

FQ2

2025-11-07

```
library(lme4)

## Loading required package: Matrix

library(tibble)
library(dplyr)

## 
## Attaching package: 'dplyr'

## The following objects are masked from 'package:stats':
## 
##     filter, lag

## The following objects are masked from 'package:base':
## 
##     intersect, setdiff, setequal, union

library(tidyverse)

## -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
## vforcats    1.0.0    vreadr      2.1.5
## vggplot2    3.5.2    vstringr   1.5.1
## vlubridate  1.9.4    vtidyrm   1.3.1
## vpurrr      1.0.4

## -- Conflicts ----- tidyverse_conflicts() --
## xtidyr::expand() masks Matrix::expand()
## xdplyr::filter() masks stats::filter()
## xdplyr::lag()    masks stats::lag()
## xtidyr::pack()  masks Matrix::pack()
## xtidyr::unpack() masks Matrix::unpack()
## i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors

library(ggthemes)
library(ggrepel)
library(lfmm)
library(RSpectra)
load("pl_lt_t.lmer.RData")
load("growth_light_time.lmer.RData")
source("manhattan_plot.R")
```

```

## 
## Attaching package: 'scales'
## 
## The following object is masked from 'package:purrr':
## 
##     discard
## 
## The following object is masked from 'package:readr':
## 
##     col_factor

```

manhattan plot for 2022

```

pheno_2022 <- ranef(pl_lt_t.lmer)$pop %>%
  as_tibble(rownames = "pop", .name_repair = "unique") %>%
  rename(blup_intercept = `Intercept)...1`,
         blup_light = `mean_light_ly_day2`) %>%
  mutate(model = "Y2022")

## New names:
## * 'Intercept' -> 'Intercept)...1'
## * 'Intercept' -> 'Intercept)...3'

geno_22 <- read.delim("Data/merged_maf_common_UCD2022.tsv", header = TRUE)

Y <- as.matrix(geno_22[,-1])
Y <- t(Y)
X <- matrix(scale(pheno_2022$blup_light), ncol = 1)
rownames(X) <- pheno_2022$pop
colnames(X) <- "blup_light"
common <- intersect(rownames(X), rownames(Y))
Y <- Y[common,]
X <- X[common, , drop = FALSE]
mod.lfmm <- lfmm_ridge(Y = Y,
                         X = X,
                         K = 2)

pv <- lfmm_test(Y = Y,
                  X = X,
                  lfmm = mod.lfmm,
                  calibrate = "gif")
pvalues <- pv$calibrated.pvalue

plot_data = tibble(loci = geno_22[,1], p_val = pv$pvalue[,1], calibrated.pvalue = pv$calibrated.pvalue[,1],
separate_wider_delim(loci, delim = ":" , names = c("chr", "bp")) |>
filter(str_detect(chr, "Chr")) |>
mutate(p.bonf = p.adjust(p_val, method = "bonferroni"),
bp = as.numeric(bp))

manhattan_22 = plot_manhattan(plot_data,
                               chr_col = "chr",
                               pos_col = "bp",

```

```

            value_col = "calibrated.pvalue",
            transform = "neglog10",
            title = "Light association",
            highlight_top_n = 20)

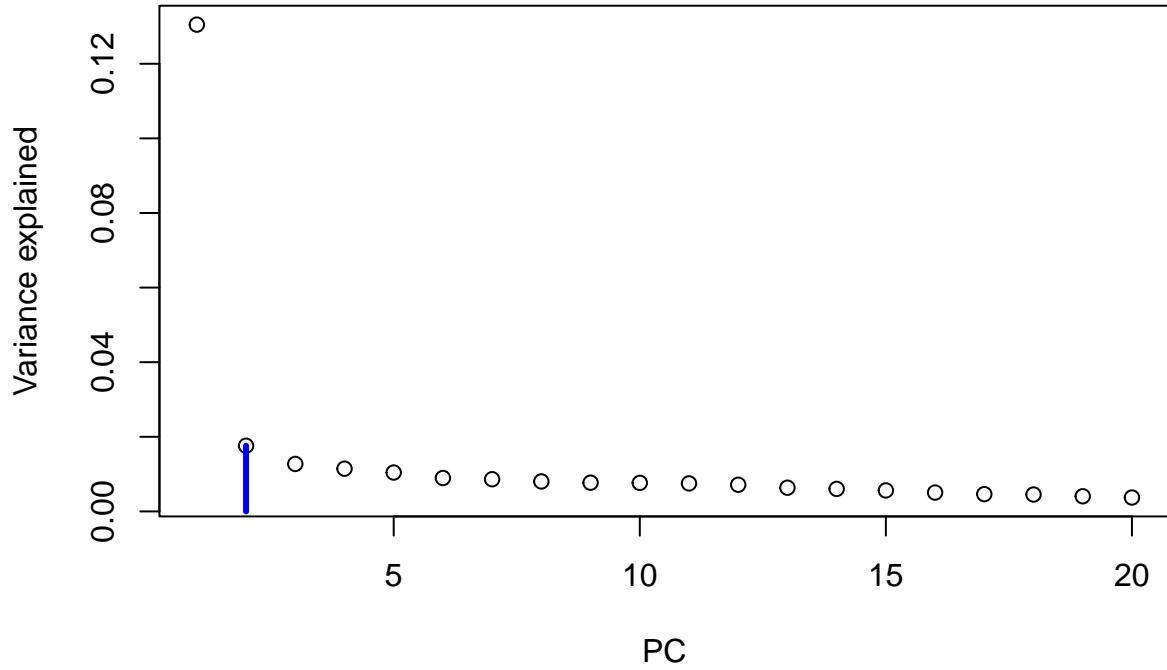
pheno_2023 <- ranef(growth_light_time.lmer)$parent_pop %>%
  as_tibble(rownames = "parent_pop", .name_repair = "unique") %>%
  rename(blup_intercept = `Intercept`...1`,
         blup_light = `weekly_avg_SlrW2`) %>%
  mutate(model = "Y2023")

## New names:
## * '(Intercept)' -> '(Intercept)...1'
## * '(Intercept)' -> '(Intercept)...3'

geno_23 <- read.delim("Data/merged_maf_common_WL2_2023.tsv", header = TRUE)

Y <- as.matrix(geno_23[,-1])
pc <- prcomp(Y)
plot(pc$sdev[1:20]^2, xlab = 'PC', ylab = "Variance explained")
points(2,pc$sdev[2]^2, type = "h", lwd = 3, col = "blue")

```



```

Y <- as.matrix(geno_23[, -1])
Y <- t(Y)
X <- matrix(scale(pheno_2023$blup_light), ncol = 1)
rownames(X) <- pheno_2023$parent_pop
colnames(X) <- "blup_light"
common <- intersect(rownames(X), rownames(Y))
Y <- Y[common,]
X <- X[common, , drop = FALSE]
mod.lfmm <- lfmm_ridge(Y = Y,
                         X = X,
                         K = 2)

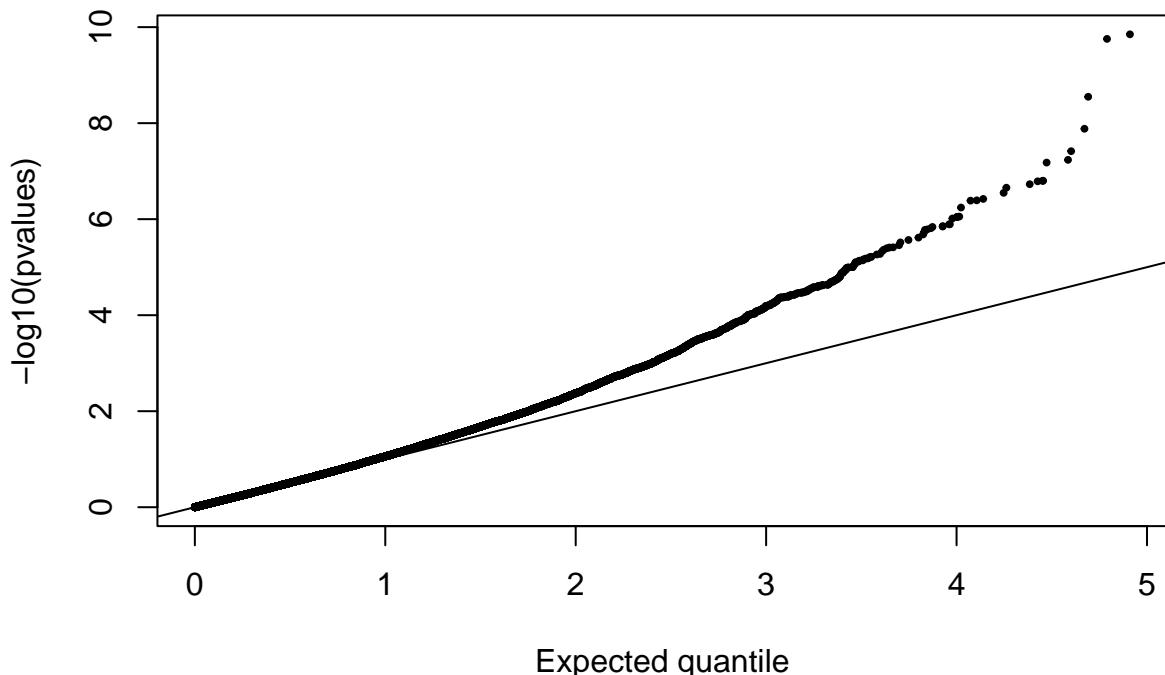
```

```

pv <- lfmm_test(Y = Y,
                  X = X,
                  lfmm = mod.lfmm,
                  calibrate = "gif")
pvalues <- pv$calibrated.pvalue

#QQ plot
qqplot(rexp(length(pvalues), rate = log(10)),
       -log10(pvalues), xlab = "Expected quantile",
       pch = 19, cex = .4)
abline(0,1)

```



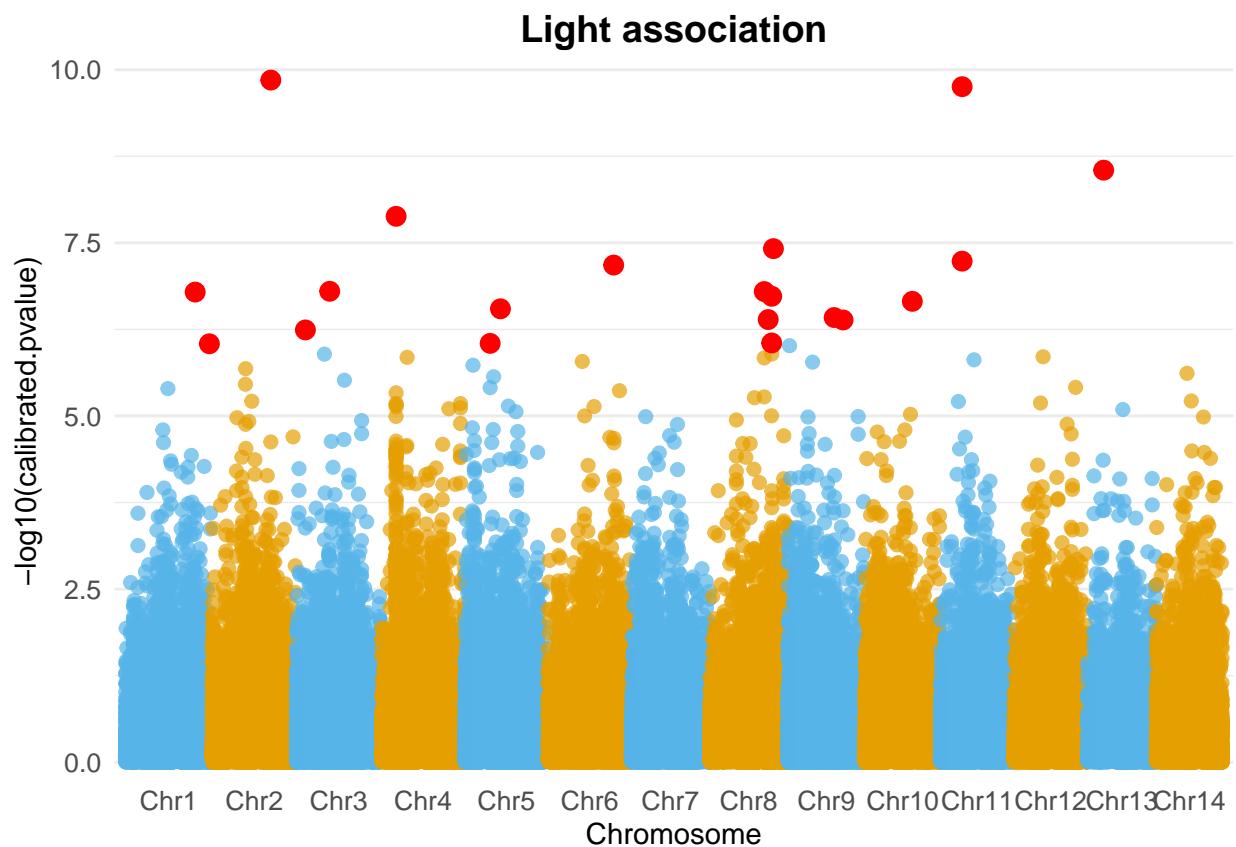
```

plot_data = tibble(loci = geno_23[,1], p_val = pv$pvalue[,1], calibrated.pvalue = pv$calibrated.pvalue[,1],
separate_wider_delim(loci, delim = ":" , names = c("chr", "bp")) |>
filter(str_detect(chr, "Chr")) |>
mutate(p.bonf = p.adjust(p_val, method = "bonferroni"),
bp = as.numeric(bp))

manhattan_23 = plot_manhattan(plot_data,
                               chr_col = "chr",
                               pos_col = "bp",
                               value_col = "calibrated.pvalue",
                               transform = "neglog10",
                               title = "Light association",
                               highlight_top_n = 20)

print(manhattan_23)

```



```

# Save plot
ggsave("light_interaction23.png", manhattan_23, width = 12, height = 6, dpi = 300)

```

```
library(cowplot)
```

```
##  
## Attaching package: 'cowplot'
```

```

## The following object is masked from 'package:ggthemes':
##
##     theme_map

## The following object is masked from 'package:lubridate':
##
##     stamp

manhattan_tot <- plot_grid(manhattan_22, manhattan_23, ncol = 2, labels = c("2022", "2023"), rel_widths

geno_table = Y
colnames(geno_table) = geno_23[, 1]

freq_plot_data = as.data.frame(X) |>
  rownames_to_column(var = "Population") |>
  mutate(loci1 = as.numeric(geno_table[, "Chr2:18026903"]),
  loci2 = as.numeric(geno_table[, "Chr11:6546598"]),
  loci3 = as.numeric(geno_table[, "Chr13:4965604"])) |>
  pivot_longer(cols = loci1:loci3, names_to = "loci", values_to = "frequency")

freq_plot_data |>
  ggplot(aes(x = blup_light, y = frequency, label = Population)) +
  geom_point(size = 3) +
  geom_smooth(method = "lm", colour = "red") +
  geom_text_repel(size = 6) +
  facet_wrap(~loci, nrow = 1, scales = "free_y") +
  theme_minimal() +
  labs(x = "BLUP light", y = "Allele frequency")

## `geom_smooth()` using formula = 'y ~ x'

## Warning: The following aesthetics were dropped during statistical transformation: label.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: label.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?
## The following aesthetics were dropped during statistical transformation: label.
## i This can happen when ggplot fails to infer the correct grouping structure in
##   the data.
## i Did you forget to specify a 'group' aesthetic or to convert a numerical
##   variable into a factor?

## Warning: ggrepel: 9 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

## Warning: ggrepel: 6 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```

```
## Warning: ggrepel: 5 unlabeled data points (too many overlaps). Consider  
## increasing max.overlaps
```

