

FQ1

2025-10-27

```
#install.packages("pkgbuild")
#pkgbuild::check_build_tools(debug = TRUE)
#install.packages(c("Rcpp", "RcppEigen", "cli", "rlang", "vctrs", "pillar", "lifecycle"))
#install.packages(c("devtools", "remotes"))
#remotes::install_github("bcm-uga/lfmm",
#                        dependencies = TRUE,
#                        build_vignettes = FALSE,
#                        force = TRUE)
#install.packages("RSpectra")

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    vtidyrr    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")
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

ranef(pl_lt_t.lmer)$pop

##          (Intercept) mean_light_ly_day2  (Intercept)      week
## BH      0.378201820   0.181455466 -0.0368452799 -1.350185e-02
## CC      0.003852729   0.001848481  0.0027535429  1.009031e-03
## CP2     -0.035813805  -0.017182918 -0.0010164742 -3.724812e-04
## DPR      0.046171917   0.022152582  0.0201096589  7.369128e-03
## FR      -0.037924633  -0.018195661  0.0002468720  9.046926e-05
## IH      -0.060123529  -0.028846352 -0.0002463103 -9.026415e-05
## LV1     -0.007806403  -0.003745393 -0.0006212819 -2.276668e-04
## LV3     -0.094419680  -0.045301123 -0.0046119563 -1.690037e-03
## LVTR    -0.017333307  -0.008316256 -0.0012304227 -4.508846e-04
## LVTR1   -0.104864146  -0.050312218 -0.0039154244 -1.434796e-03
## SC       0.142156078   0.068204318 -0.0071389157 -2.616037e-03
## SQ1     -0.085706122  -0.041120491 -0.0009667857 -3.542761e-04
## SQ2     -0.035580273  -0.017070872 -0.0029897426 -1.095582e-03
## SQ3     -0.060711094  -0.029128257 -0.0010998341 -4.030303e-04
## TM2      0.275791970   0.132320782  0.0312759461  1.146097e-02
## WL1     -0.127059752  -0.060961332  0.0025501706  9.345026e-04
## WL2     -0.032785916  -0.015730183 -0.0032584037 -1.194030e-03
## WR      -0.035261922  -0.016918133 -0.0024652688 -9.033904e-04
## WV      -0.002874061  -0.001378931 -0.0003568963 -1.307835e-04
## Y011    -0.019674190  -0.009439376 -0.0021028205 -7.705723e-04
## Y04     -0.008900585  -0.004270365 -0.0006776643 -2.483280e-04
## Y07     -0.040787750  -0.019569340 -0.0070035206 -2.566421e-03
## Y08     -0.038547344  -0.018494428 -0.0033019093 -1.209975e-03

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:

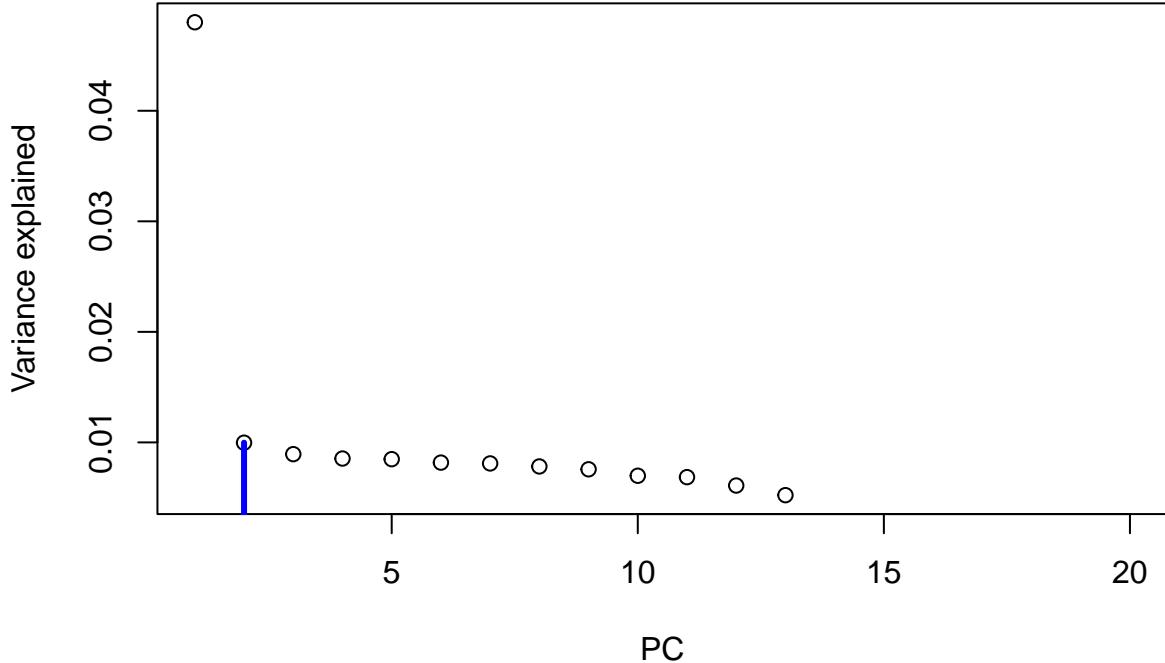
```

## * '(Intercept)' -> '(Intercept)...1'
## * '(Intercept)' -> '(Intercept)...3'

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

Y <- as.matrix(geno_22[,-1])
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")

```



```

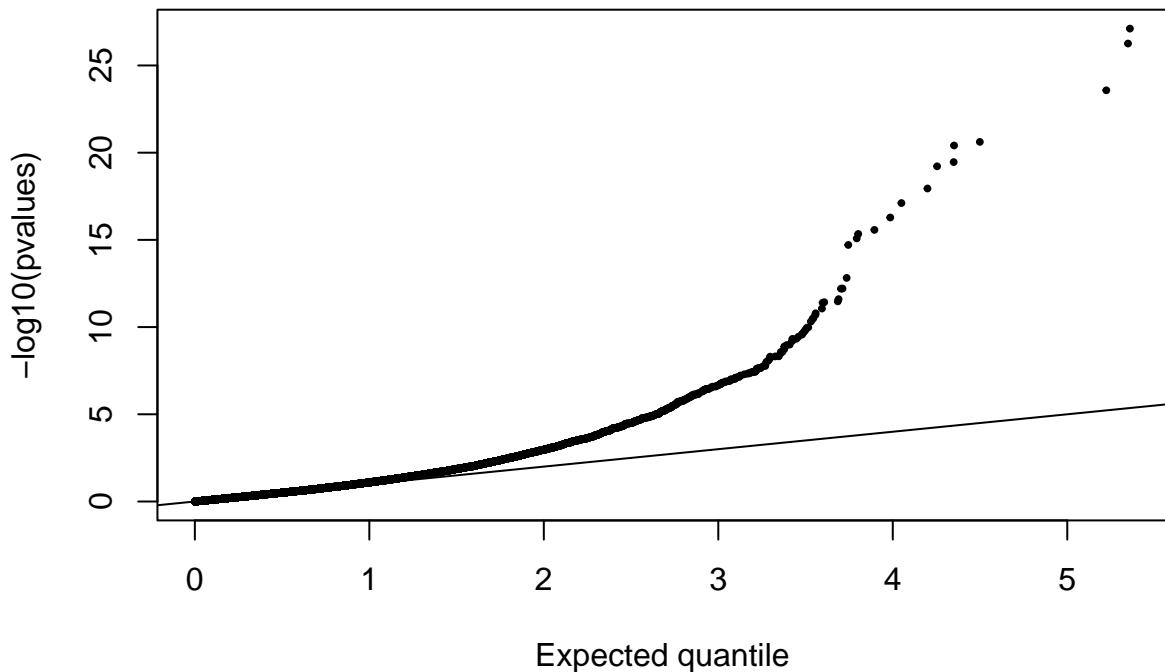
Y <- as.matrix(geno_22[,-1])
Y <- t(Y)
X <- matrix(scale(pheno_2022$blup_light), ncol = 1)
rownames(X) <- pheno_2022$pop
colnames(X) <- "blup_light"
common <- intersect(rownames(X), rownames(Y))
Y <- Y[common,]
X <- X[common, , drop = FALSE]
mod.lfmm <- lfmm_ridge(Y = Y,
                        X = X,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

## performs association testing using the fitted model:
pv <- lfmm_test(Y = Y,
                  X = X,

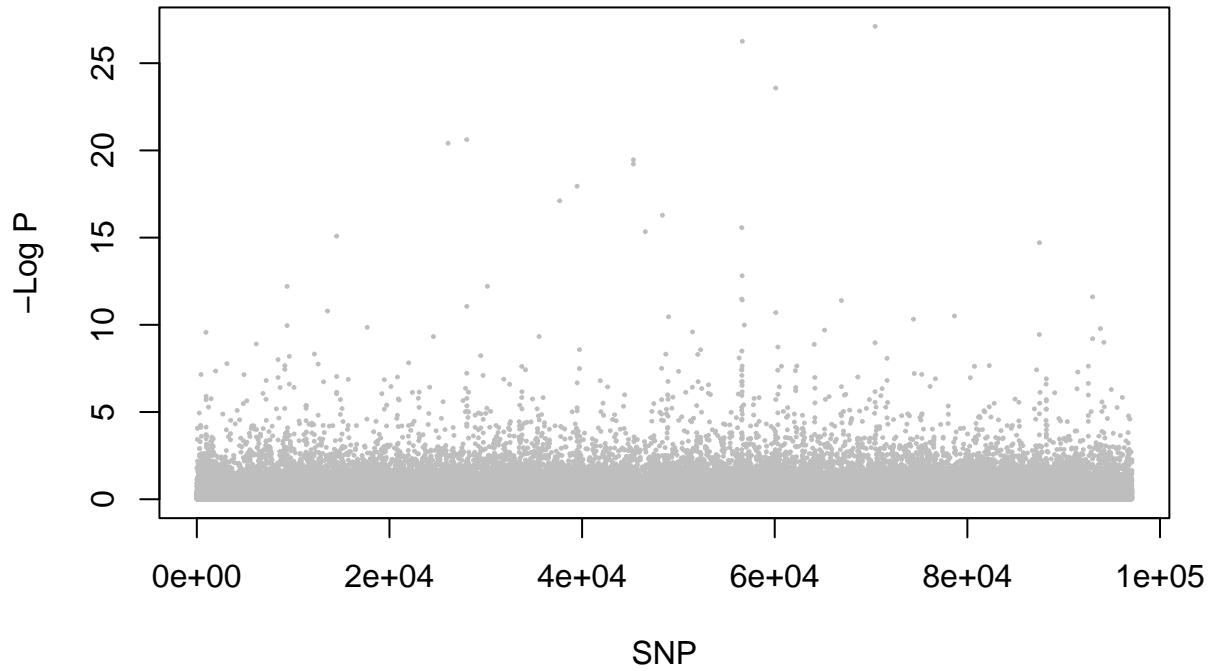
```

```
    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)
```



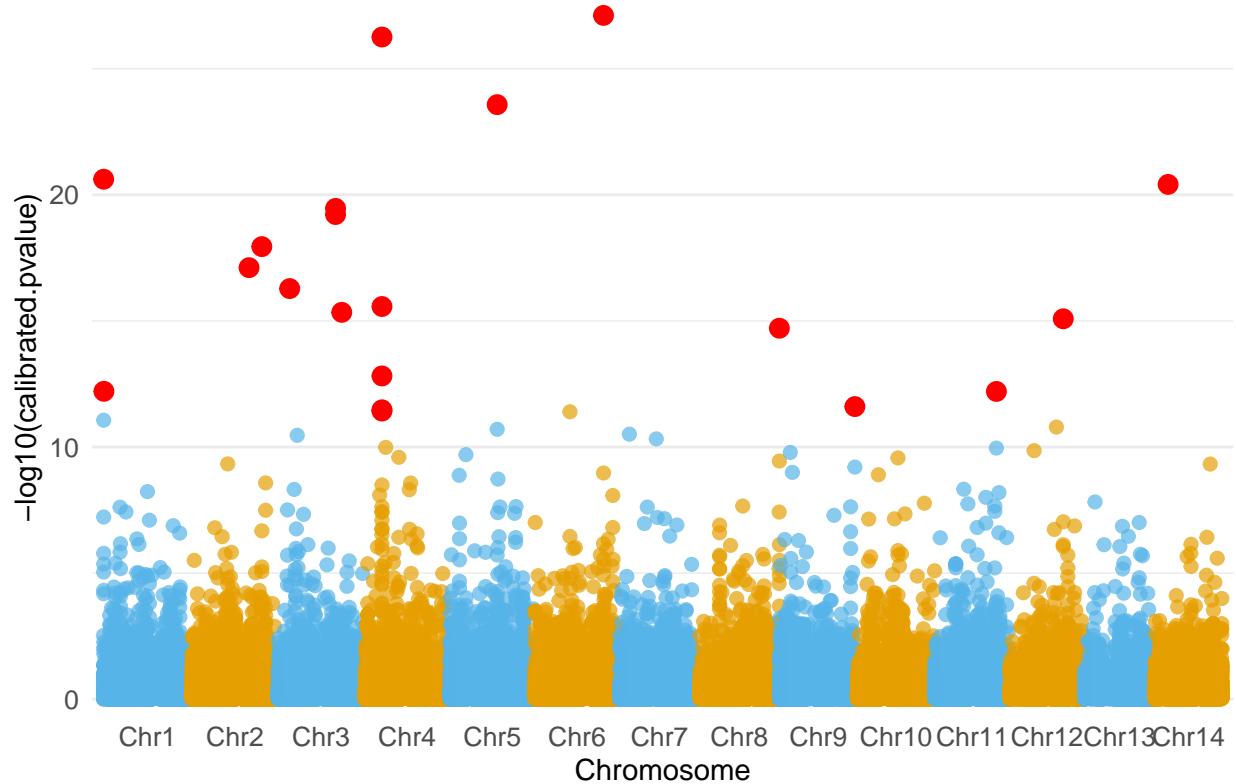
```
## Manhattan plot
plot(-log10(pvalues),
      pch = 19,
      cex = .2,
      xlab = "SNP", ylab = "-Log P",
      col = "grey")
```



Create a data frame for plotting the manhattan plot.

```
plot_data = tibble(loci = geno_22[,1], p_val = pv$pvalue[,1], calibrated.pvalue = pv$calibrated.pvalue[  
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))  
  
light_manhattan = plot_manhattan(plot_data,  
                                chr_col = "chr",  
                                pos_col = "bp",  
                                value_col = "calibrated.pvalue",  
                                transform = "neglog10",  
                                title = "Light association",  
                                highlight_top_n = 20)  
  
print(light_manhattan)
```

Light association



```
# Save plot
ggsave("light_interaction22.png", light_manhattan, width = 12, height = 6, dpi = 300)
```

Plot the correlation of allele frequency and the blups

```
geno_table = Y
colnames(geno_table) = geno_22[, 1]

freq_plot_data = as.data.frame(X) |>
  rownames_to_column(var = "Population") |>
  mutate(loci1 = as.numeric(geno_table[, "Chr6:20652277"]),
  loci2 = as.numeric(geno_table[, "Chr4:5383224"]),
  loci3 = as.numeric(geno_table[, "Chr5:14295930"])) |>
  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: 4 unlabeled data points (too many overlaps). Consider
## increasing max.overlaps

```

