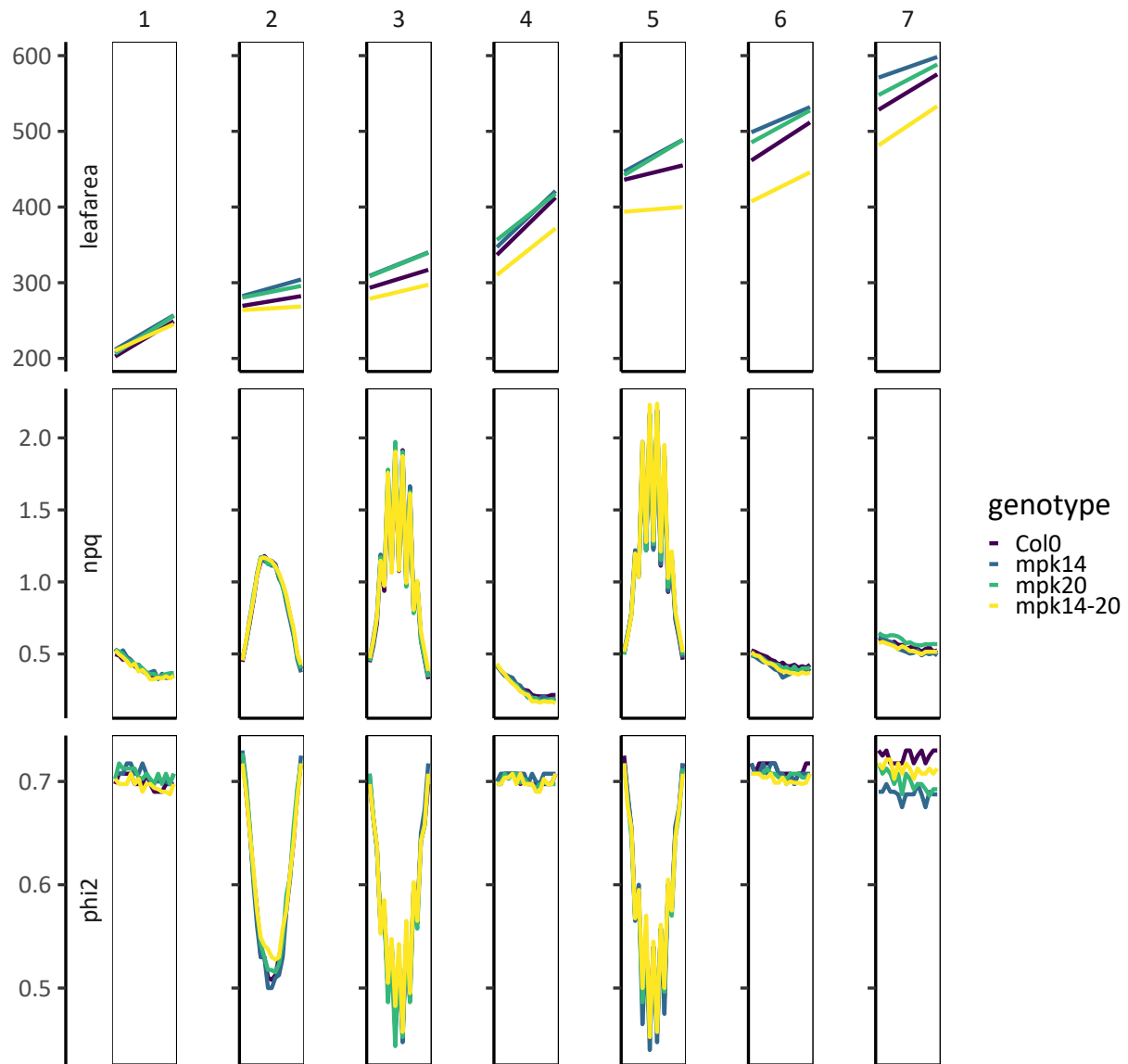


January





January



Februray

```
plot_data <- filter(depi_data, depi_data$month ==
  "Feb")

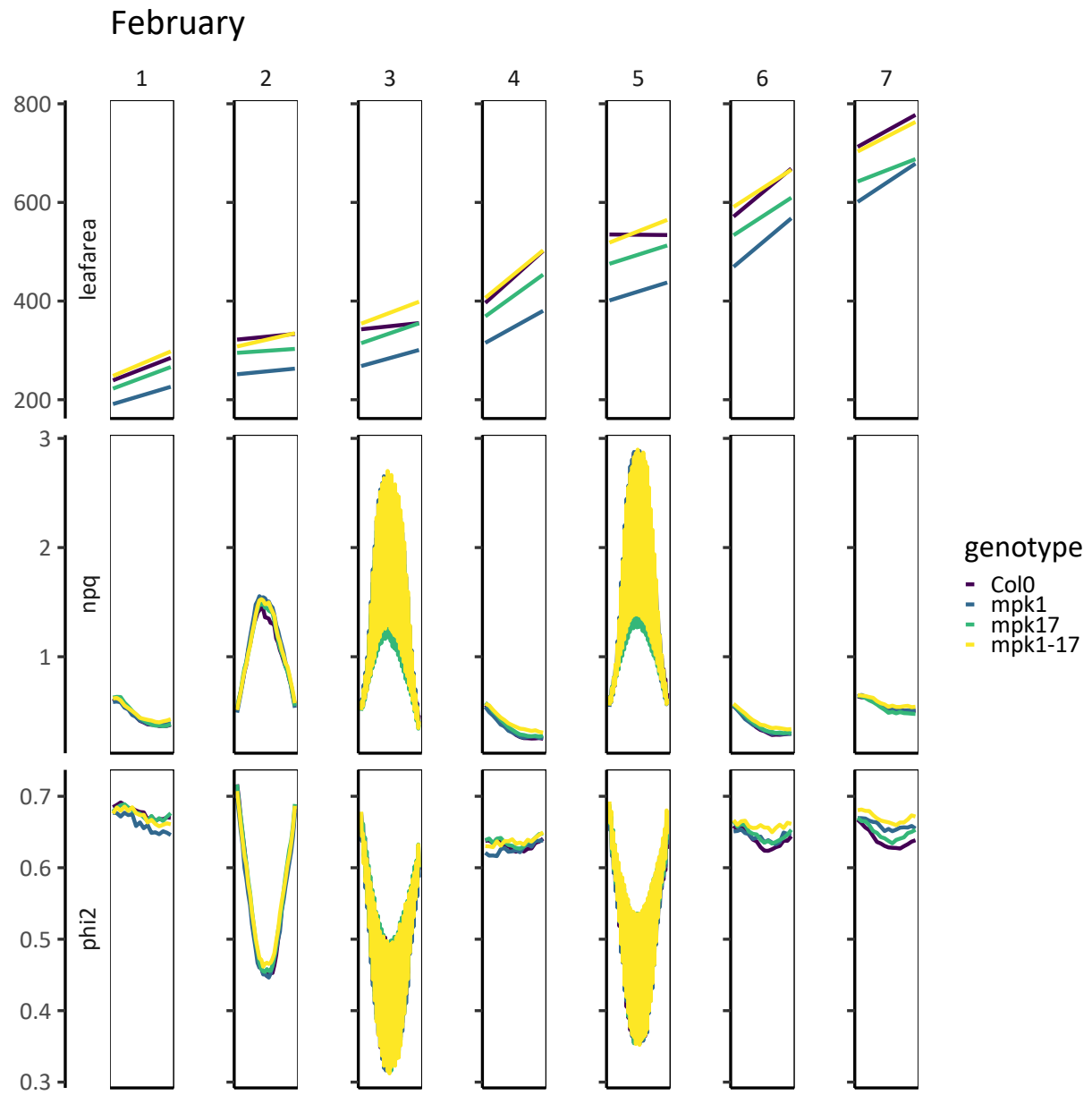
plot_data <- cell_370_data(plot_data)
plot_data <- add_number(plot_data)
plot_data <- add_day_col(plot_data)
plot_data$genotype <- reorder(plot_data$genotype,
  plot_data$number)

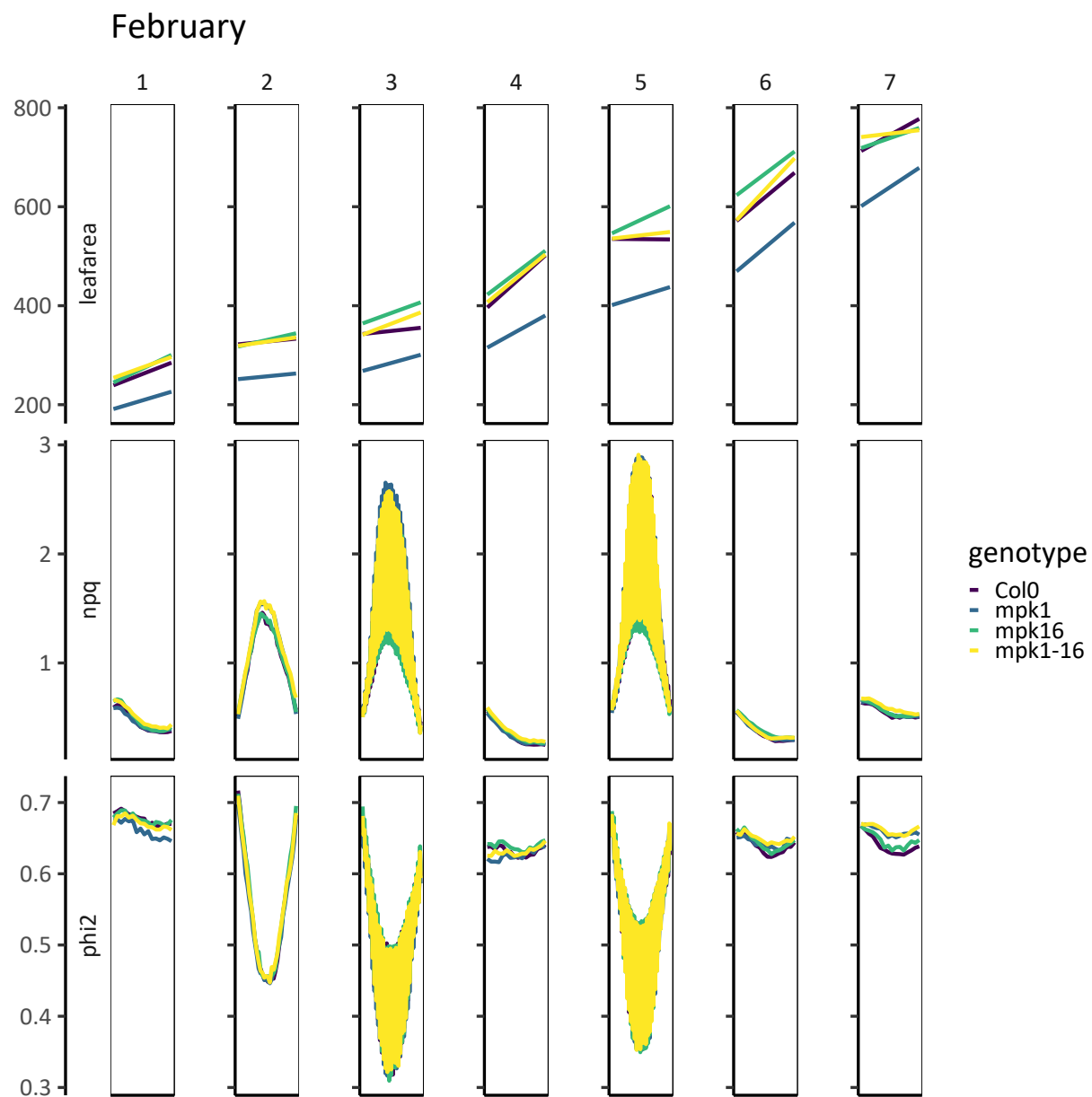
for (element in genotype_combinations) {
```

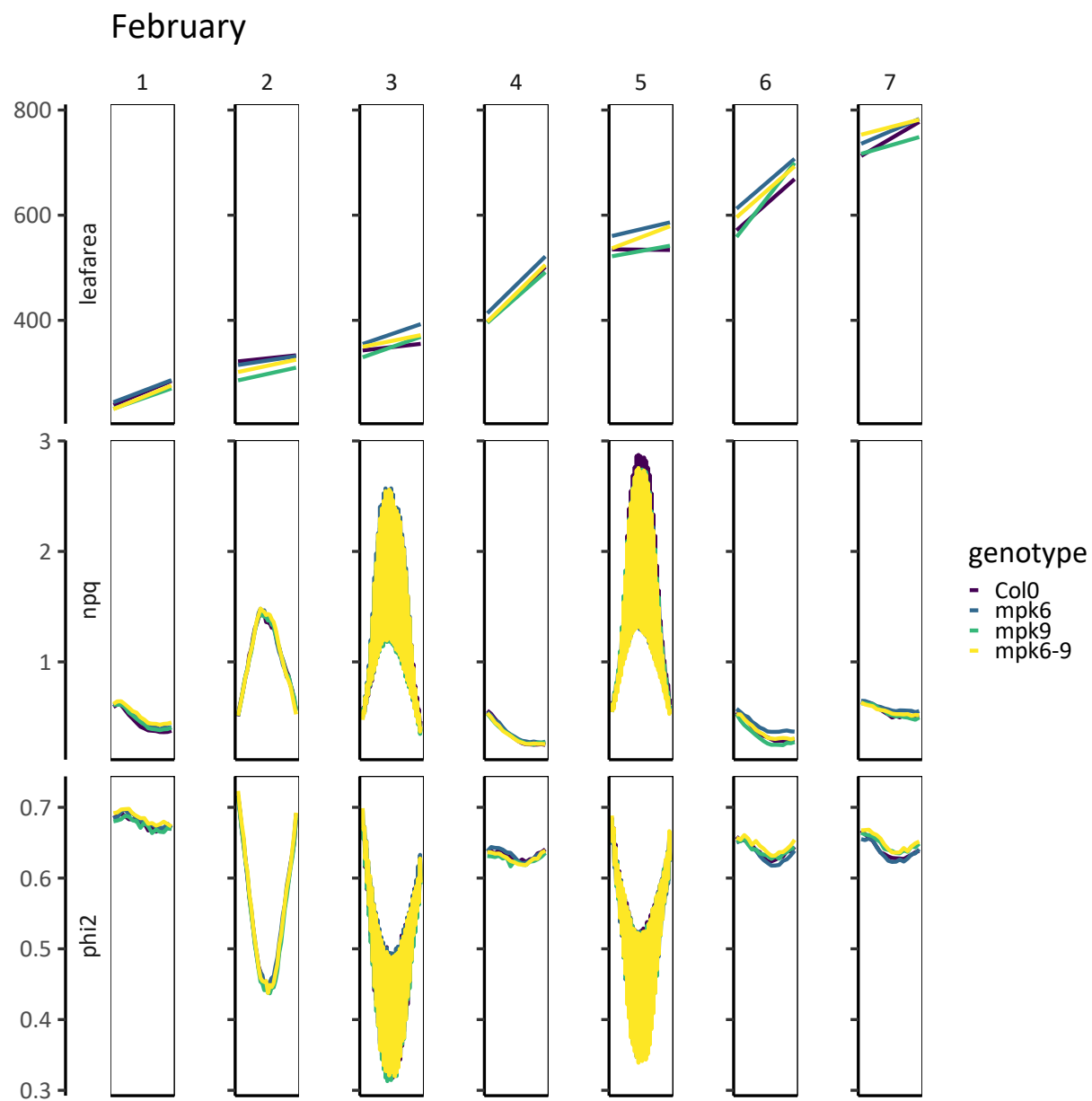
```

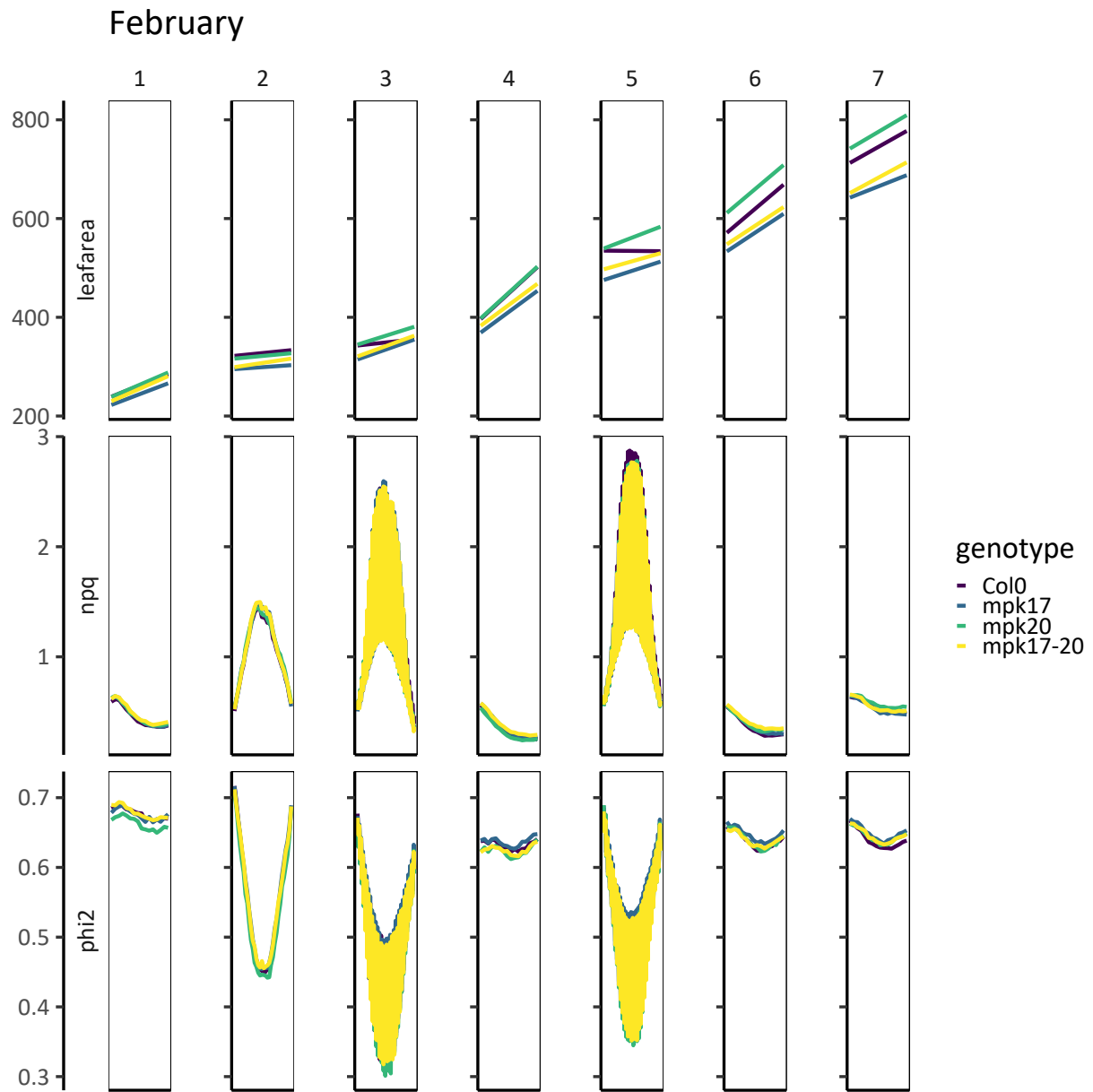
data <- filter(plot_data, genotype %in%
  c(element, "Col0"))
plot <- ggplot(data = data, aes(x = time_point,
  y = med)) + geom_line(aes(color = genotype),
  size = 3) + facet_rep_grid(measurement ~
  day, scales = "free", switch = "y",
  repeat.tick.labels = FALSE) + labs(x = "Hours",
  y = NULL, title = "February") + theme_tufte(base_family = "Calibri",
  base_size = 50) + theme(strip.background.x = element_blank(),
  axis.title.x = element_blank(), axis.text.x = element_blank(),
  axis.ticks.x = element_blank(), panel.border = element_rect(color = "black",
  fill = NA, size = 1), axis.line = element_line(),
  panel.spacing = unit(1, "lines")) +
  scale_color_viridis_d(begin = 0,
  end = 1, option = "viridis",
  aesthetics = c("colour", "fill"))
print(plot)
}

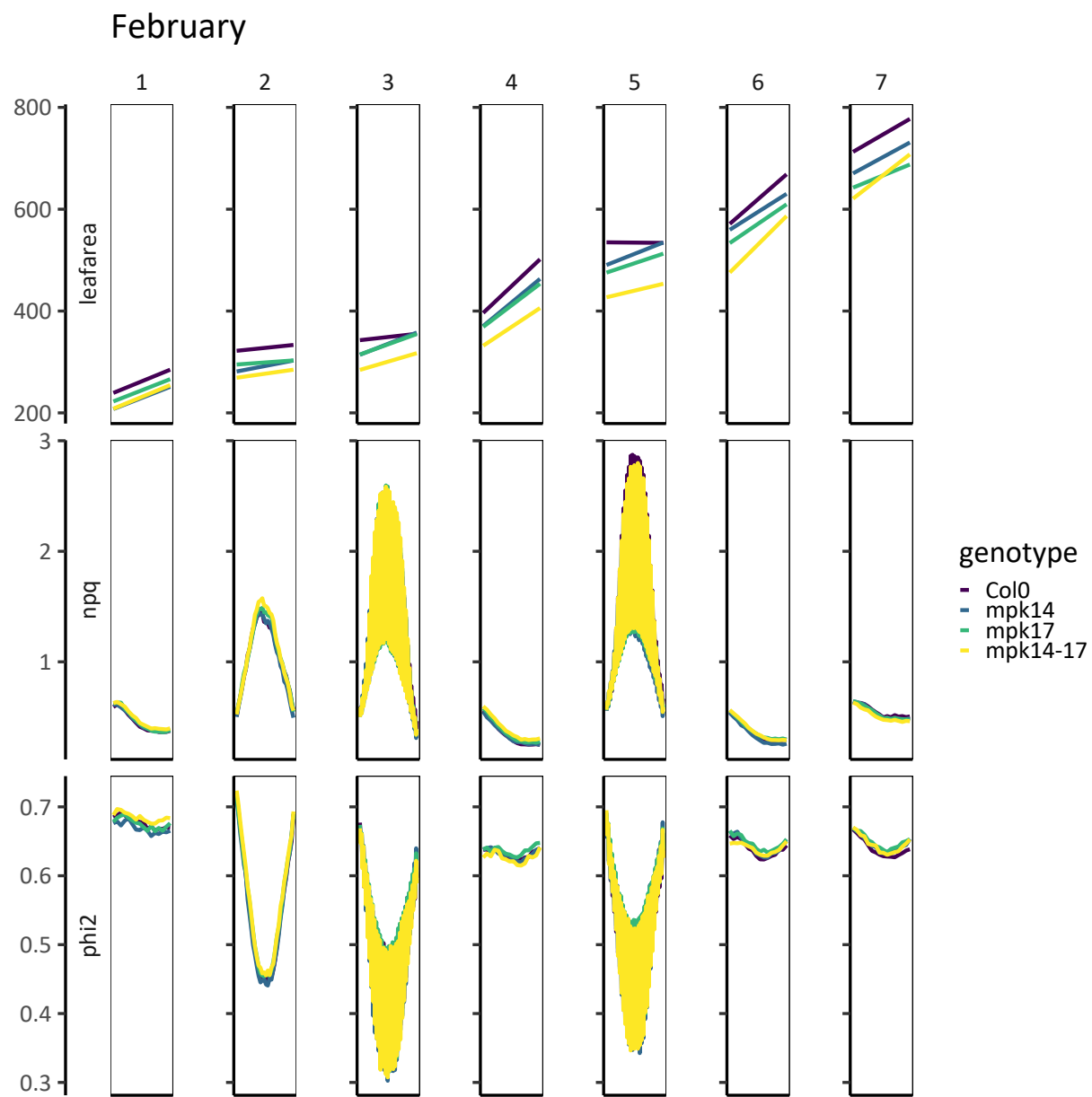
```



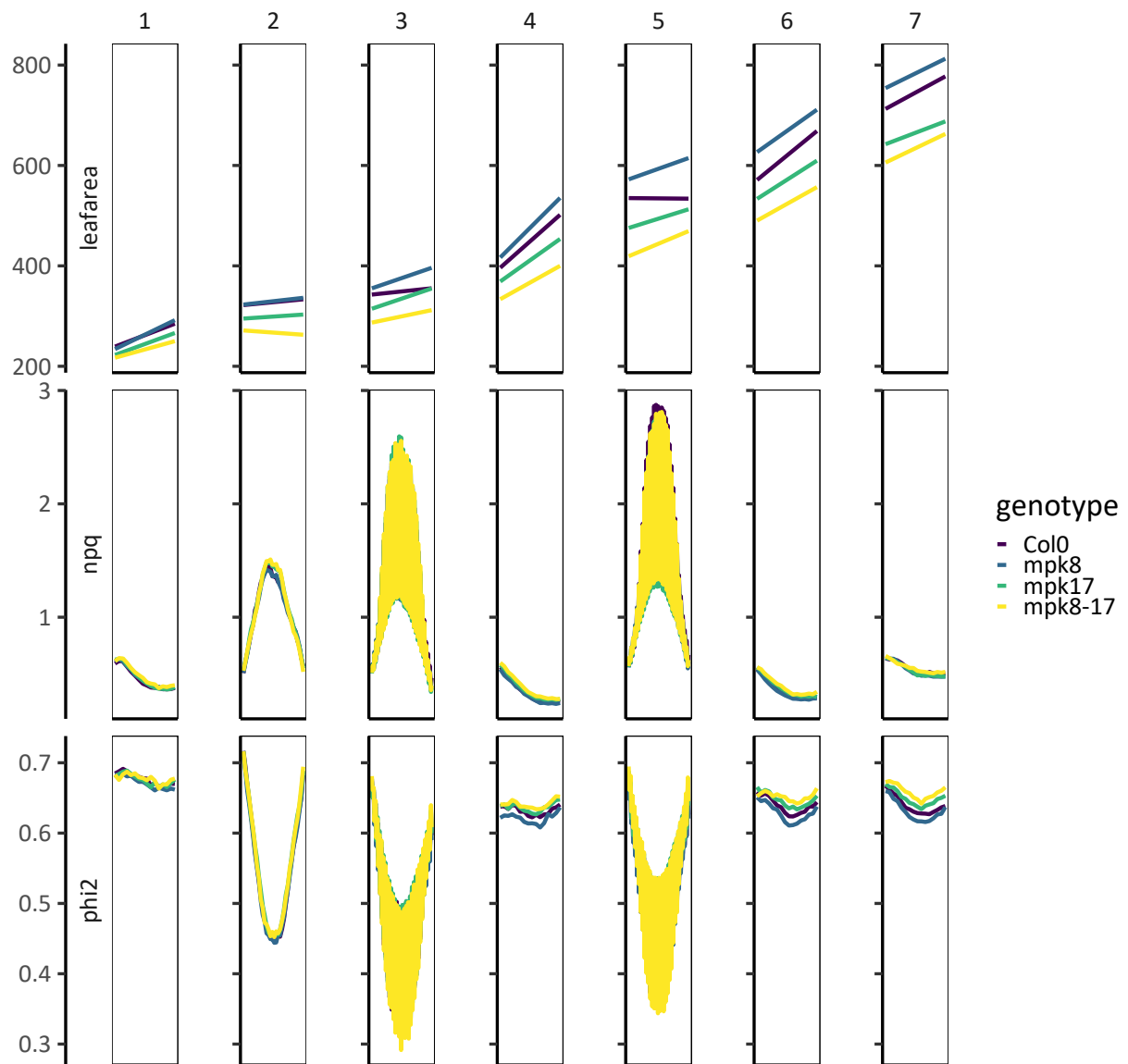




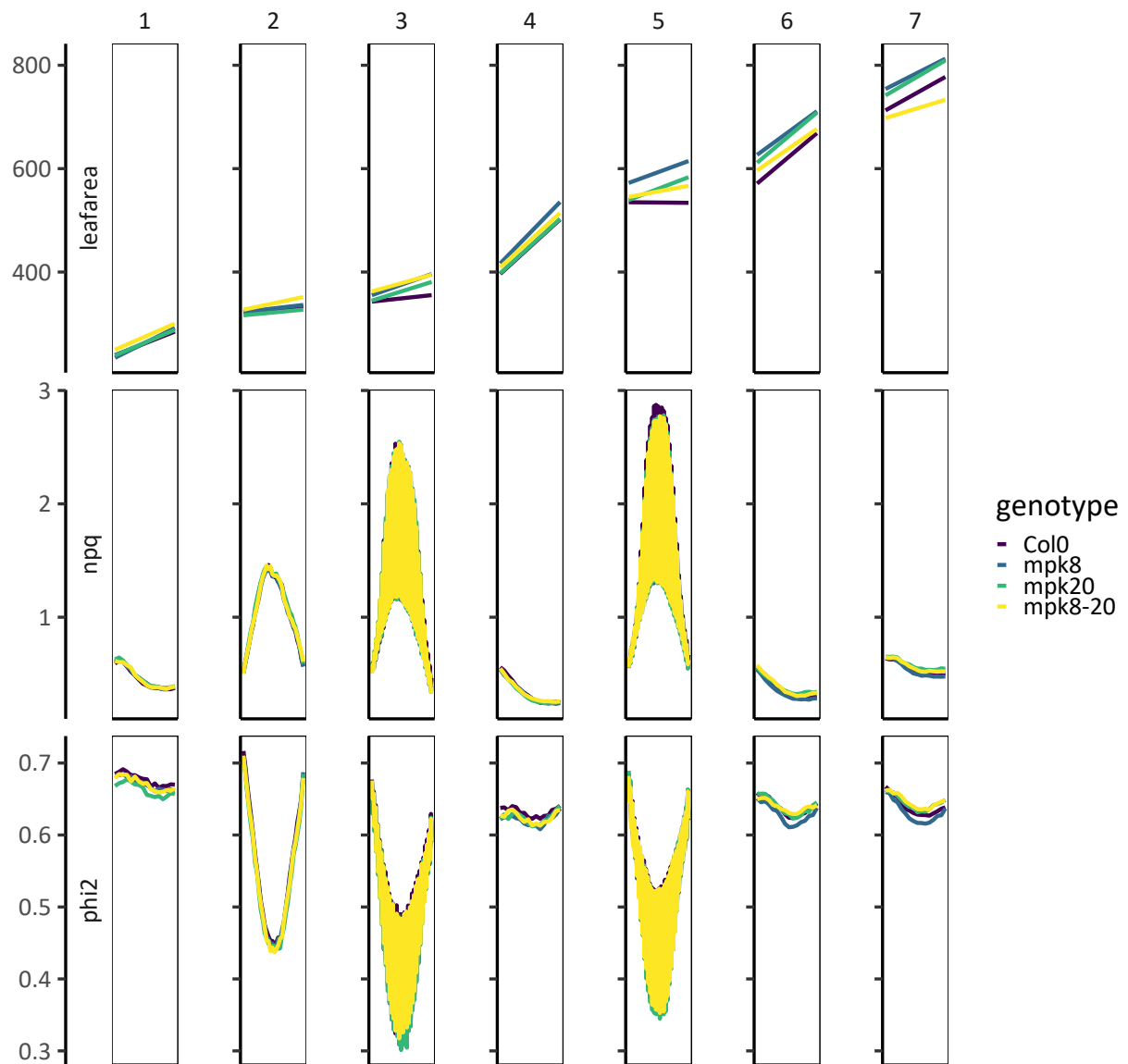




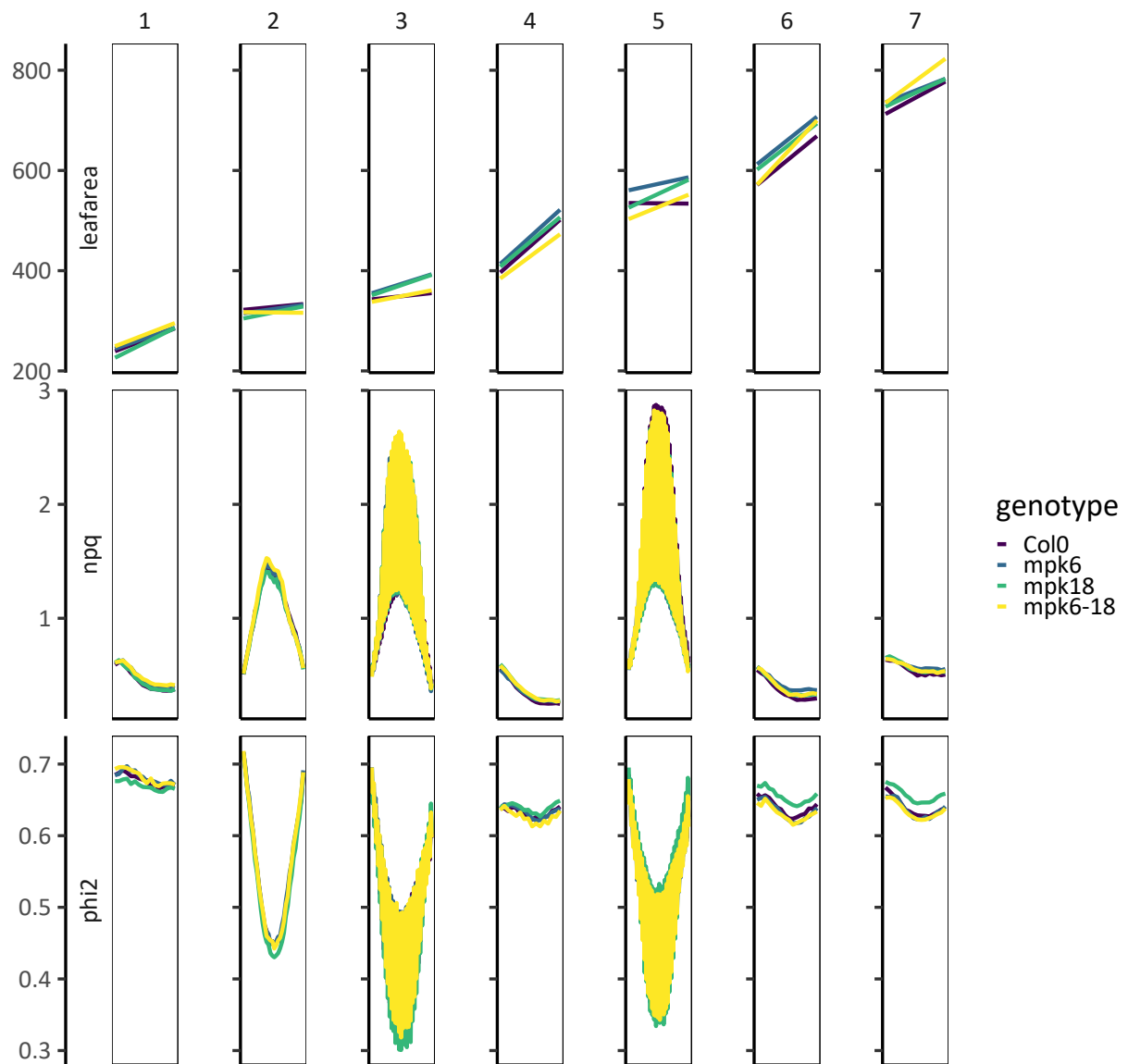
February

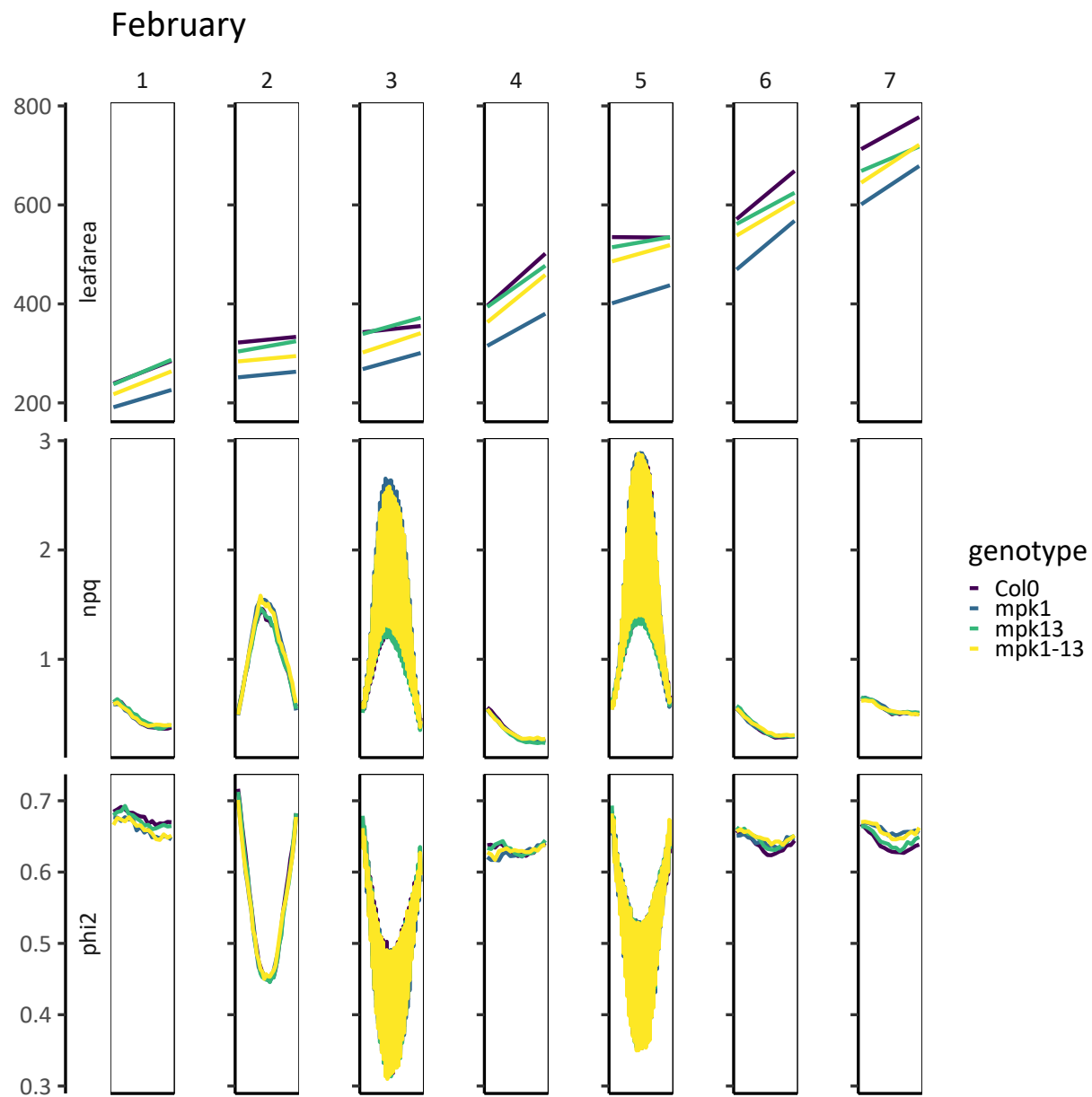


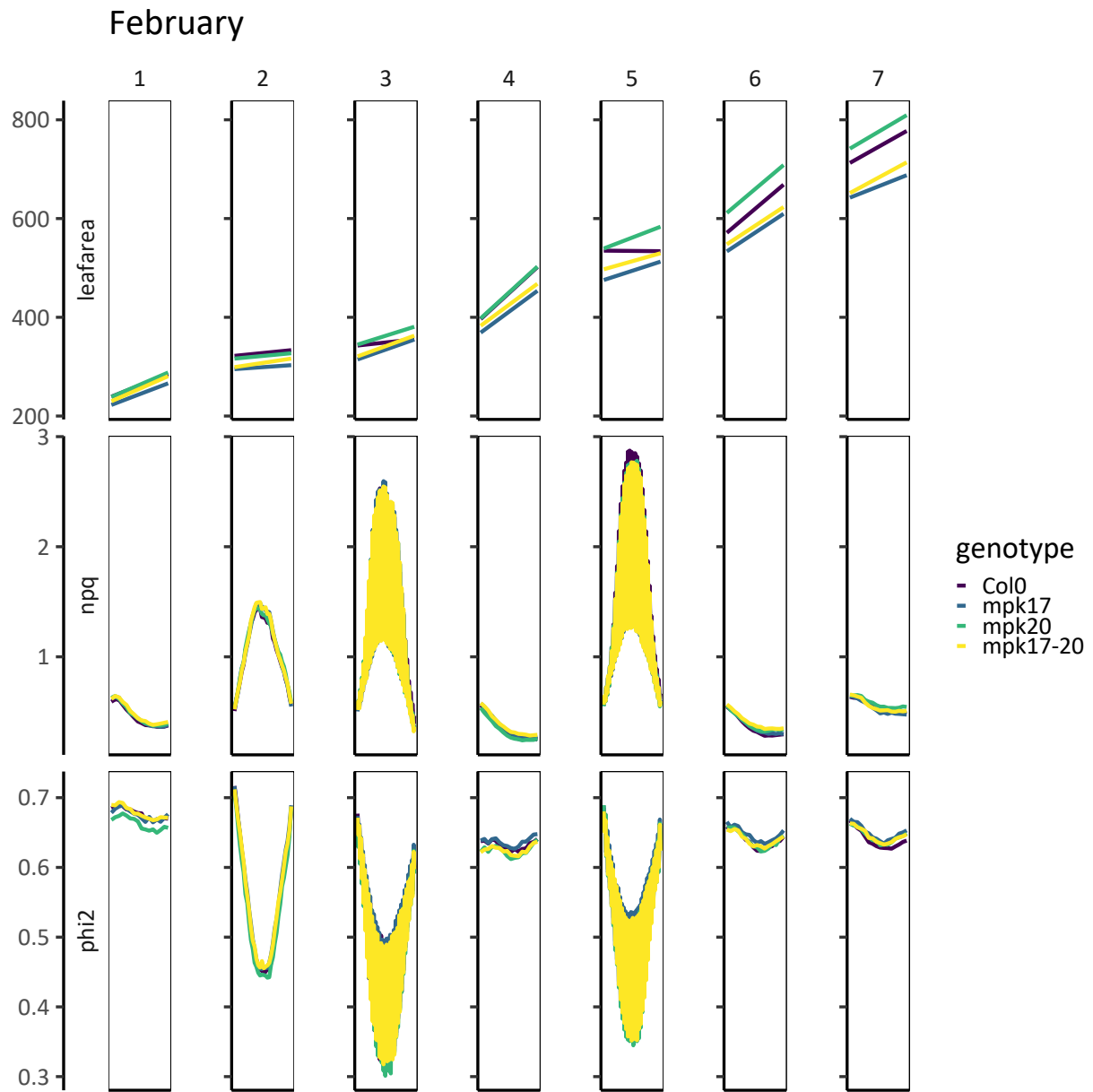
February

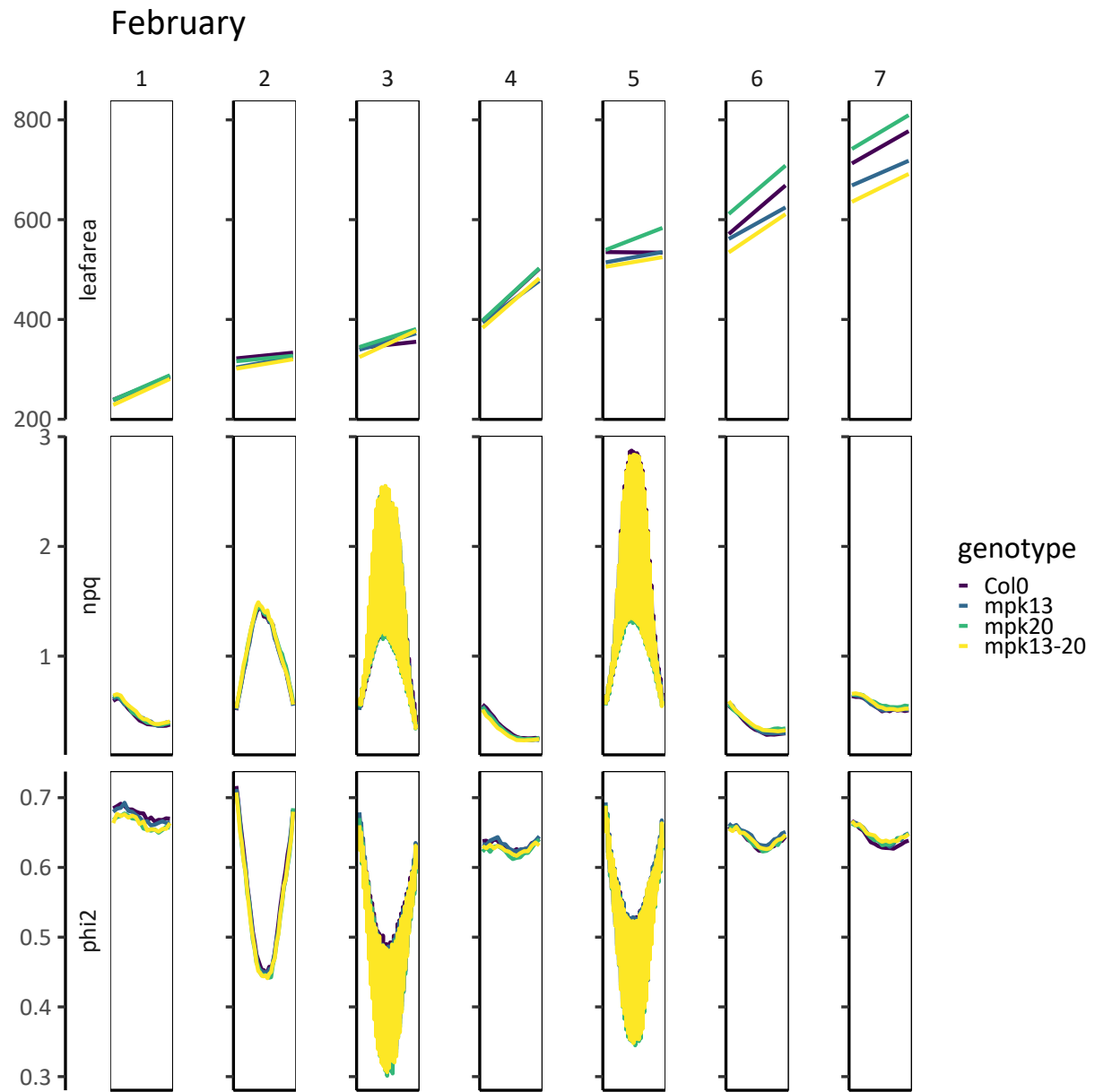


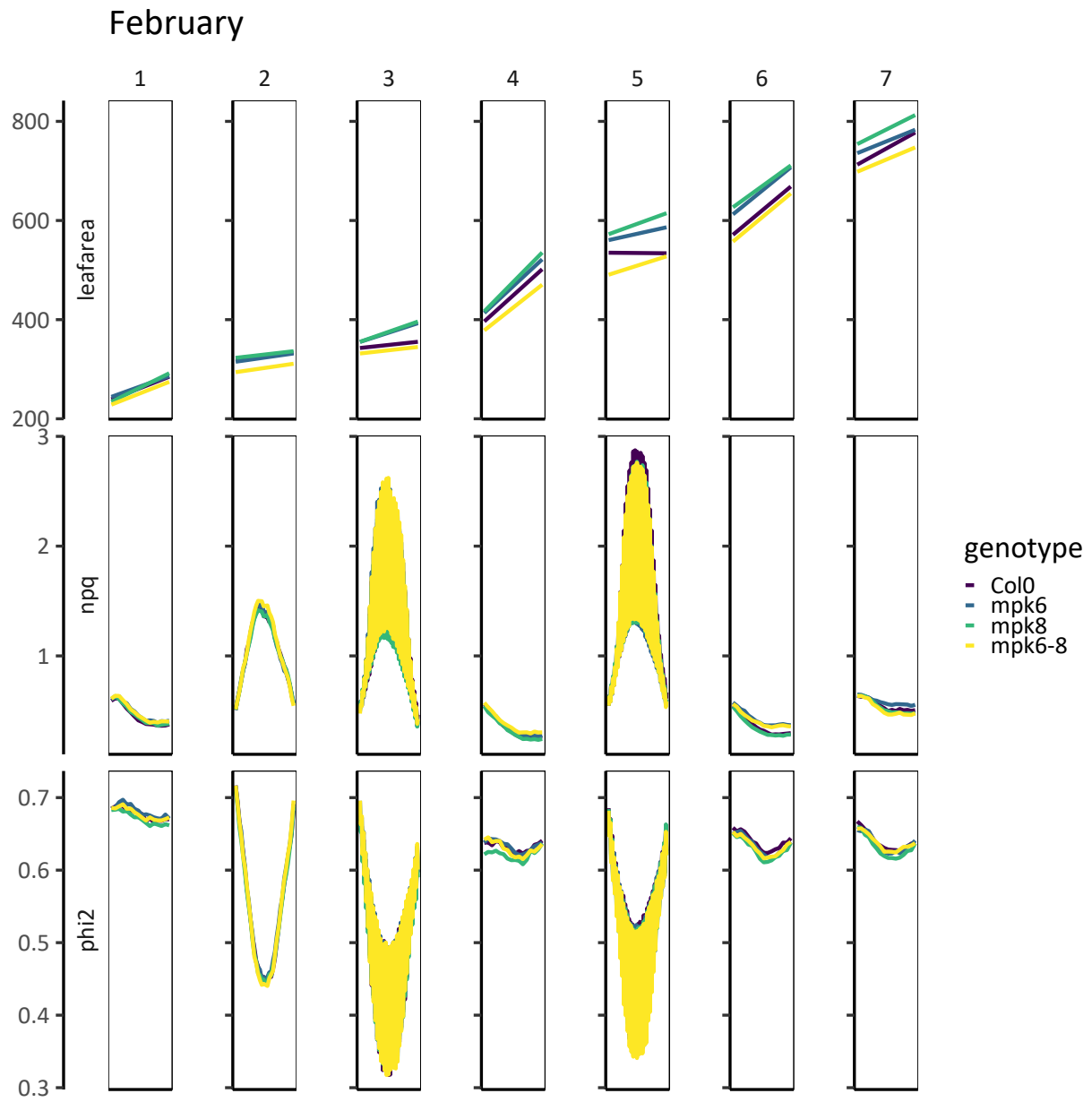
February



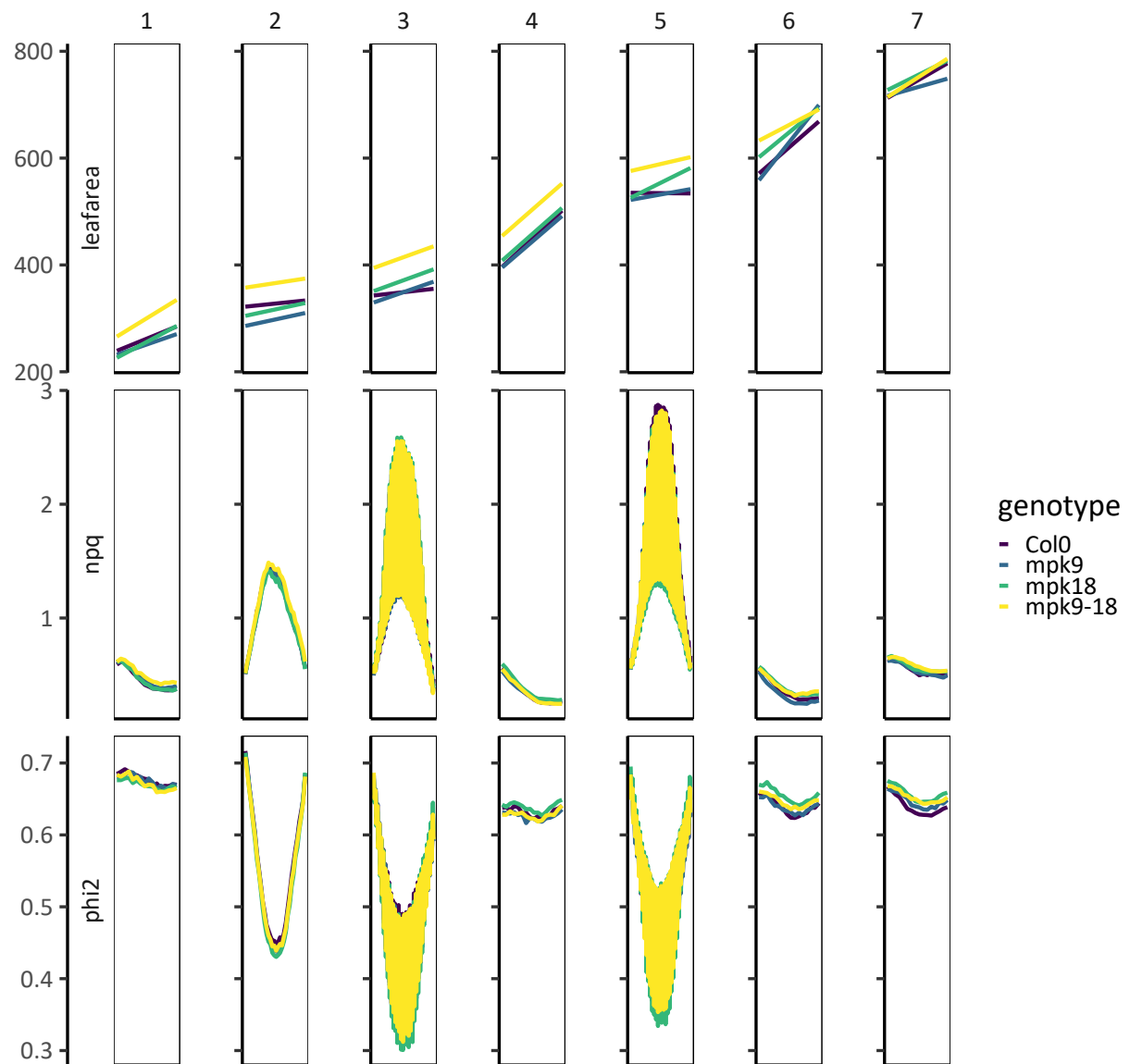




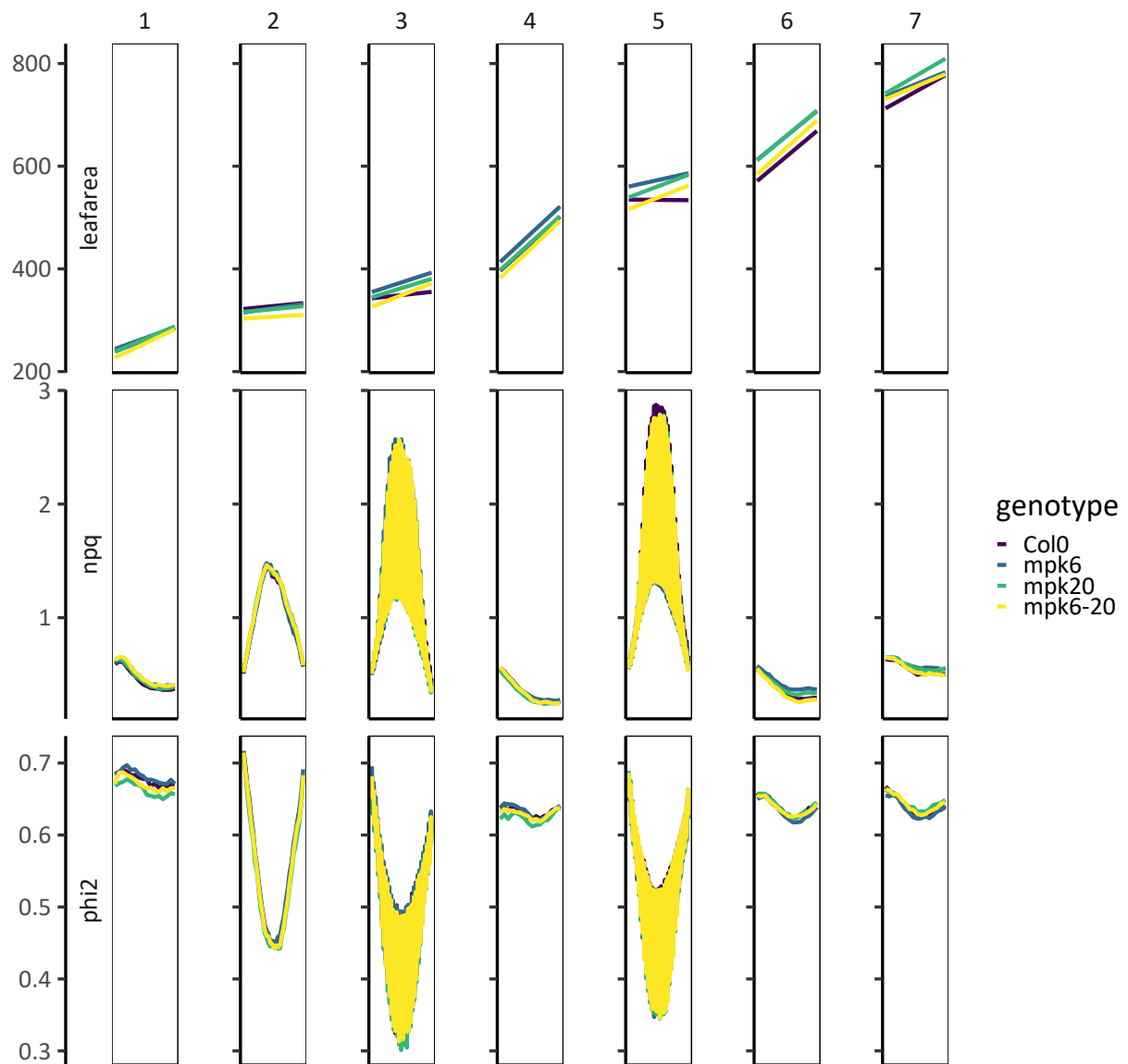


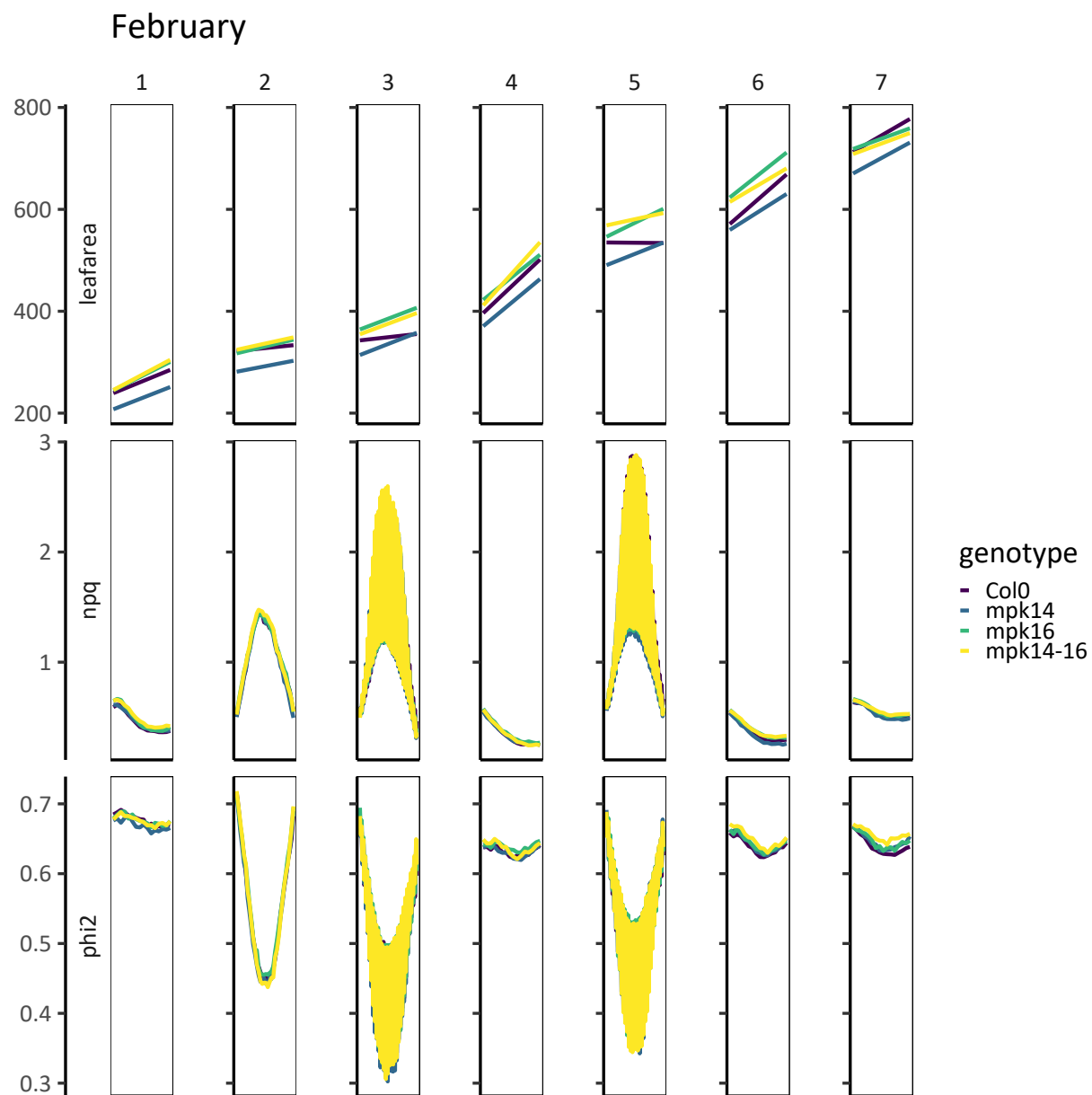


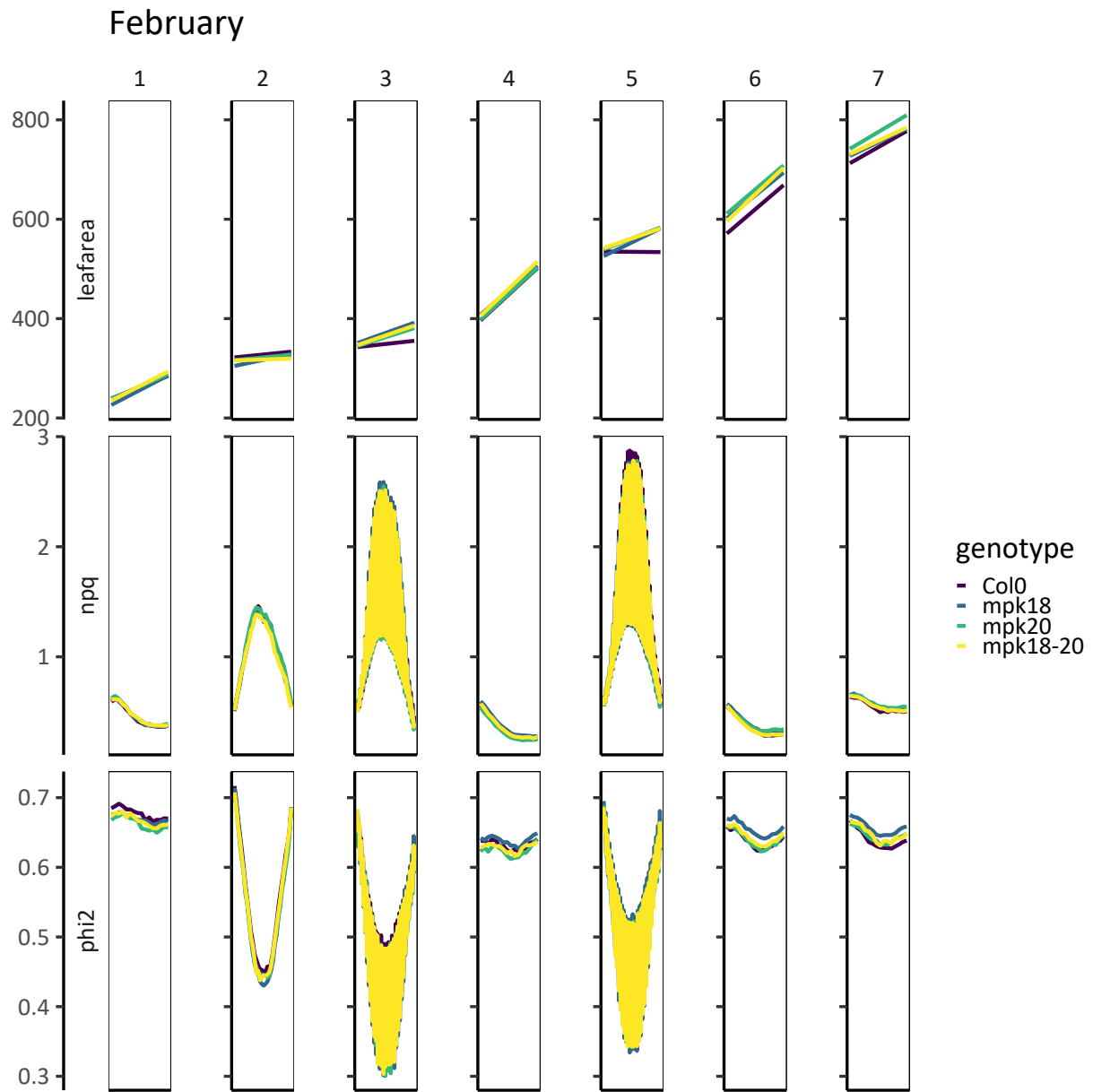
February

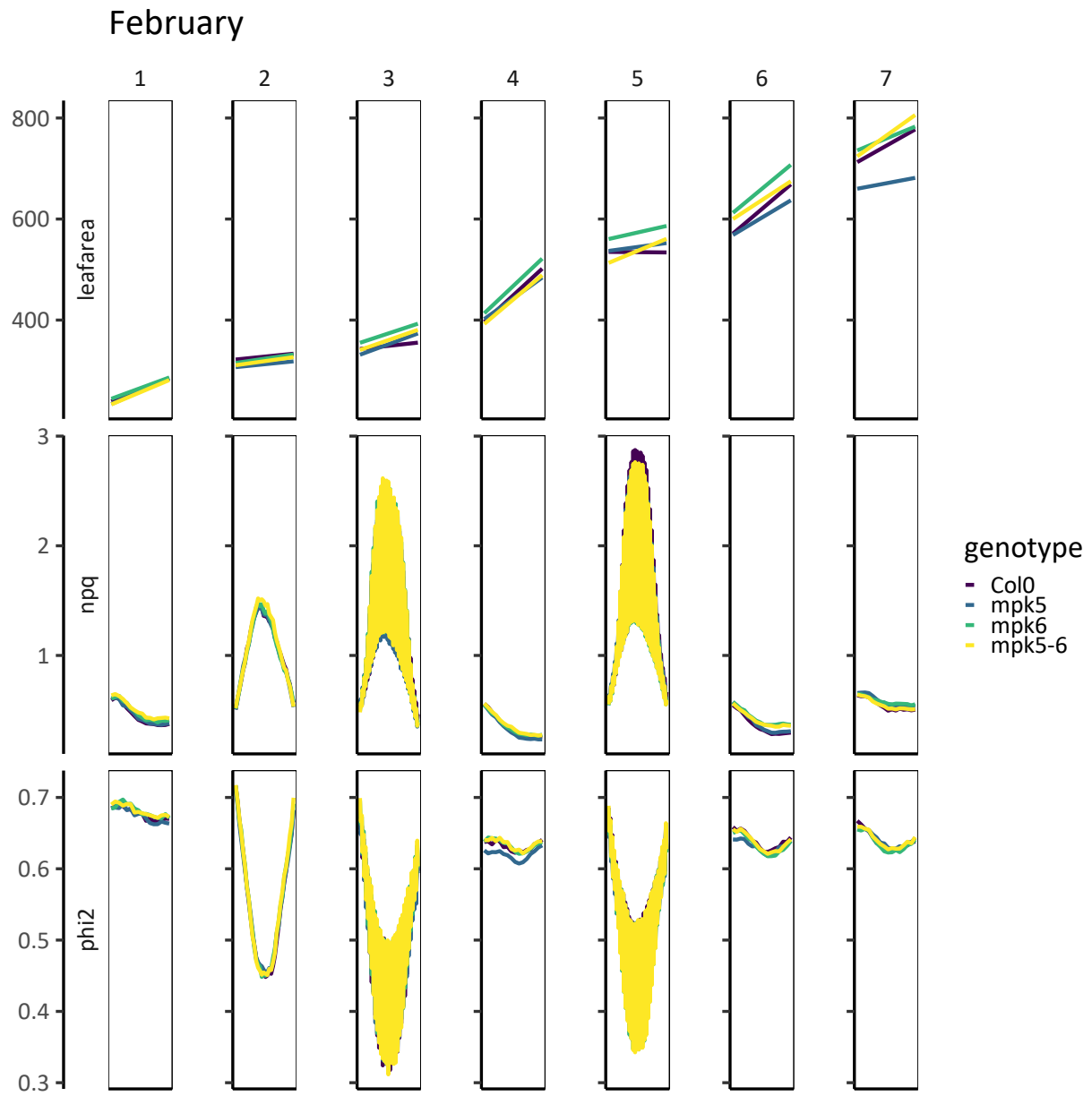


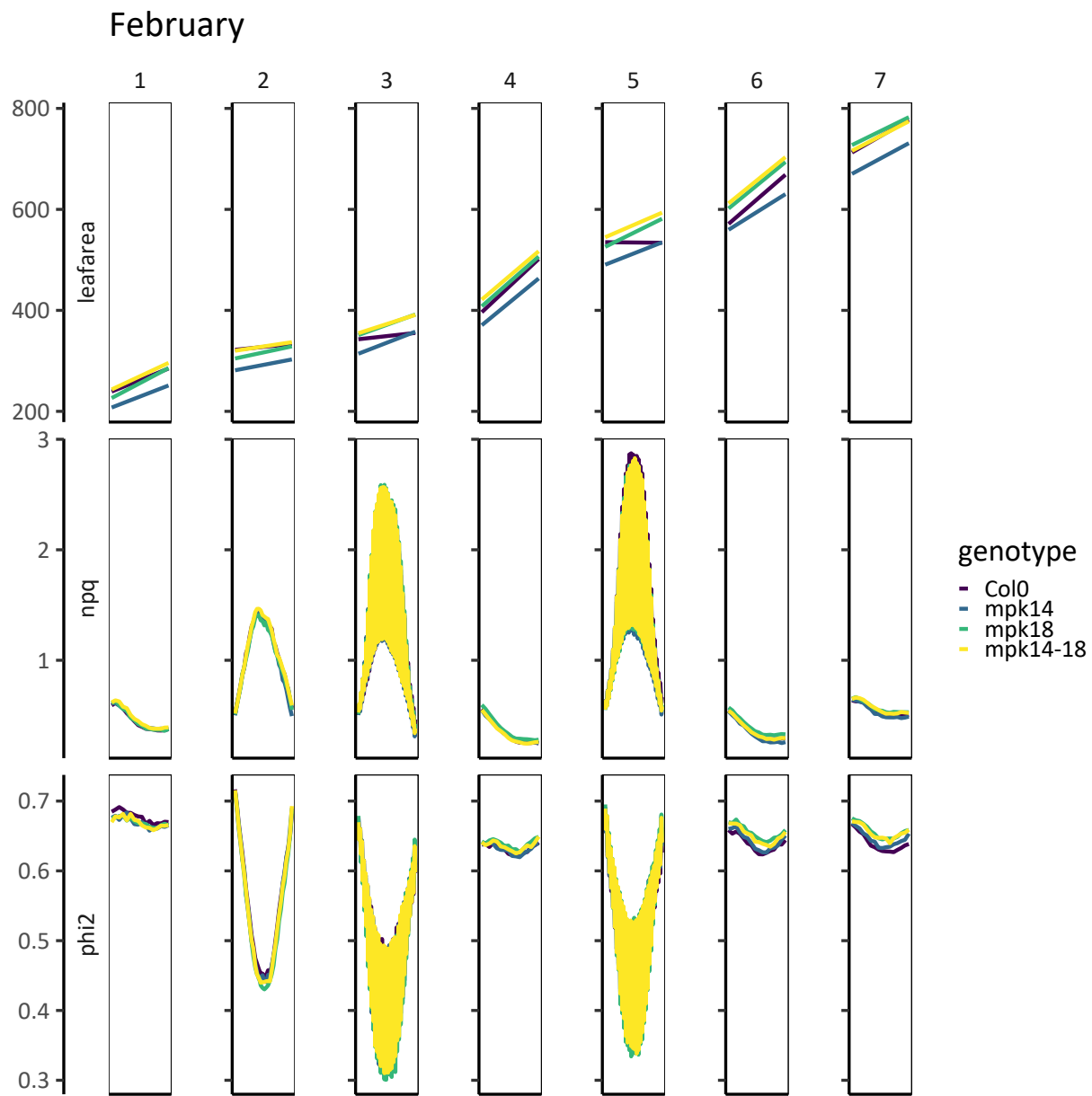
February

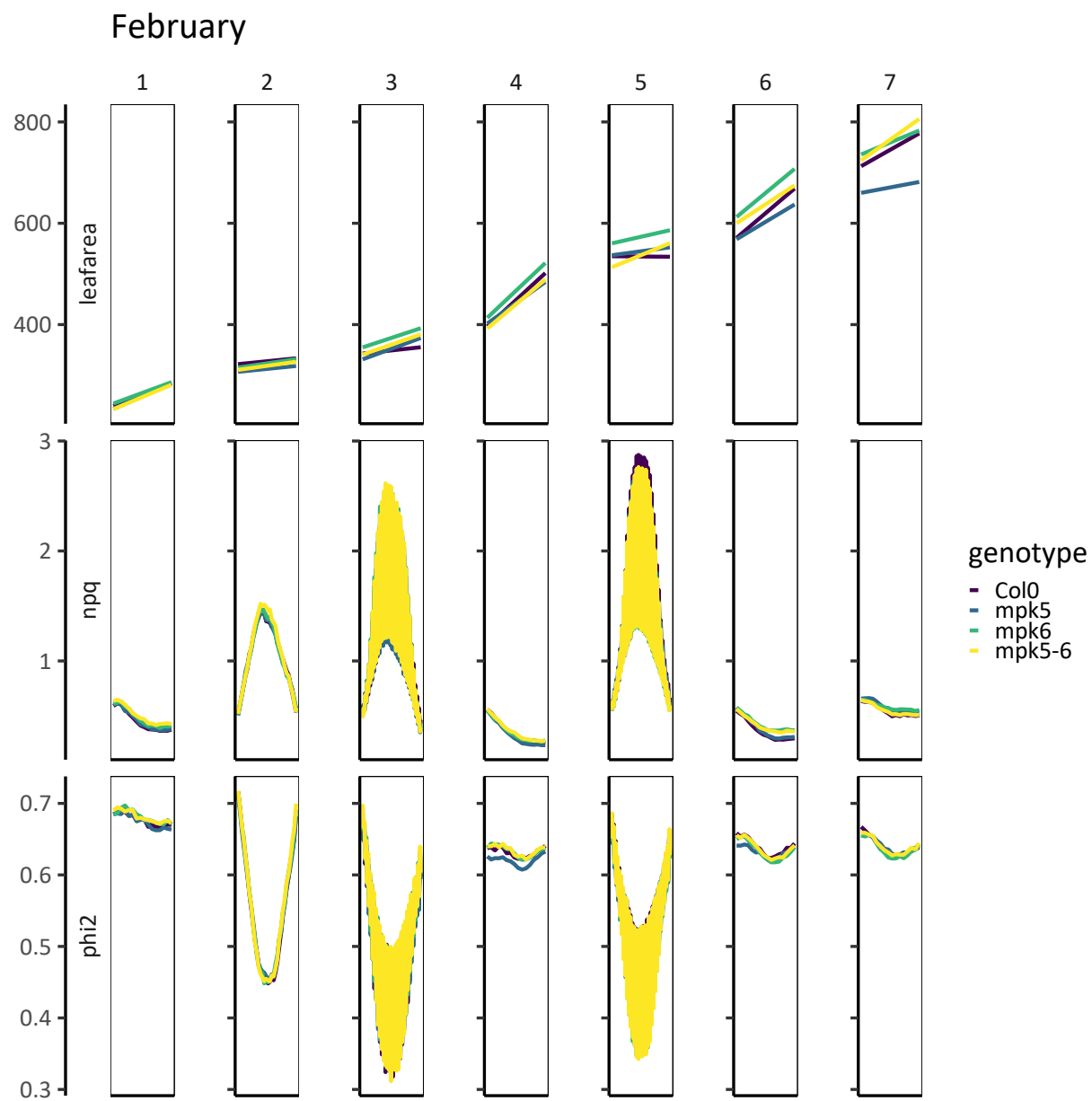


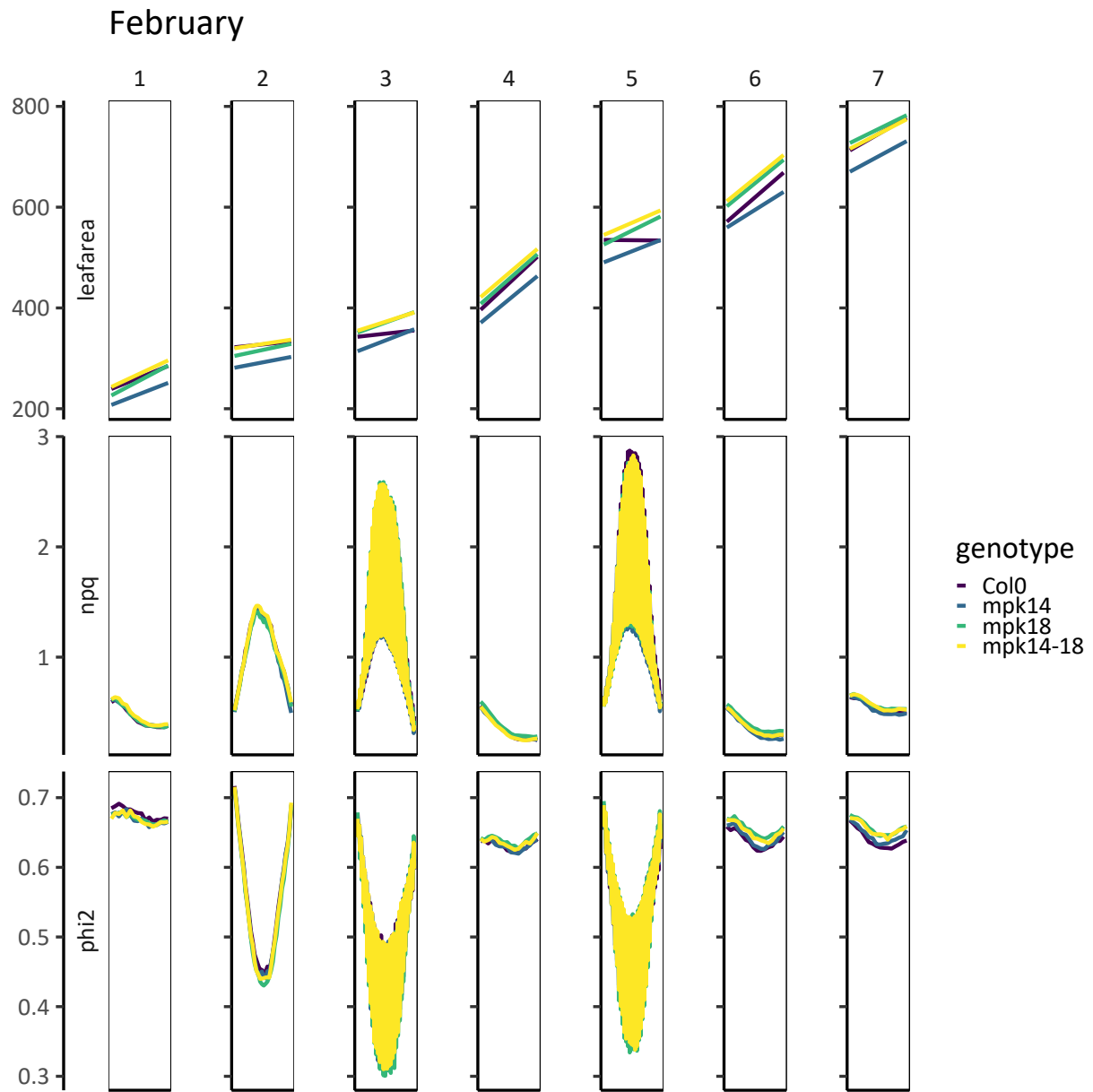


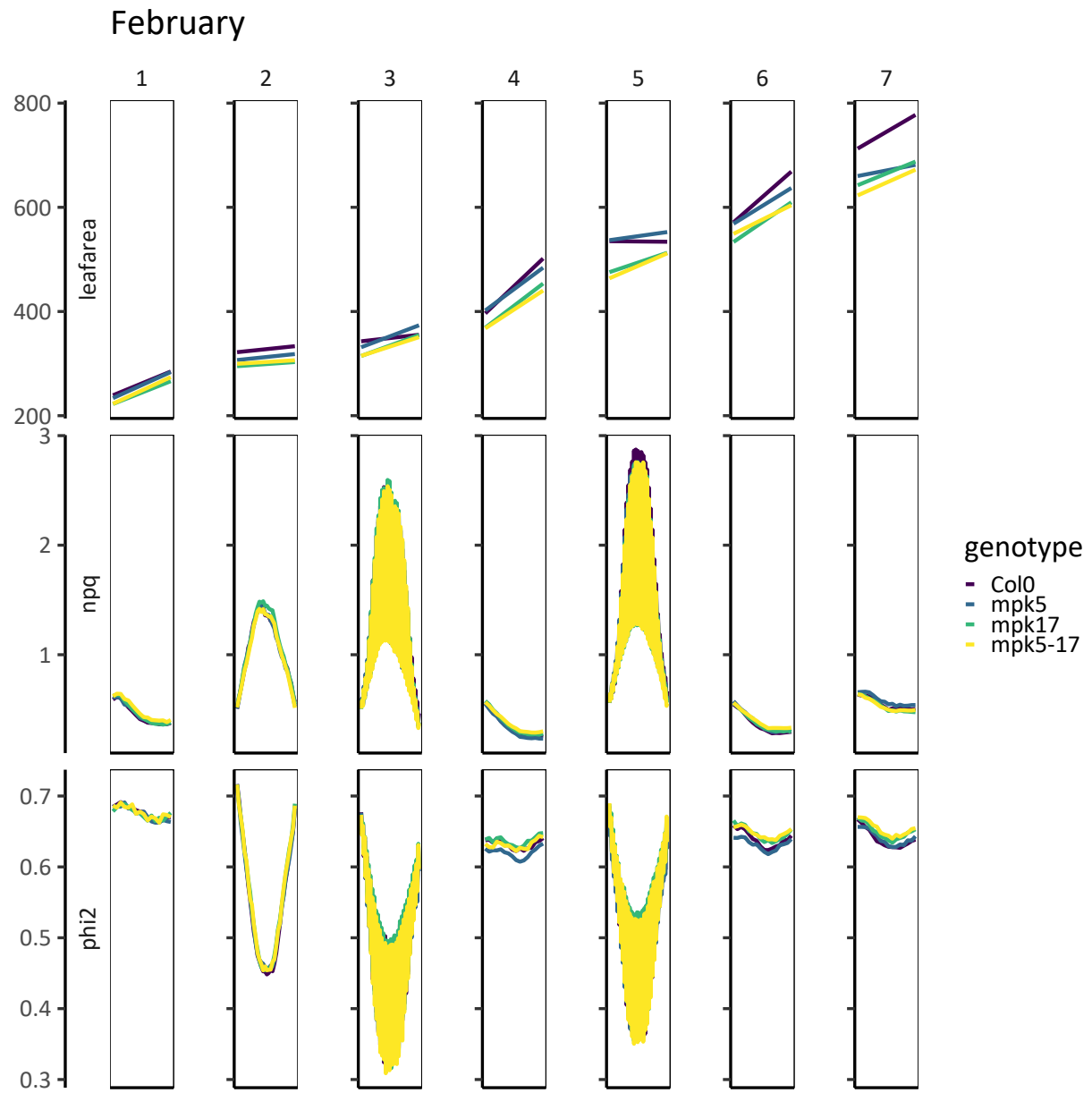


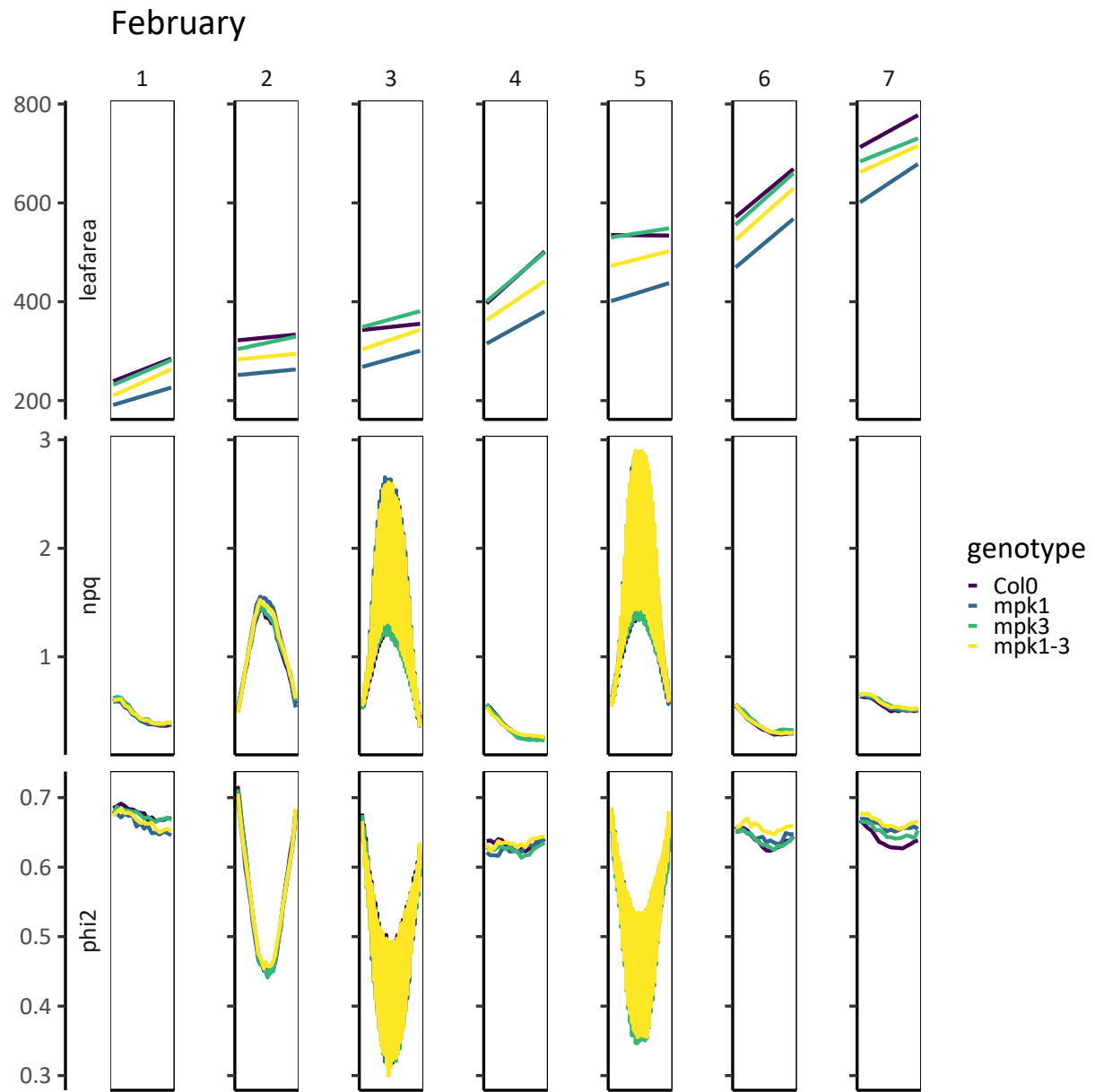


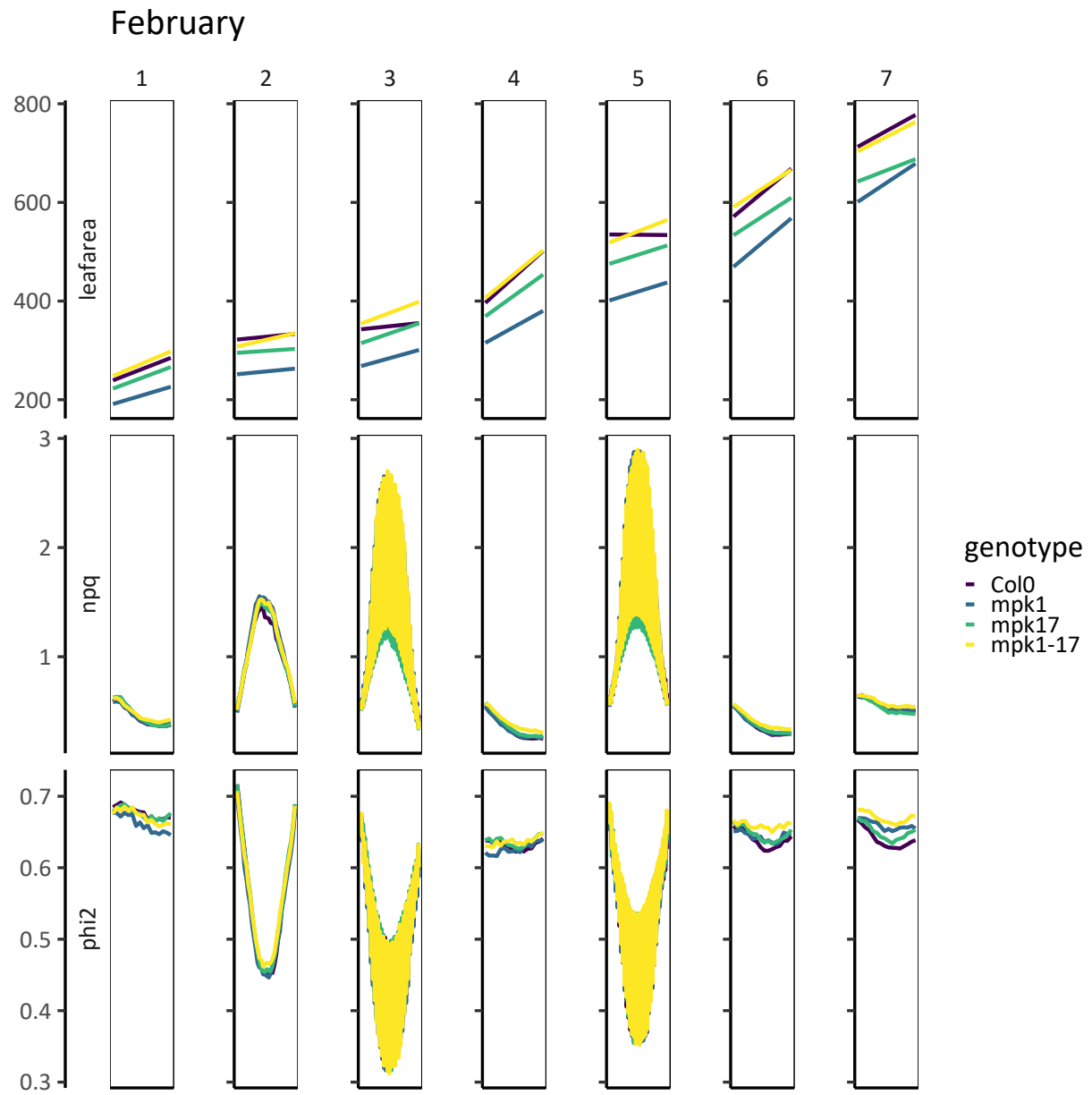


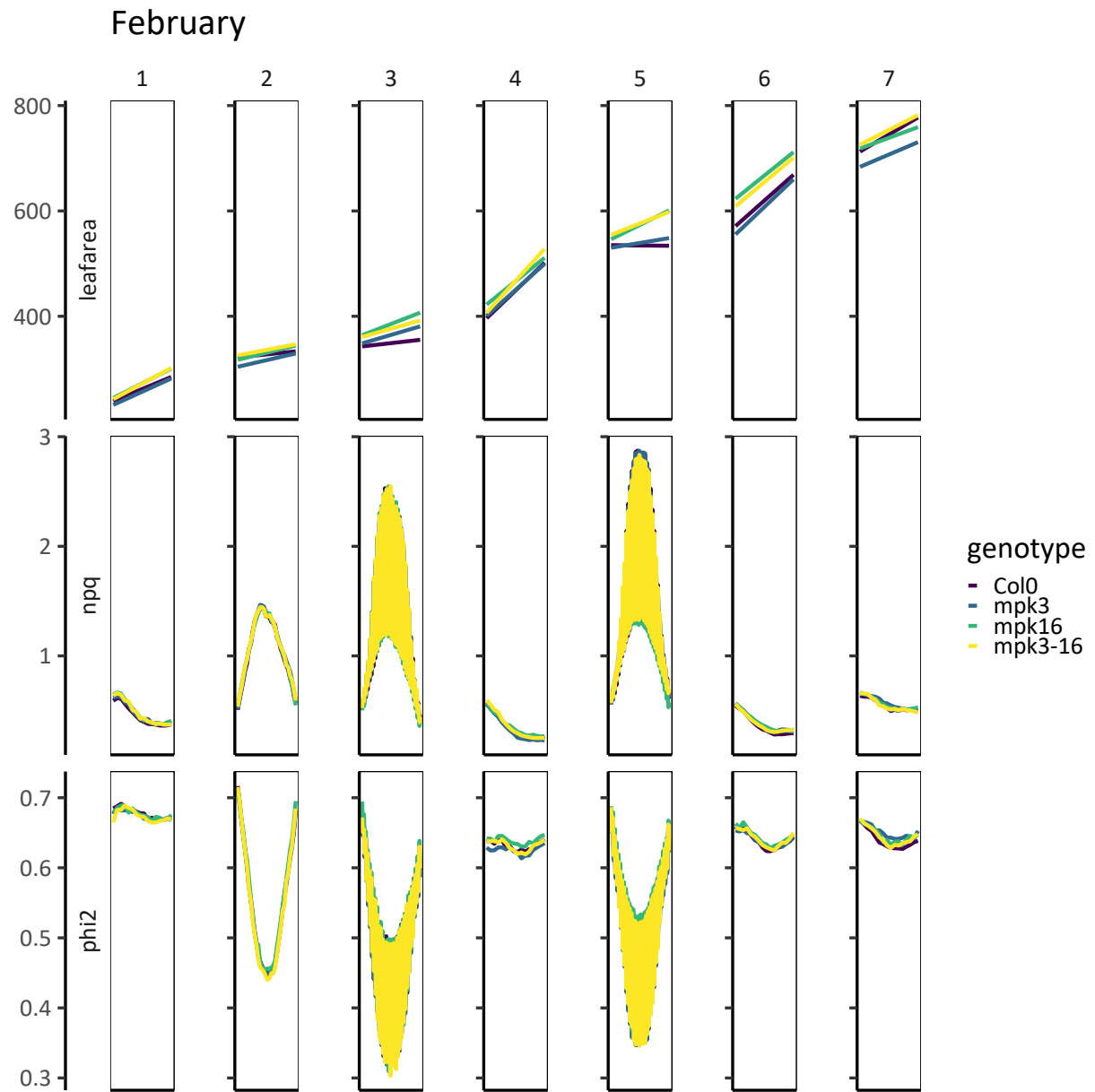












February

