Plotting Results

Evan Gorstein

2023-07-30

Read in the data:

```
gene_results <- read_csv("../sim_results/best_results.csv")
```

```
## Rows: 5517 Columns: 1034
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): setting, data_id
## dbl (1032): lambda, loglike, aic, bic, n_nz, tp, sigma, psi_1, psi_2, psi_3,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
gwas_results <- read_csv("../sim_results/gwas/best_results.csv")
```

```
## Rows: 2000 Columns: 1035
## -- Column specification --------------------------------------------------------
## Delimiter: ","
## chr    (2): setting, data_id
## dbl (1033): lambda, loglike, aic, bic, n_nz, tp, sigma, psi_1, psi_2, psi_3,...
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

These data frames contain the fits with the best two BIC's (across all lambdas) for each dataset generated under each setting.

Gene expression results

Let's check on the count for each setting

```
gene_results %>%
  count(setting)
```

```
## # A tibble: 28 x 2
##    setting                                              n
##    <chr>                                            <int>
##  1 dim1000_random1_rho0.0_nz10_covid_lasso-results    197
##  2 dim1000_random1_rho0.0_nz10_covid_scad-results     200
##  3 dim1000_random1_rho0.0_nz5_covid_lasso-results     200
##  4 dim1000_random1_rho0.0_nz5_covid_scad-results      200
##  5 dim1000_random1_rho0.6_nz10_covid_lasso-results    198
##  6 dim1000_random1_rho0.6_nz10_covid_scad-results     200
```

```
##  7 dim1000_random1_rho0.6_nz5_covid_lasso-results      200
##  8 dim1000_random1_rho0.6_nz5_covid_scad-results       200
##  9 dim1000_random3_rho0.0_nz10_covdiag_lasso-results   200
## 10 dim1000_random3_rho0.0_nz10_covdiag_scad-results    200
## # ... with 18 more rows
```

These numbers should all be 200. Unfortunately, they're not all 200 because for some settings not all 100 data sets receive top two results because some datasets failed to converge for all lambdas.

Looking at just the best lambda:

```
best_gene_results <- gene_results %>%
  arrange(setting, data_id, bic) %>%
  group_by(setting, data_id) %>%
  filter(row_number() == 1) %>%
  ungroup()

count(best_gene_results, setting)
```

```
## # A tibble: 28 x 2
##    setting                                              n
##    <chr>                                            <int>
##  1 dim1000_random1_rho0.0_nz10_covid_lasso-results     99
##  2 dim1000_random1_rho0.0_nz10_covid_scad-results     100
##  3 dim1000_random1_rho0.0_nz5_covid_lasso-results     100
##  4 dim1000_random1_rho0.0_nz5_covid_scad-results      100
##  5 dim1000_random1_rho0.6_nz10_covid_lasso-results     99
##  6 dim1000_random1_rho0.6_nz10_covid_scad-results     100
##  7 dim1000_random1_rho0.6_nz5_covid_lasso-results     100
##  8 dim1000_random1_rho0.6_nz5_covid_scad-results      100
##  9 dim1000_random3_rho0.0_nz10_covdiag_lasso-results  100
## 10 dim1000_random3_rho0.0_nz10_covdiag_scad-results   100
## # ... with 18 more rows
```

These numbers should all be 100.

Plotting gene results for nz = 5

False positive rate

In each of these boxplots, the data being plotted are the FPRs of a 100 models fit to 100 different simulated datasets. In particular, there is one model for each dataset, the model that minimizes BIC.

```
fp <- best_gene_results %>%
  filter(str_detect(setting, "nz5")) %>%
  mutate(
    cor = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "covid_(.*)-", group = 1),
    p = factor(
      str_extract(setting, "dim([0-9]+)", group = 1),
      levels = c('500', "1000")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(cor == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = (n_nz - tp)/(as.numeric(as.character(p))-5), col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9)) +
  facet_grid(`Correlated predictors` ~ p) +
```

```
  labs(y = "False positive rate") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        #strip.text.x = element_text(size = 20),
        #strip.text.y = element_text(size = 10),
        text = element_text(size = 15),
        axis.text.y = element_text(size = 10)) +
  scale_y_continuous(labels = scales::percent, sec.axis = sec_axis(~ . , name = "Correlated predictors?"
  scale_x_continuous(sec.axis = sec_axis(~ . , name = "Fixed effect dimension", breaks = NULL, labels =
  scale_color_brewer(type = "qual")

ggsave(
    "../plots/fp_nz5.pdf",
    fp,
    width = 20,
    height = 15,
    units = "cm"
  )
```

True positives

```
tp <- best_gene_results %>%
  filter(str_detect(setting, "nz5")) %>%
  mutate(
    cor = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "covid_(.*)-", group = 1),
    p = factor(
      str_extract(setting, "dim([0-9]+)", group = 1),
      levels = c('500', "1000")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(cor == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = tp, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9)) +
  facet_grid(`Correlated predictors` ~ p) +
  labs(y = "True positives (out of 5)") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        #strip.text.x = element_text(size = 20),
        #strip.text.y = element_text(size = 10),
        text = element_text(size = 15),
        axis.text.y = element_text(size = 10)) +
  scale_y_continuous(labels = scales::label_number(accuracy=1), limits = c(0,5),
                     sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels =
  scale_x_continuous(sec.axis = sec_axis(~ . , name = "Fixed effect dimension", breaks = NULL, labels =
  scale_color_brewer(type = "qual")

ggsave(
  "../plots/tp_nz5.pdf",
  tp,
  width = 20,
  height = 15,
  units = "cm"
```

```
)
```

All exactly 5! Perfect results

Parameter estimates

How do we do at estimating the unpenalized intercept

```
df_true <- data.frame(y_value = 1)

unp <- best_gene_results %>%
  filter(str_detect(setting, "nz5")) %>%
  mutate(
    cor = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "covid_(.*)-", group = 1),
    p = factor(
      str_extract(setting, "dim([0-9]+)", group = 1),
      levels = c('500', "1000")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(cor == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = beta_1, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9)) +
  geom_hline(data = df_true, aes(yintercept = y_value, linetype = "True Parameter")) +
  facet_grid(`Correlated predictors` ~ p) +
  labs(y = "Estimated intercept") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        #strip.text.x = element_text(size = 20),
        #strip.text.y = element_text(size = 10),
        legend.title = element_blank(),
        text = element_text(size = 15),
        axis.text.y = element_text(size = 10),
        legend.text = element_text(size = 10)) +
  scale_y_continuous(labels = scales::label_number(accuracy=.1), limits = c(0,2),
                     sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels =
  scale_x_continuous(sec.axis = sec_axis(~ . , name = "Fixed effect dimension", breaks = NULL, labels =
  scale_color_brewer(type = "qual",
                     labels = c("Estimates with LASSO",
                                "Estimates with SCAD"))


ggsave(
  "../plots/unp_nz5.pdf",
  unp,
  width = 20,
  height = 15,
  units = "cm"
)
```

How do we do at estimating the penalized second component of beta?

```
df_true <- data.frame(y_value = 2)

penal <- best_gene_results %>%
```

```
  filter(str_detect(setting, "nz5")) %>%
  mutate(
    cor = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "covid_(.*)-", group = 1),
    p = factor(
      str_extract(setting, "dim([0-9]+)", group = 1),
      levels = c('500', "1000")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(cor == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = beta_2, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9)) +
  geom_hline(data = df_true, aes(yintercept = y_value, linetype = "True Parameter")) +
  facet_grid(`Correlated predictors` ~ p) +
  labs(y = "Estimated intercept") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        #strip.text.x = element_text(size = 20),
        #strip.text.y = element_text(size = 10),
        legend.title = element_blank(),
        text = element_text(size = 15),
        axis.text.y = element_text(size = 10),
        legend.text = element_text(size = 10)) +
  scale_y_continuous(labels = scales::label_number(accuracy=.1), limits = c(1,3),
                     sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels =
  scale_x_continuous(sec.axis = sec_axis(~ . , name = "Fixed effect dimension", breaks = NULL, labels =
  scale_color_brewer(type = "qual",
                     labels = c("Estimates with LASSO",
                                "Estimates with SCAD"))


ggsave(
  "../plots/penal_nz5.pdf",
  penal,
  width = 20,
  height = 15,
  units = "cm"
)
```

LASSO is biased downward!

How do we do at estimating the variance component?

```
df_true <- data.frame(y_value = .56)

var_comp <- best_gene_results %>%
  filter(str_detect(setting, "nz5")) %>%
  mutate(
    cor = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "covid_(.*)-", group = 1),
    p = factor(
      str_extract(setting, "dim([0-9]+)", group = 1),
      levels = c('500', "1000")
    )
  ) %>%
```

```
  mutate(`Correlated predictors` = if_else(cor == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = psi_1, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9)) +
  geom_hline(data = df_true, aes(yintercept = y_value, linetype = "True Parameter")) +
  facet_grid(`Correlated predictors` ~ p) +
  labs(y = "Estimated intercept") +
  theme(axis.text.x = element_blank(),
        axis.ticks.x = element_blank(),
        #strip.text.x = element_text(size = 20),
        #strip.text.y = element_text(size = 10),
        legend.title = element_blank(),
        text = element_text(size = 15),
        axis.text.y = element_text(size = 10),
        legend.text = element_text(size = 10)) +
  scale_y_continuous(labels = scales::label_number(accuracy=.1),
                     sec.axis =
                       sec_axis(~ . , name = "Correlated predictors?",
                                breaks = NULL, labels = NULL)) +
  scale_x_continuous(sec.axis =
                       sec_axis(~ . , name = "Fixed effect dimension",
                                breaks = NULL, labels = NULL)) +
  scale_color_brewer(type = "qual",
                     labels = c("Estimates with LASSO",
                                "Estimates with SCAD"))

ggsave(
  "../plots/var_comp_nz5.pdf",
  var_comp,
  width = 20,
  height = 15,
  units = "cm"
)
```

All estimates are biased downward.

Plotting gene results for nz = 10

False positive rate

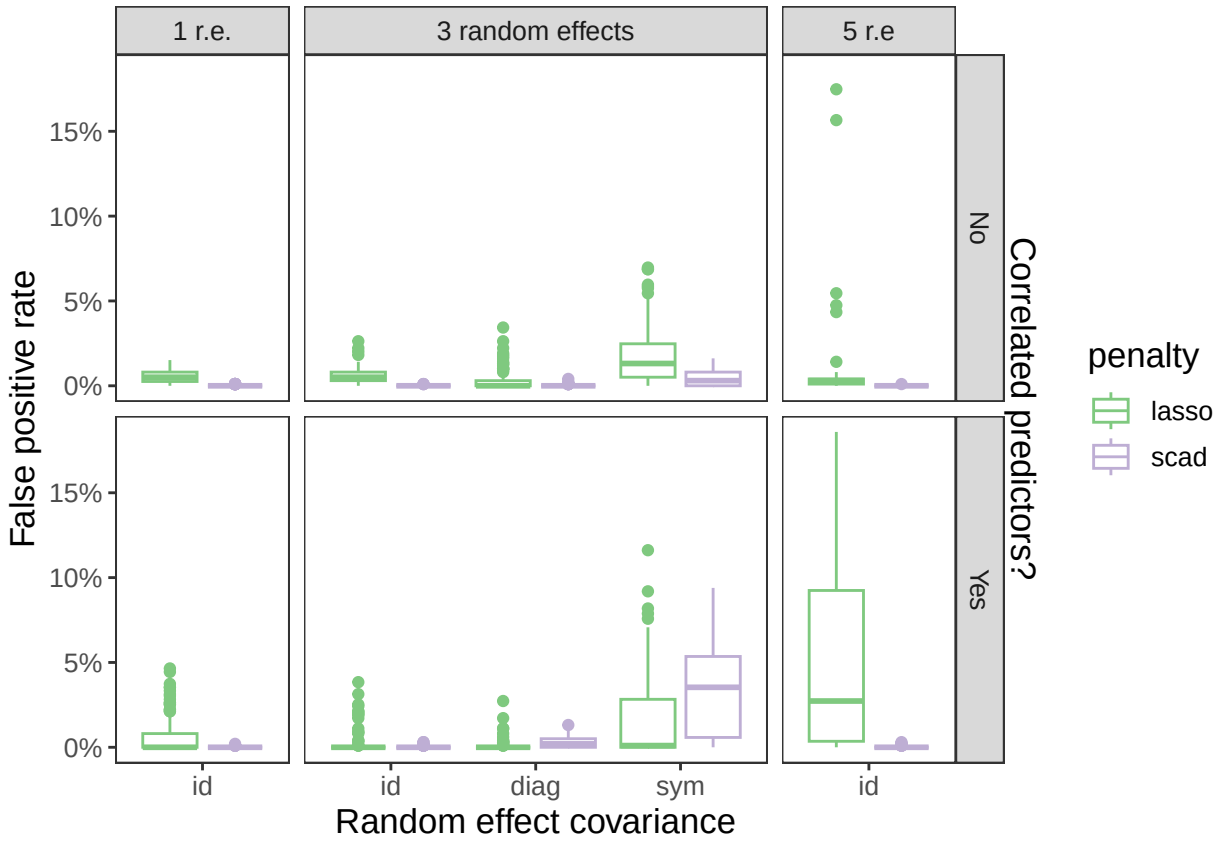
```
q.labs = c("1 r.e.", "3 random effects", "5 r.e")
names(q.labs) <- c("1", "3", "5")


fp_nz10 <- best_gene_results %>%
  filter(str_detect(setting, "nz10")) %>%
  mutate(
    cor = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "cov.*_(.*)-", group = 1),
    q = str_extract(setting, "random([0-9])", group = 1),
    cov_str = factor(
      str_extract(setting, "cov(.*)_", group = 1),
      levels = c("id", "diag", "sym")
    )
  ) %>%
```

```
  mutate(`Correlated predictors` = if_else(cor == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = (n_nz - tp)/990, x = cov_str, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9)) +
  labs(y = "False positive rate", x = "Random effect covariance") +
  facet_grid(
    `Correlated predictors` ~ q,
    scales = "free_x",
    space = "free",
    labeller = labeller(q = q.labs)
  ) + #Get rid of space argument if you want box plot width to automatically adjust to space in facet
  theme(#strip.text.x = element_text(size = 10 ),
        #strip.text.y = element_text(size = 10),
        text = element_text(size = 13),
        axis.text.y = element_text(size = 10)) +
  scale_y_continuous(labels = scales::percent,
                     sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels =
  scale_color_brewer(type = "qual")


# Note that there are fewer than 100 data-sets for some of the lasso settings (but none of the scad set
ggsave(
  "../plots/fp_nz10.pdf",
  fp_nz10,
  width = 20,
  height = 15,
  units = "cm"
)

fp_nz10
```

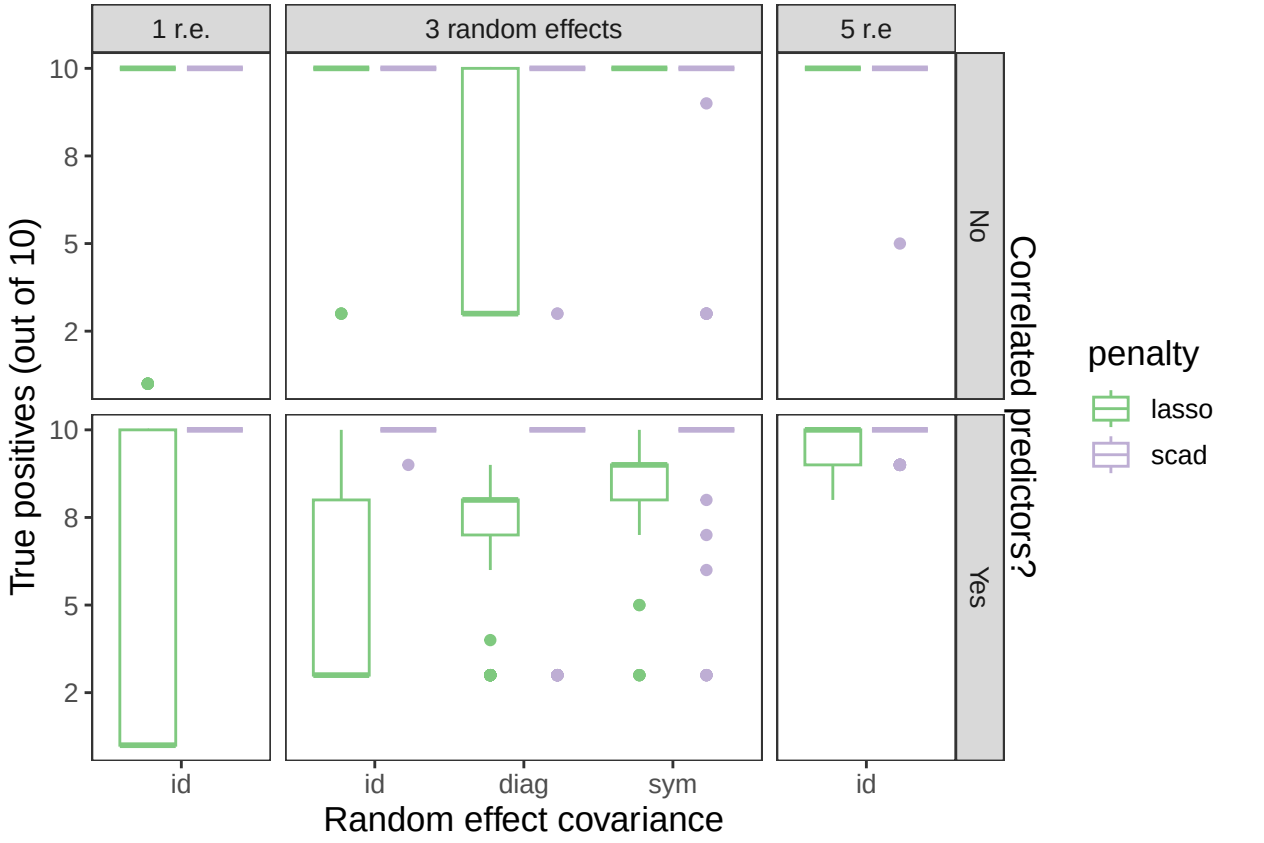


True positives

```
tp_nz10 <- best_gene_results %>%
  filter(str_detect(setting, "nz10")) %>%
  mutate(
    correlation = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "cov.*_(.*)-", group = 1),
    q = str_extract(setting, "random([0-9])", group = 1),
    cov_str = factor(
      str_extract(setting, "cov(.*)_", group = 1),
      levels = c("id", "diag", "sym")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(correlation == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = tp, x = cov_str, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9))  +
  labs(y = "True positives (out of 10)", x = "Random effect covariance") +
  facet_grid(
    `Correlated predictors` ~ q,
    labeller = labeller(q = q.labs),
    scales = "free_x",
    space = "free"
  ) + #Get rid of space argument if you want box plot width to automatically adjust to space in facet
    scale_y_continuous(labels = scales::label_number(accuracy=1),
                       sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels
    theme(#strip.text.x = element_text(size = 10 ),
```

```
      #strip.text.y = element_text(size = 10),
      text = element_text(size = 13),
      axis.text.y = element_text(size = 10)) +
  scale_color_brewer(type = "qual")

tp_nz10
```




```
ggsave(
  "../plots/tp_nz10.pdf",
  tp_nz10,
  width = 20,
  height = 15,
  units = "cm"
)
```

Combining false postive and true positive plots

```
full_plot = fp_nz10 / tp_nz10 + plot_layout(guides = "collect") & theme(legend.position = 'right')

ggsave("../plots/fullplot_nz10.pdf",
       full_plot,
       width = 6,
       height = 6,
       units = "in"
       )
```

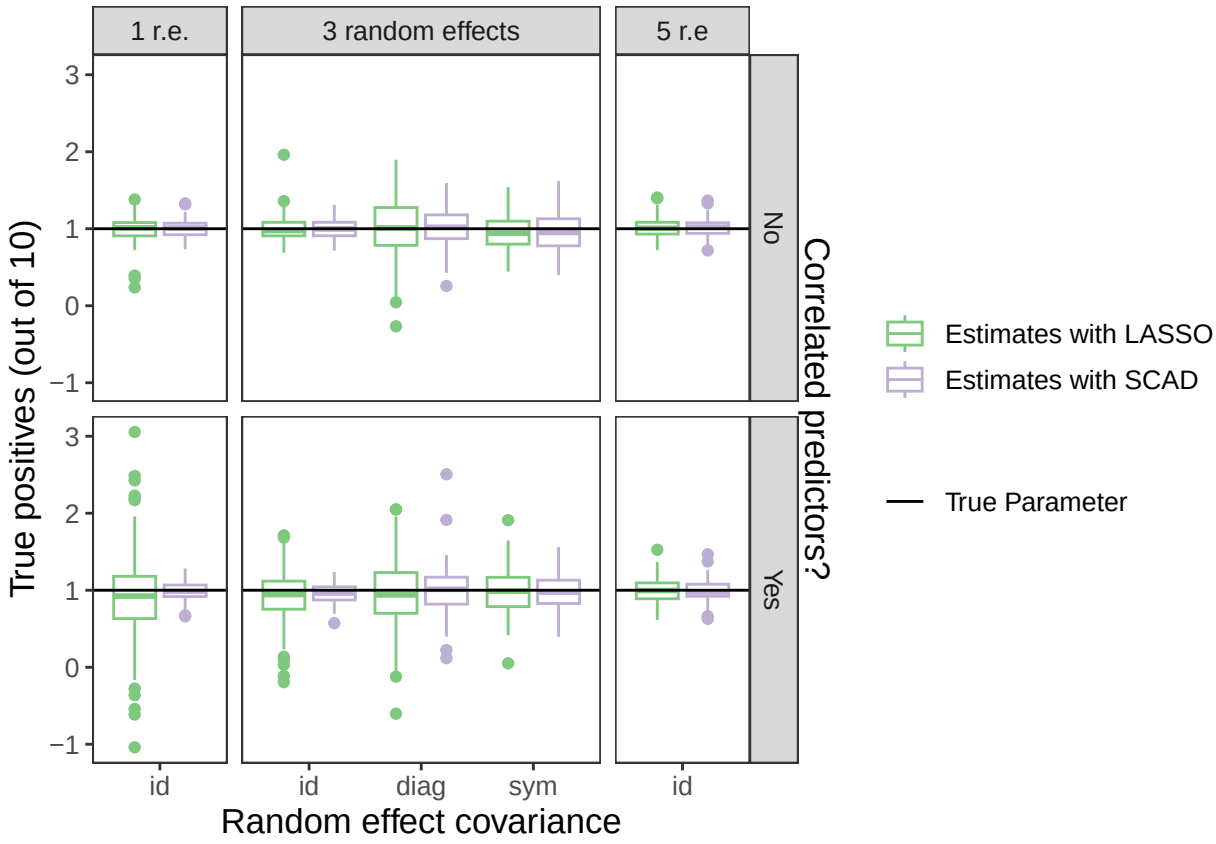
Parameter estimates

How do we do at estimating the unpenalized intercept

```
df_true <- data.frame(y_value = 1)

unp_nz10 <- best_gene_results %>%
  filter(str_detect(setting, "nz10")) %>%
  mutate(
    correlation = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "cov.*_(.*)-", group = 1),
    q = str_extract(setting, "random([0-9])", group = 1),
    cov_str = factor(
      str_extract(setting, "cov(.*)_", group = 1),
      levels = c("id", "diag", "sym")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(correlation == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = beta_1, x = cov_str, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9))  +
  geom_hline(data = df_true, aes(yintercept = y_value, linetype = "True Parameter")) +
  labs(y = "True positives (out of 10)", x = "Random effect covariance") +
  facet_grid(
    `Correlated predictors` ~ q,
    labeller = labeller(q = q.labs),
    scales = "free_x",
    space = "free"
  ) + #Get rid of space argument if you want box plot width to automatically adjust to space in facet
    scale_y_continuous(labels = scales::label_number(accuracy=1),
                       sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels
    theme(#strip.text.x = element_text(size = 10 ),
      #strip.text.y = element_text(size = 10),
      legend.title = element_blank(),
      text = element_text(size = 13),
      axis.text.y = element_text(size = 10)) +
  scale_color_brewer(type = "qual",
                     labels = c("Estimates with LASSO",
                                "Estimates with SCAD"))

unp_nz10
```



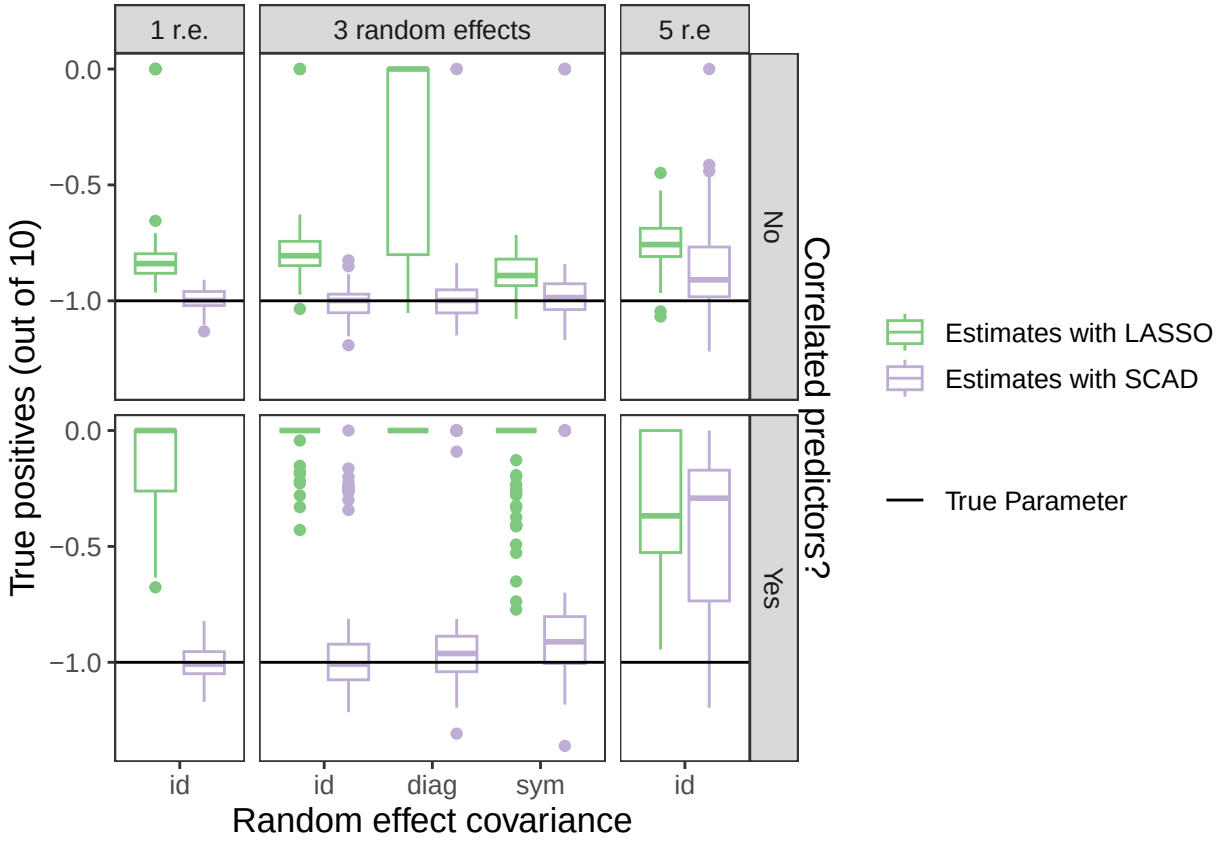

```
ggsave(
  "../plots/unp_nz10.pdf",
  unp_nz10,
  width = 20,
  height = 15,
  units = "cm"
)
```

```
df_true <- data.frame(y_value = -1)

penal_nz10 <- best_gene_results %>%
  filter(str_detect(setting, "nz10")) %>%
  mutate(
    correlation = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "cov.*_(.*)-", group = 1),
    q = str_extract(setting, "random([0-9])", group = 1),
    cov_str = factor(
      str_extract(setting, "cov(.*)_", group = 1),
      levels = c("id", "diag", "sym")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(correlation == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = beta_6, x = cov_str, col = penalty)) +
  geom_boxplot(position = position_dodge(width = 0.9))  +
  geom_hline(data = df_true, aes(yintercept = y_value, linetype = "True Parameter")) +
  labs(y = "True positives (out of 10)", x = "Random effect covariance") +
  facet_grid(
```

```
    `Correlated predictors` ~ q,
    labeller = labeller(q = q.labs),
    scales = "free_x",
    space = "free"
  ) + #Get rid of space argument if you want box plot width to automatically adjust to space in facet
    scale_y_continuous(labels = scales::label_number(accuracy=.1),
                       sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels
    theme(#strip.text.x = element_text(size = 10 ),
      #strip.text.y = element_text(size = 10),
      legend.title = element_blank(),
      text = element_text(size = 13),
      axis.text.y = element_text(size = 10)) +
  scale_color_brewer(type = "qual",
                     labels = c("Estimates with LASSO",
                                "Estimates with SCAD"))
```

```
penal_nz10
```




```
ggsave(
  "../plots/penal_nz10.pdf",
  penal_nz10,
  width = 20,
  height = 15,
  units = "cm"
)
```

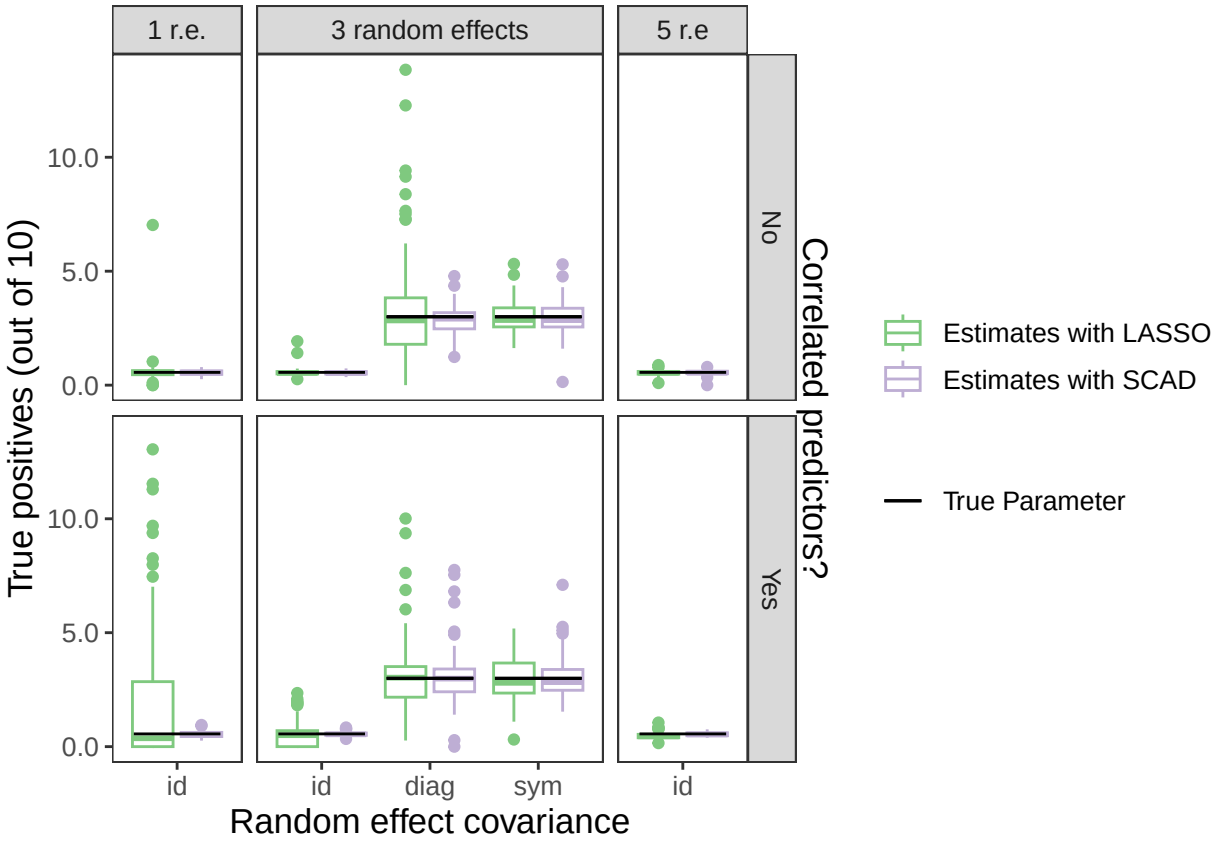
```r
df_true1 <- data.frame(cov_str = c(1, 2, 3), y_value = c(.56, 3, 3), q = c(1, 3, 3))
df_true2 <- data.frame(cov_str = c(1, 2, 3), y_value = c(.56, 3, 3), q = c(3, 3, 3))
df_true3 <- data.frame(cov_str = c(1, 2, 3), y_value = c(.56, 3, 3), q = c(5, 3, 3))

var_comp_nz10 <- best_gene_results %>%
  filter(str_detect(setting, "nz10")) %>%
  mutate(
    correlation = str_extract(setting, "rho([0-9]+\\.[0-9]+)", group = 1),
    penalty = str_extract(setting, "cov.*_(.*)-", group = 1),
    q = str_extract(setting, "random([0-9])", group = 1),
    cov_str = factor(
      str_extract(setting, "cov(.*)_", group = 1),
      levels = c("id", "diag", "sym")
    )
  ) %>%
  mutate(`Correlated predictors` = if_else(correlation == "0.0", "No", "Yes")) %>%
  ggplot(aes(y = psi_1, x = cov_str)) +
  geom_boxplot(position = position_dodge(width = 0.9), aes(col = penalty))  +
  geom_segment(data = df_true1, aes(x = cov_str-.4, xend = cov_str+.4,
                                    y = y_value, yend = y_value,
                                    linetype = "True Parameter")) +
  geom_segment(data = df_true2, aes(x = cov_str-.4, xend = cov_str+.4,
                                    y = y_value, yend = y_value,
                                    linetype = "True Parameter")) +
  geom_segment(data = df_true3, aes(x = cov_str-.4, xend = cov_str+.4,
                                    y = y_value, yend = y_value,
                                    linetype = "True Parameter")) +
  labs(y = "True positives (out of 10)", x = "Random effect covariance") +
  facet_grid(
    `Correlated predictors` ~ q,
    labeller = labeller(q = q.labs),
    scales = "free_x",
    space = "free"
  ) + #Get rid of space argument if you want box plot width to automatically adjust to space in facet
    scale_y_continuous(labels = scales::label_number(accuracy=.1),
                       sec.axis = sec_axis(~ . , name = "Correlated predictors?", breaks = NULL, labels
    theme(#strip.text.x = element_text(size = 10 ),
      #strip.text.y = element_text(size = 10),
      legend.title = element_blank(),
      text = element_text(size = 13),
      axis.text.y = element_text(size = 10)) +
  scale_color_brewer(type = "qual",
                     labels = c("Estimates with LASSO",
                                "Estimates with SCAD"))

var_comp_nz10
```

```
## Warning: Combining variables of class <character> and <numeric> was deprecated in
## ggplot2 3.4.0.
## i Please ensure your variables are compatible before plotting (location:
##   `combine_vars()`)
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```

```
## Warning: Combining variables of class <numeric> and <character> was deprecated in
## ggplot2 3.4.0.
## i Please ensure your variables are compatible before plotting (location:
##   `combine_vars()`)
## This warning is displayed once every 8 hours.
## Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
## generated.
```




```
ggsave(
  "../plots/var_comp_nz10.pdf",
  var_comp_nz10,
  width = 20,
  height = 15,
  units = "cm"
)
```

GWAS Results