

Transporter Colocalization Analysis (ImageJ Coloc2)

CT Berezin

2022-12-01

M1 = occludin overlapping P-gp/Bcrp/Mrp2 M2 = P-gp/Bcrp/Mrp2 overlapping occludin

Reading in the data

P-gp

```
pgp_filelist <- list.files(path="../data/IHC/coloc2_results_pgp/",
                           pattern="*.txt", full.names = TRUE)
pgp_cols <- c("sample", "m1", "m2", "r_pearson", "Pval", "r_rand", "sd_rand")
pgp_coloc2 <- as.data.frame(matrix(1,length(pgp_cols)))
names(pgp_coloc2) <- pgp_cols

for (i in pgp_filelist) {
  test <- read_lines(i)
  sample <- str_subset(test, "Working on") %>%
    str_extract(., "(?<=:[:space:]).*")
  r_pearson <- str_subset(test, "no threshold") %>%
    str_extract(., "(?<=:[:space:]).*")
  m1 <- str_subset(test, "tM1") %>% str_extract(., "(?<=:[:space:]).*")
  m2 <- str_subset(test, "tM2") %>% str_extract(., "(?<=:[:space:]).*")
  Pval <- str_subset(test, "P-Value") %>% str_extract(., "(?<=:[:space:]).*")
  r_rand <- str_subset(test, "Costes Shuffled Mean") %>%
    str_extract(., "(?<=:[:space:]).*")
  sd_rand <- str_subset(test, "Costes Shuffled Std") %>%
    str_extract(., "(?<=:[:space:]).*")
  res <- c(all_of(sample), all_of(m1), all_of(m2), all_of(r_pearson),
          all_of(Pval), all_of(r_rand), all_of(sd_rand))
  pgp_coloc2 <- rbind(pgp_coloc2, res)
}
```

```
## Warning: Using 'all_of()' outside of a selecting function was deprecated in tidysselect
## 1.2.0.
## i See details at
## <https://tidysselect.r-lib.org/reference/faq-selection-context.html>
```

```
pgp_coloc2 <- pgp_coloc2 %>% na.omit(pgp_coloc2) %>%
  mutate(sex = as.factor(str_sub(sample, start=1L, end=1L)),
         mouseID = as.factor(str_extract(sample, "(?<=-)[:digit:]+")),
```

```

#prep = as.factor(str_extract(sample, "[:lower:]{2,}")),
prep = as.factor(str_extract(sample, "wm|section")),
image = str_extract(sample, "[:alpha:]{2,}.*"), .before=m1

pgp_coloc2$prep

## [1] section section section section section section wm      wm      wm
## [10] wm      wm      wm      wm      wm      wm      wm      wm      wm
## [19] wm      wm      wm      wm      wm      wm      section section section
## [28] section section section section section section section section section
## [37] section section section section section section section wm      wm      wm
## [46] wm      wm      wm      wm      wm      wm
## Levels: section wm

pgp_coloc2 <- pgp_coloc2 %>% select(-sample) %>%
  mutate(m1 = as.double(m1),
         m2 = as.double(m2),
         r_pearson = as.double(r_pearson),
         Pval = as.double(Pval),
         r_rand = as.double(r_rand),
         sd_rand = as.double(sd_rand),
         coloc = Pval>0.95)

#write.xlsx(pgp_coloc2, "../data/IHC/coloc2-pgp-totals.xlsx")

head(pgp_coloc2)

```

```

##   sex mouseID   prep          image    m1    m2 r_pearson Pval r_rand
## 2   F      20 section slide1-section2-20x-1 0.816 0.760    0.64    1    0
## 3   F      20 section slide1-section2-20x-2 0.789 0.770    0.58    1    0
## 4   F      20 section slide1-section3-20x-1 0.845 0.640    0.50    1    0
## 5   F      20 section slide1-section3-20x-2 0.063 0.057    0.61    1    0
## 6   F      20 section slide2-section2-20x-1 0.613 0.361    0.62    1    0
## 7   F      20 section slide2-section2-20x-2 0.778 0.772    0.62    1    0
##   sd_rand coloc
## 2      0  TRUE
## 3      0  TRUE
## 4      0  TRUE
## 5      0  TRUE
## 6      0  TRUE
## 7      0  TRUE

```

Bcrp

```

bcrp_filelist <- list.files(path="../data/IHC/coloc2_results_bcrp/",
                           pattern="*.txt", full.names = TRUE)
bcrp_cols <- c("sample", "m1", "m2", "r_pearson", "Pval", "r_rand", "sd_rand")
bcrp_coloc2 <- as.data.frame(matrix(,1,length(bcrp_cols)))
names(bcrp_coloc2) <- bcrp_cols

```

```

for (i in bcrp_filelist) {
  test <- read_lines(i)
  sample <- str_subset(test, "Working on") %>%
    str_extract(., "(?<=:[:space:]).*")
  r_pearson <- str_subset(test, "no threshold") %>%
    str_extract(., "(?<=:[:space:]).*")
  m1 <- str_subset(test, "tM1") %>% str_extract(., "(?<=:[:space:]).*")
  m2 <- str_subset(test, "tM2") %>% str_extract(., "(?<=:[:space:]).*")
  Pval <- str_subset(test, "P-Value") %>% str_extract(., "(?<=:[:space:]).*")
  r_rand <- str_subset(test, "Costes Shuffled Mean") %>%
    str_extract(., "(?<=:[:space:]).*")
  sd_rand <- str_subset(test, "Costes Shuffled Std") %>%
    str_extract(., "(?<=:[:space:]).*")
  print(sample)
  res <- c(all_of(sample), all_of(m1), all_of(m2), all_of(r_pearson),
    all_of(Pval), all_of(r_rand), all_of(sd_rand))
  bcrp_coloc2 <- rbind(bcrp_coloc2, res)
}

```

```

## [1] "F-22-section1-40x-1-avg4"
## [1] "F-22-section1-40x-2-avg4"
## [1] "F-22-section1-40x-3-avg4"
## [1] "F-22-section3-40x-1-avg4"
## [1] "F-22-section3-40x-2-avg4"
## [1] "F-22-section3-40x-3-avg4"
## [1] "F-GTL-section1-40x-1-avg4"
## [1] "F-GTL-section1-40x-2-avg4"
## [1] "F-GTL-section1-40x-3-avg4"
## [1] "F-GTL-section3-40x-1-avg4"
## [1] "F-GTL-section3-40x-2-avg4"
## [1] "F-GTL-section3-40x-3-avg4"
## [1] "F-McK0-98-section1-40x-1-avg4"
## [1] "F-McK0-98-section1-40x-2-avg4"
## [1] "F-McK0-98-section1-40x-3-avg4"
## [1] "F-McK0-98-section3-40x-1-avg4"
## [1] "F-McK0-98-section3-40x-2-avg4"
## [1] "F-McK0-98-section3-40x-3-avg4"
## [1] "M-80-section1-40x-1-avg4"
## [1] "M-80-section1-40x-2-avg4"
## [1] "M-80-section1-40x-3-avg4"
## [1] "M-80-section3-40x-1-avg4"
## [1] "M-80-section3-40x-2-avg4"
## [1] "M-80-section3-40x-3-avg4"
## [1] "M-81-section1-40x-1-avg4"
## [1] "M-81-section1-40x-2-avg4"
## [1] "M-81-section1-40x-3-avg4"
## [1] "M-81-section3-40x-1-avg4"
## [1] "M-81-section3-40x-2-avg4"
## [1] "M-81-section3-40x-3-avg4"
## [1] "M-82-section1-40x-1-avg4"
## [1] "M-82-section1-40x-2-avg4"
## [1] "M-82-section1-40x-3-avg4"
## [1] "M-82-section3-40x-1-avg4"

```

```
## [1] "M-82-section3-40x-2-avg4"
## [1] "M-82-section3-40x-3-avg4"
## [1] "M-McK0-92-section1-40x-1-avg4"
## [1] "M-McK0-92-section1-40x-2-avg4"
## [1] "M-McK0-92-section1-40x-3-avg4"
## [1] "M-McK0-92-section3-40x-1-avg4"
## [1] "M-McK0-92-section3-40x-2-avg4"
## [1] "M-McK0-92-section3-40x-3-avg4"
```

```
bcrp_coloc2 <- bcrp_coloc2 %>% na.omit(bcrp_coloc2) %>%
  mutate(sex = as.factor(str_sub(sample, start=1L, end=1L)),
         mouseID = as.factor(str_extract(sample, "(?<=)[:digit:]+")),
         prep = as.factor(str_extract(sample, "[:lower:]{2,}")),
         image = str_extract(sample, "[:alpha:]{2,}.*"), .before=m1)

bcrp_coloc2 <- bcrp_coloc2 %>% select(-sample) %>%
  mutate(m1 = as.double(m1),
         m2 = as.double(m2),
         r_pearson = as.double(r_pearson),
         Pval = as.double(Pval),
         r_rand = as.double(r_rand),
         sd_rand = as.double(sd_rand),
         coloc = Pval>0.95)

write.xlsx(bcrp_coloc2, "../data/IHC/coloc2-bcrp-totals.xlsx")

head(bcrp_coloc2)
```

```
##   sex mouseID   prep          image    m1    m2 r_pearson Pval r_rand
## 2   F      22 section section1-40x-1-avg4 0.618 0.229    0.32    1     0
## 3   F      22 section section1-40x-2-avg4 0.787 0.359    0.63    1     0
## 4   F      22 section section1-40x-3-avg4 0.782 0.290    0.51    1     0
## 5   F      22 section section3-40x-1-avg4 0.517 0.461    0.49    1     0
## 6   F      22 section section3-40x-2-avg4 0.687 0.541    0.58    1     0
## 7   F      22 section section3-40x-3-avg4 0.555 0.431    0.58    1     0
##   sd_rand coloc
## 2      0  TRUE
## 3      0  TRUE
## 4      0  TRUE
## 5      0  TRUE
## 6      0  TRUE
## 7      0  TRUE
```

Mrp2

```
mrp2_filelist <- list.files(path="../data/IHC/coloc2_results_mrp2/",
                           pattern="*.txt", full.names = TRUE)
mrp2_cols <- c("sample", "m1", "m2", "r_pearson", "Pval", "r_rand", "sd_rand")
mrp2_coloc2 <- as.data.frame(matrix(,1,length(mrp2_cols)))
names(mrp2_coloc2) <- mrp2_cols
```

```

for (i in mrp2_filelist) {
  test <- read_lines(i)
  sample <- str_subset(test, "Working on") %>%
    str_extract(., "(?<=:[:space:]).*")
  r_pearson <- str_subset(test, "no threshold") %>%
    str_extract(., "(?<=:[:space:]).*")
  m1 <- str_subset(test, "tM1") %>% str_extract(., "(?<=:[:space:]).*")
  m2 <- str_subset(test, "tM2") %>% str_extract(., "(?<=:[:space:]).*")
  Pval <- str_subset(test, "P-Value") %>% str_extract(., "(?<=:[:space:]).*")
  r_rand <- str_subset(test, "Costes Shuffled Mean") %>%
    str_extract(., "(?<=:[:space:]).*")
  sd_rand <- str_subset(test, "Costes Shuffled Std") %>%
    str_extract(., "(?<=:[:space:]).*")
  res <- c(all_of(sample), all_of(m1), all_of(m2), all_of(r_pearson),
    all_of(Pval), all_of(r_rand), all_of(sd_rand))
  mrp2_coloc2 <- rbind(mrp2_coloc2, res)
}

mrp2_coloc2 <- mrp2_coloc2 %>% na.omit(mrp2_coloc2) %>%
  mutate(sex = as.factor(str_sub(sample, start=1L, end=1L)),
    mouseID = as.factor(str_extract(sample, "(?<=--)[:digit:]+")),
    image = str_extract(sample, "s.*(?=_)"),
    roi = str_extract(sample, "(?<=_.)*"), .before=m1)

mrp2_coloc2 <- mrp2_coloc2 %>% select(-sample) %>%
  mutate(m1 = as.double(m1),
    m2 = as.double(m2),
    r_pearson = as.double(r_pearson),
    Pval = as.double(Pval),
    r_rand = as.double(r_rand),
    sd_rand = as.double(sd_rand),
    coloc = Pval>0.95)

#write.xlsx(mrp2_coloc2, "../data/IHC/coloc2-mrp2-totals.xlsx")

head(mrp2_coloc2)

```

```

##   sex mouseID      image roi    m1    m2 r_pearson Pval r_rand sd_rand
## 2   F      22 section2-40x-1 gcl1 0.063 0.360    0.12 1.00 -0.01 0.04
## 3   F      22 section2-40x-1 inl1 0.000 0.087   -0.01 0.34 0.00 0.02
## 4   F      22 section2-40x-1 inl2 0.000 0.097   -0.03 0.12 0.00 0.02
## 5   F      22 section2-40x-2 gcl1 0.111 0.001    0.01 0.77 0.00 0.01
## 6   F      22 section2-40x-2 inl1 0.000 0.183   -0.02 0.18 0.00 0.03
## 7   F      22 section2-40x-3 gcl1 0.000 0.181   -0.02 0.14 0.00 0.02
##   coloc
## 2  TRUE
## 3 FALSE
## 4 FALSE
## 5 FALSE
## 6 FALSE
## 7 FALSE

```

Generating summary statistics

P-gp

```
pgp_sumstats <- pgp_coloc2 %>% summarise(m1_mean = mean(m1),
                                         m1_sd = sd(m1),
                                         m2_mean = mean(m2),
                                         m2_sd = sd(m2),
                                         r_mean = mean(r_pearson),
                                         r_sd = sd(r_pearson),
                                         n_total = n(),
                                         n_coloc = sum(coloc)
                                         )

pgp_sumstats
```

```
##   m1_mean   m1_sd  m2_mean   m2_sd  r_mean   r_sd n_total n_coloc
## 1   0.298 0.3149236 0.2291373 0.2644154 0.324902 0.1636261     51     51
```

```
bcrp_sumstats <- bcrp_coloc2 %>% summarise(m1_mean = mean(m1),
                                           m1_sd = sd(m1),
                                           m2_mean = mean(m2),
                                           m2_sd = sd(m2),
                                           r_mean = mean(r_pearson),
                                           r_sd = sd(r_pearson),
                                           n_total = n(),
                                           n_coloc = sum(coloc)
                                           )

bcrp_sumstats
```

```
##   m1_mean   m1_sd  m2_mean   m2_sd  r_mean   r_sd n_total n_coloc
## 1 0.7374762 0.1322315 0.2660952 0.08701775 0.4604762 0.104437     42     42
```

```
mrp2_sumstats <- mrp2_coloc2 %>% summarise(m1_mean = mean(m1),
                                           m1_sd = sd(m1),
                                           m2_mean = mean(m2),
                                           m2_sd = sd(m2),
                                           r_mean = mean(r_pearson),
                                           r_sd = sd(r_pearson),
                                           n_total = n(),
                                           n_coloc = sum(coloc)
                                           )

mrp2_sumstats
```

```
##   m1_mean   m1_sd  m2_mean   m2_sd  r_mean   r_sd n_total n_coloc
## 1 0.1098828 0.1436731 0.2081172 0.1569569 0.03496552 0.07127923    145     55
```

```
mrp2_sumstats_image <- mrp2_coloc2 %>% group_by(image,mouseID,sex,coloc) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
```

```

    m2_sd = sd(m2),
    r_mean = mean(r_pearson),
    r_sd = sd(r_pearson),
    n_total = n(),
    n_coloc = sum(coloc)
  )

```

'summarise()' has grouped output by 'image', 'mouseID', 'sex'. You can override
using the '.groups' argument.

```
mrp2_sumstats_image
```

```

## # A tibble: 59 x 12
## # Groups:   image, mouseID, sex [40]
##   image  mouseID sex  coloc m1_mean  m1_sd m2_mean  m2_sd  r_mean  r_sd
##   <chr>   <fct> <fct> <lgl>   <dbl>  <dbl>  <dbl>   <dbl>  <dbl>  <dbl>
## 1 section~ 40    F    FALSE 0.0527 0.0352 0.0933 0.107 0.0167 0.00577
## 2 section~ 92    M    FALSE 0      0      0.204 0.0566 -0.055 0.0778
## 3 section~ 40    F    FALSE 0.155  NA     0.023  NA     0.02  NA
## 4 section~ 40    F    TRUE  0.127 0.0396 0.129 0.0465 0.0533 0.00577
## 5 section~ 92    M    FALSE 0.019 0.0329 0.213 0.0721 0      0.0346
## 6 section~ 40    F    FALSE 0      NA     0.079  NA     0      NA
## 7 section~ 40    F    TRUE  0.0465 0.0573 0.102 0.0983 0.025 0.00707
## 8 section~ 92    M    FALSE 0      NA     0.352  NA    -0.02  NA
## 9 section~ 92    M    TRUE  0.308 0.0629 0.258 0.0643 0.11  0.0283
## 10 section~ 40   F    FALSE 0.013  NA     0.362  NA     0.02  NA
## # ... with 49 more rows, and 2 more variables: n_total <int>, n_coloc <int>

```

```

pgp_sumstats_animals <- pgp_coloc2 %>% group_by(mouseID,sex,prep) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc)
  )

```

'summarise()' has grouped output by 'mouseID', 'sex'. You can override using
the '.groups' argument.

```
pgp_sumstats_animals
```

```

## # A tibble: 10 x 11
## # Groups:   mouseID, sex [10]
##   mouseID sex  prep  m1_mean  m1_sd m2_mean  m2_sd r_mean  r_sd n_total
##   <fct>   <fct> <fct>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <int>
## 1 10      M    wm      0.01  0.00784 0.043 0.0168 0.128 0.0295      5
## 2 11      M    wm      0.0258 0.0409 0.0503 0.0192 0.145 0.0191      4
## 3 20      F    section 0.651 0.299 0.56 0.293 0.595 0.0505      6

```

```
## 4 39      F      wm      0.227 0.258      0.0782 0.135      0.375 0.0947      4
## 5 80      M      section 0.320 0.282      0.246 0.290      0.37 0.140      6
## 6 81      M      section 0.283 0.251      0.204 0.286      0.377 0.0838      6
## 7 82      M      section 0.381 0.325      0.208 0.270      0.352 0.183      6
## 8 83      M      wm      0.430 0.373      0.368 0.274      0.328 0.0804      5
## 9 87      F      wm      0.416 0.366      0.297 0.247      0.3 0.0628      5
## 10 9      M      wm      0.0265 0.0155      0.0812 0.0538      0.138 0.0419      4
## # ... with 1 more variable: n_coloc <int>
```

```
bcrp_sumstats_animals <- bcrp_coloc2 %>% group_by(mouseID,sex,prep) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc)
  )
```

'summarise()' has grouped output by 'mouseID', 'sex'. You can override using
the '.groups' argument.

```
bcrp_sumstats_animals
```

```
## # A tibble: 7 x 11
## # Groups:   mouseID, sex [7]
##   mouseID sex prep m1_mean m1_sd m2_mean m2_sd r_mean r_sd n_total
##   <fct> <fct> <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>
## 1 22      F section 0.658 0.114 0.385 0.115 0.518 0.110 6
## 2 40      F section 0.774 0.0717 0.274 0.0454 0.502 0.117 6
## 3 80      M section 0.775 0.0587 0.162 0.0183 0.443 0.0638 6
## 4 81      M section 0.663 0.127 0.276 0.0642 0.42 0.103 6
## 5 82      M section 0.754 0.0375 0.275 0.0144 0.408 0.0337 6
## 6 92      M section 0.806 0.0483 0.264 0.0486 0.503 0.0403 6
## 7 98      F section 0.732 0.280 0.226 0.0874 0.428 0.177 6
## # ... with 1 more variable: n_coloc <int>
```

```
mrp2_sumstats_animals <- mrp2_coloc2 %>% group_by(mouseID,sex,coloc) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc)
  )
```

'summarise()' has grouped output by 'mouseID', 'sex'. You can override using
the '.groups' argument.


```
uneven <- mrp2_sumstats_animals %>% group_by(mouseID) %>%
  summarise(n = n(), coloc = coloc, sex=sex) %>%
  filter(n==1) %>% select(-n) %>%
  mutate(coloc = toString(as.integer(coloc))) %>%
  mutate(coloc = str_replace(toString(coloc), pattern = "0", replacement = "1")) %>%
  mutate(coloc = as.logical(as.integer(coloc)))
```

'summarise()' has grouped output by 'mouseID'. You can override using the
'.groups' argument.

```
mrp2_sumstats_animals <- rbind(mrp2_sumstats_animals, uneven) %>% arrange(., mouseID)
mrp2_sumstats_animals
```

```
## # A tibble: 16 x 11
## # Groups:   mouseID, sex [8]
##   mouseID sex  coloc m1_mean m1_sd m2_mean m2_sd r_mean r_sd
##   <fct>   <fct> <lgl>   <dbl> <dbl> <dbl> <dbl> <dbl> <dbl>
## 1 22      F     FALSE 0.0234 0.0497 0.202 0.118 -0.0262 0.0421
## 2 22      F     TRUE  0.04   0.0325 0.244 0.164 0.08   0.0566
## 3 40      F     FALSE 0.0543 0.0658 0.115 0.117 0.0122 0.0109
## 4 40      F     TRUE  0.0796 0.0552 0.136 0.0669 0.0486 0.0177
## 5 62      M     FALSE 0.0655 0.0959 0.0723 0.0747 -0.0174 0.0254
## 6 62      M     TRUE  NA     NA     NA     NA     NA     NA
## 7 77      M     FALSE 0.035  0.0492 0.135 0.0385 0.00933 0.0249
## 8 77      M     TRUE  0.242  0.142  0.200 0.0504 0.08   0.0486
## 9 81      M     FALSE 0.00646 0.00785 0.434 0.239 -0.00385 0.0112
## 10 81     M     TRUE  0.029  0.0428 0.420 0.291 0.044  0.0313
## 11 82     M     FALSE 0.0468 0.0604 0.143 0.0994 0.024  0.0397
## 12 82     M     TRUE  0.326  0.170  0.218 0.116 0.128  0.0669
## 13 92     M     FALSE 0.0406 0.0896 0.198 0.103 -0.0167 0.0424
## 14 92     M     TRUE  0.306  0.0802 0.306 0.0831 0.115  0.03
## 15 98     F     FALSE 0.056  0.0752 0.251 0.0982 0.005  0.0404
## 16 98     F     TRUE  0.253  0.124  0.262 0.0856 0.131  0.0752
## # ... with 2 more variables: n_total <int>, n_coloc <int>
```

```
pgp_sumstats_sex <- pgp_coloc2 %>% group_by(sex) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc)
  )
pgp_sumstats_sex
```

```
## # A tibble: 2 x 9
##   sex  m1_mean m1_sd m2_mean m2_sd r_mean r_sd n_total n_coloc
##   <fct> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int> <int>
## 1 F    0.459 0.341 0.344 0.305 0.438 0.150    15    15
## 2 M    0.231 0.282 0.181 0.234 0.278 0.147    36    36
```

```

bcrp_sumstats_sex <- bcrp_coloc2 %>% group_by(sex) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc)
            )
bcrp_sumstats_sex

```

```

## # A tibble: 2 x 9
##   sex    m1_mean m1_sd m2_mean m2_sd r_mean  r_sd n_total n_coloc
##   <fct>   <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl>   <int>   <int>
## 1 F      0.721 0.176   0.295 0.107  0.483 0.136     18     18
## 2 M      0.750 0.0896  0.244 0.0623  0.444 0.0719    24     24

```

```

mrp2_sumstats_sex <- mrp2_coloc2 %>% group_by(sex) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc)
            )
mrp2_sumstats_sex

```

```

## # A tibble: 2 x 9
##   sex    m1_mean m1_sd m2_mean m2_sd r_mean  r_sd n_total n_coloc
##   <fct>   <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl>   <int>   <int>
## 1 F      0.104 0.124   0.201 0.113 0.0411 0.0794     53     24
## 2 M      0.113 0.154   0.212 0.178 0.0314 0.0664     92     31

```

```

pgp_sumstats_prep <- pgp_coloc2 %>% group_by(prepare) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            mean_P = mean(Pval)
            )
pgp_sumstats_prep

```

```

## # A tibble: 2 x 9
##   prep    m1_mean m1_sd m2_mean m2_sd r_mean  r_sd n_total mean_P
##   <fct>   <dbl> <dbl>   <dbl> <dbl> <dbl> <dbl>   <int>   <dbl>
## 1 section 0.408 0.308   0.304 0.306  0.423 0.155     24     1
## 2 wm      0.200 0.292   0.162 0.205  0.237 0.116     27     1

```

```

mrp2_sumstats_coloc <- mrp2_coloc2 %>% group_by(coloc) %>%
  summarise(m1_mean = mean(m1),
            m1_sd = sd(m1),
            m2_mean = mean(m2),
            m2_sd = sd(m2),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            mean_P = mean(Pval)
            )
mrp2_sumstats_coloc

```

```

## # A tibble: 2 x 9
##   coloc m1_mean m1_sd m2_mean m2_sd r_mean r_sd n_total mean_P
##   <lgl>   <dbl> <dbl>   <dbl> <dbl>   <dbl> <dbl>   <int>   <dbl>
## 1 FALSE  0.0393 0.0672   0.187 0.166 -0.00622 0.0329    90  0.431
## 2 TRUE   0.225 0.160   0.243 0.136  0.102   0.0656    55  0.996

```

Statistical Tests

```

pgp_sumstats_animals$prep <- as.factor(pgp_sumstats_animals$prep)
coin::wilcox_test(pgp_sumstats_animals$r_mean ~ pgp_sumstats_animals$prep, conf.int=TRUE, distribution=

```

```

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sumstats_animals$r_mean by pgp_sumstats_animals$prep (section, wm)
## Z = 2.132, p-value = 0.0381
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  0.001666667 0.450000000
## sample estimates:
## difference in location
##                0.2218333

```

```

coin::wilcox_test(pgp_sumstats_animals$m1_mean ~ pgp_sumstats_animals$prep, conf.int=TRUE, distribution=

```

```

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sumstats_animals$m1_mean by pgp_sumstats_animals$prep (section, wm)
## Z = 1.2792, p-value = 0.2571
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.1103000  0.6241667
## sample estimates:
## difference in location
##                0.256375

```

```
coin::wilcox_test(pgp_sumstats_animals$m2_mean ~ pgp_sumstats_animals$prep, conf.int=TRUE, distribution="
```

```
##  
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: pgp_sumstats_animals$m2_mean by pgp_sumstats_animals$prep (section, wm)  
## Z = 1.2792, p-value = 0.2571  
## alternative hypothesis: true mu is not equal to 0  
## 95 percent confidence interval:  
## -0.1212667 0.4817500  
## sample estimates:  
## difference in location  
## 0.1592083
```

```
coin::wilcox_test(pgp_sumstats_animals$r_mean ~ pgp_sumstats_animals$sex, conf.int=TRUE, distribution="
```

```
##  
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: pgp_sumstats_animals$r_mean by pgp_sumstats_animals$sex (F, M)  
## Z = 1.2536, p-value = 0.2667  
## alternative hypothesis: true mu is not equal to 0  
## 95 percent confidence interval:  
## -0.0700 0.4575  
## sample estimates:  
## difference in location  
## 0.172
```

```
coin::wilcox_test(pgp_sumstats_animals$m1_mean ~ pgp_sumstats_animals$sex, conf.int=TRUE, distribution=
```

```
##  
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: pgp_sumstats_animals$m1_mean by pgp_sumstats_animals$sex (F, M)  
## Z = 1.2536, p-value = 0.2667  
## alternative hypothesis: true mu is not equal to 0  
## 95 percent confidence interval:  
## -0.1537500 0.6249167  
## sample estimates:  
## difference in location  
## 0.21725
```

```
coin::wilcox_test(pgp_sumstats_animals$m2_mean ~ pgp_sumstats_animals$sex, conf.int=TRUE, distribution=
```

```
##  
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: pgp_sumstats_animals$m2_mean by pgp_sumstats_animals$sex (F, M)  
## Z = 1.0256, p-value = 0.3833  
## alternative hypothesis: true mu is not equal to 0  
## 95 percent confidence interval:
```

```
## -0.1680833 0.5097500
## sample estimates:
## difference in location
## 0.0932

pgp_wm_sumstats <- pgp_sumstats_animals %>% filter(prepare == "wm")

coin::wilcox_test(pgp_wm_sumstats$r_mean ~ pgp_wm_sumstats$sex, conf.int=TRUE, distribution="exact")

## Warning in cci(alpha): cannot compute confidence interval

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: pgp_wm_sumstats$r_mean by pgp_wm_sumstats$sex (F, M)
## Z = 1.3887, p-value = 0.2667
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## NA NA
## sample estimates:
## difference in location
## 0.16725

coin::wilcox_test(pgp_wm_sumstats$m1_mean ~ pgp_wm_sumstats$sex, conf.int=TRUE, distribution="exact")

## Warning in cci(alpha): cannot compute confidence interval

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: pgp_wm_sumstats$m1_mean by pgp_wm_sumstats$sex (F, M)
## Z = 0.92582, p-value = 0.5333
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## NA NA
## sample estimates:
## difference in location
## 0.209375

coin::wilcox_test(pgp_wm_sumstats$m2_mean ~ pgp_wm_sumstats$sex, conf.int=TRUE, distribution="exact")

## Warning in cci(alpha): cannot compute confidence interval

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: pgp_wm_sumstats$m2_mean by pgp_wm_sumstats$sex (F, M)
## Z = 0.46291, p-value = 0.8
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## NA NA
```

```
## sample estimates:
## difference in location
##          0.031625
```

```
pgp_sect_sumstats <- pgp_sumstats_animals %>% filter(prepare == "section")
```

```
coin::wilcox_test(pgp_sect_sumstats$r_mean ~ pgp_sect_sumstats$sex, conf.int=TRUE, distribution="exact")
```

```
## Warning in cci(alpha): cannot compute confidence interval
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sect_sumstats$r_mean by pgp_sect_sumstats$sex (F, M)
## Z = 1.3416, p-value = 0.5
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  NA NA
## sample estimates:
## difference in location
##          0.225
```

```
coin::wilcox_test(pgp_sect_sumstats$m1_mean ~ pgp_sect_sumstats$sex, conf.int=TRUE, distribution="exact")
```

```
## Warning in cci(alpha): cannot compute confidence interval
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sect_sumstats$m1_mean by pgp_sect_sumstats$sex (F, M)
## Z = 1.3416, p-value = 0.5
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  NA NA
## sample estimates:
## difference in location
##          0.3311667
```

```
coin::wilcox_test(pgp_sect_sumstats$m2_mean ~ pgp_sect_sumstats$sex, conf.int=TRUE, distribution="exact")
```

```
## Warning in cci(alpha): cannot compute confidence interval
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sect_sumstats$m2_mean by pgp_sect_sumstats$sex (F, M)
## Z = 1.3416, p-value = 0.5
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  NA NA
## sample estimates:
## difference in location
##          0.3523333
```

```
coin::wilcox_test(bcrp_sumstats_animals$r_mean ~ bcrp_sumstats_animals$sex, conf.int=TRUE, distribution="asymptotic")
```

```
## Warning in cci(alpha): cannot compute confidence interval
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: bcrp_sumstats_animals$r_mean by bcrp_sumstats_animals$sex (F, M)
## Z = 1.0607, p-value = 0.4
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## NA NA
## sample estimates:
## difference in location
## 0.03916667
```

```
coin::wilcox_test(bcrp_sumstats_animals$m1_mean ~ bcrp_sumstats_animals$sex, conf.int=TRUE, distribution="asymptotic")
```

```
## Warning in cci(alpha): cannot compute confidence interval
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: bcrp_sumstats_animals$m1_mean by bcrp_sumstats_animals$sex (F, M)
## Z = -1.0607, p-value = 0.4
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## NA NA
## sample estimates:
## difference in location
## -0.027
```

```
coin::wilcox_test(bcrp_sumstats_animals$m2_mean ~ bcrp_sumstats_animals$sex, conf.int=TRUE, distribution="asymptotic")
```

```
## Warning in cci(alpha): cannot compute confidence interval
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: bcrp_sumstats_animals$m2_mean by bcrp_sumstats_animals$sex (F, M)
## Z = 0.35355, p-value = 0.8571
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## NA NA
## sample estimates:
## difference in location
## 0.03708333
```

```

mrp2_sumstats_animals$coloc <- as.factor(mrp2_sumstats_animals$coloc)

mrp2_sumstats_animals$r_mean <- replace_na(mrp2_sumstats_animals$r_mean, 0)
mrp2_sumstats_animals$m1_mean <- replace_na(mrp2_sumstats_animals$m1_mean, 0)
mrp2_sumstats_animals$m2_mean <- replace_na(mrp2_sumstats_animals$m2_mean, 0)

coin::wilcoxsign_test(r_mean ~ coloc | mouseID, data=mrp2_sumstats_animals, conf.int=TRUE, distribution="normal")

##
## Exact Wilcoxon-Pratt Signed-Rank Test
##
## data: y by x (pos, neg)
## stratified by block
## Z = -2.5205, p-value = 0.007812
## alternative hypothesis: true mu is not equal to 0

coin::wilcoxsign_test(m1_mean ~ coloc | mouseID, data=mrp2_sumstats_animals, conf.int=TRUE, distribution="normal")

##
## Exact Wilcoxon-Pratt Signed-Rank Test
##
## data: y by x (pos, neg)
## stratified by block
## Z = -1.9604, p-value = 0.05469
## alternative hypothesis: true mu is not equal to 0

coin::wilcoxsign_test(m2_mean ~ coloc | mouseID, data=mrp2_sumstats_animals, conf.int=TRUE, distribution="normal")

##
## Exact Wilcoxon-Pratt Signed-Rank Test
##
## data: y by x (pos, neg)
## stratified by block
## Z = -1.4003, p-value = 0.1953
## alternative hypothesis: true mu is not equal to 0

coin::wilcox_test(mrp2_sumstats_animals$r_mean ~ mrp2_sumstats_animals$sex, conf.int=TRUE, distribution="normal")

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$r_mean by mrp2_sumstats_animals$sex (F, M)
## Z = 0.37991, p-value = 0.7333
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.06642857 0.07066667
## sample estimates:
## difference in location
## 0.004785714

```



```
coin::wilcox_test(mrp2_sumstats_animals$m1_mean ~ mrp2_sumstats_animals$sex, conf.int=TRUE, distribution="normal")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$m1_mean by mrp2_sumstats_animals$sex (F, M)
## Z = 0.3254, p-value = 0.7925
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.20228571 0.05057143
## sample estimates:
## difference in location
## 0.009990476
```

```
coin::wilcox_test(mrp2_sumstats_animals$m2_mean ~ mrp2_sumstats_animals$sex, conf.int=TRUE, distribution="normal")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$m2_mean by mrp2_sumstats_animals$sex (F, M)
## Z = 0.21693, p-value = 0.8749
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.1706786 0.1158833
## sample estimates:
## difference in location
## 0.003275794
```

Plotting the data

```
pgp_coloc2$coloc <- as.logical(pgp_coloc2$coloc)
pgp_coloc2$mouseID <- as.factor(pgp_coloc2$mouseID)

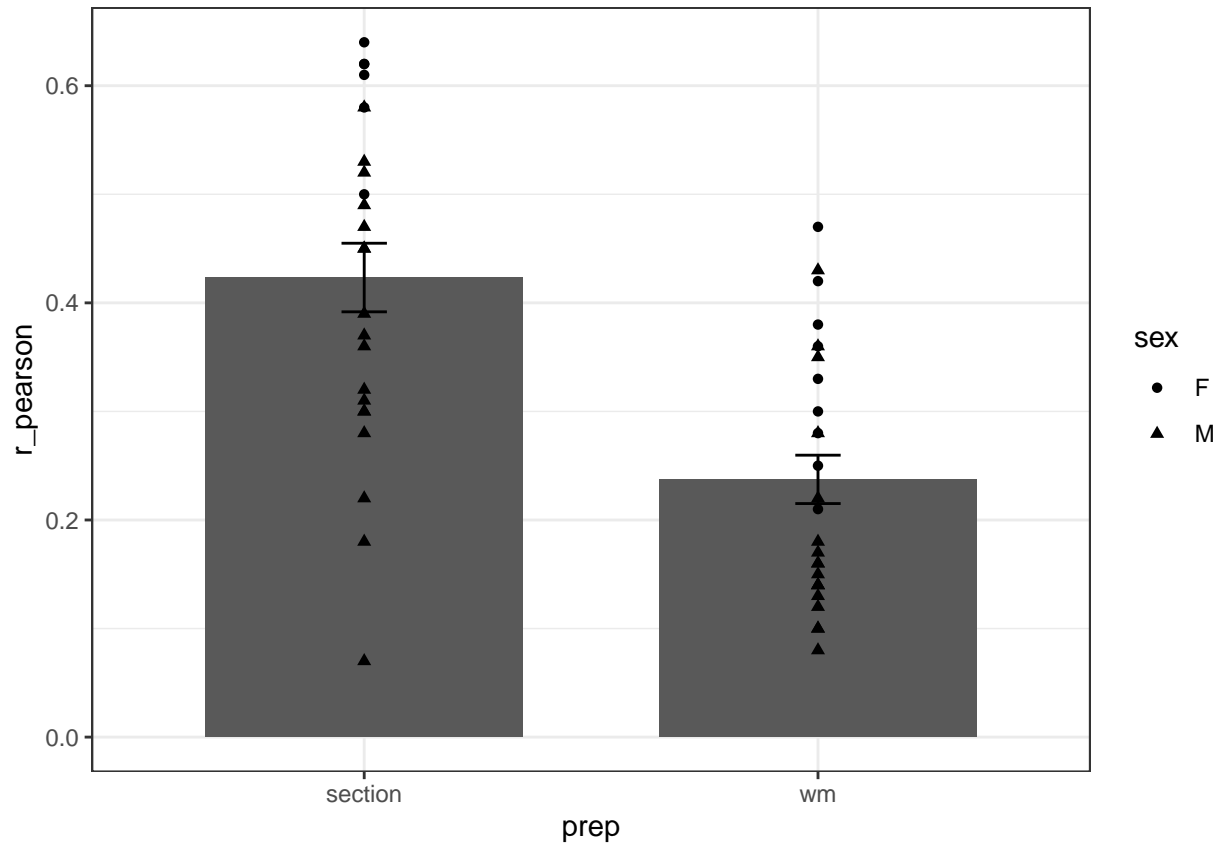
pgp_coloc2_long <- pgp_coloc2 %>% pivot_longer(cols = m1:m2, names_to = "m_stat_type", values_to = "m_stat_val")

pgp_coloc2_long <- pgp_coloc2_long %>% group_by(mouseID, m_stat_type, prep, sex) %>%
  summarise(m_mean = mean(m_stat_val),
            m_sd = sd(m_stat_val),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc))
```

```
## 'summarise()' has grouped output by 'mouseID', 'm_stat_type', 'prep'. You can
## override using the '.groups' argument.
```

```
#per image
pgp_coloc2 %>% ggplot(aes(x=prep, y=r_pearson)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
```

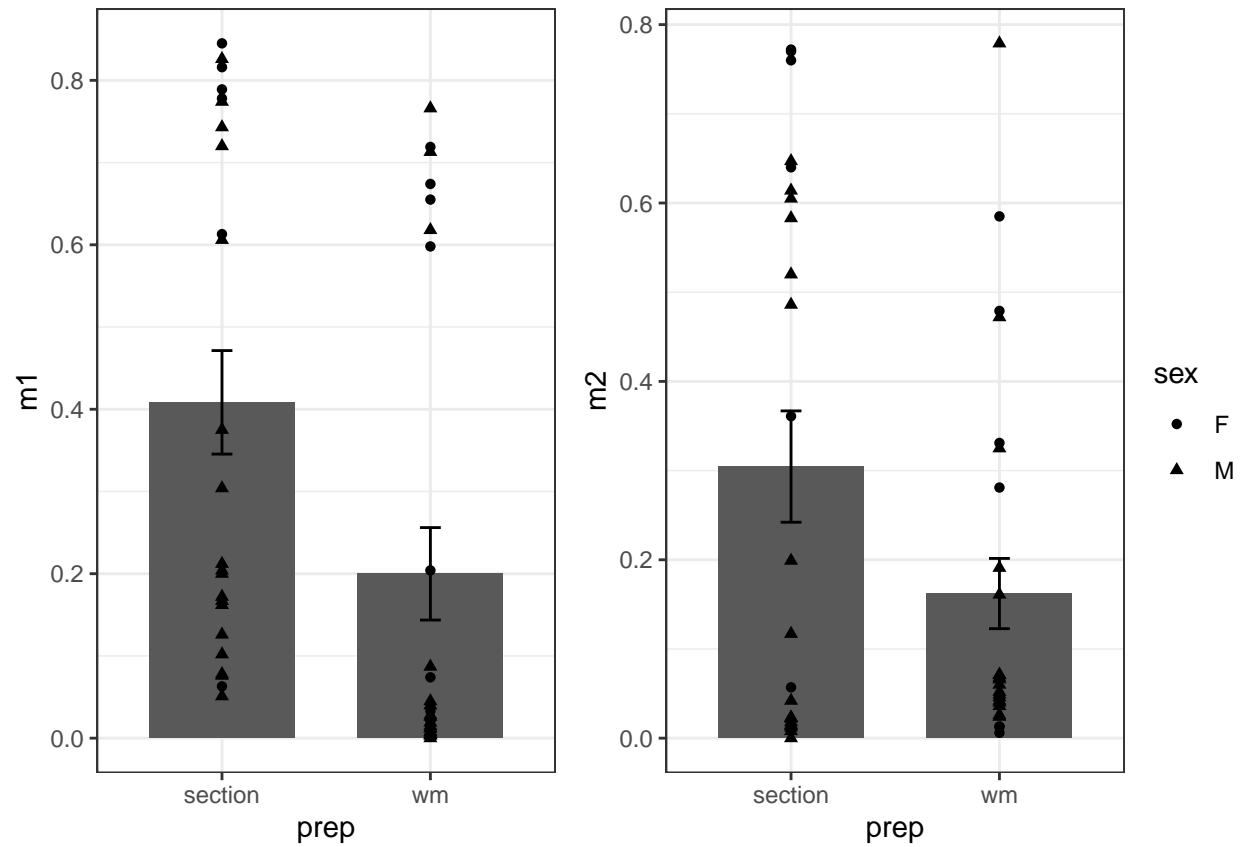
```
stat_summary(fun.data = mean_se,
             geom = "errorbar", width=0.1) +
geom_point(color="black", aes(shape=sex)) +
theme_bw()
```



```
pgp_m1_plot <- ggplot(pgp_coloc2, aes(x=prep, y=m1)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
             geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

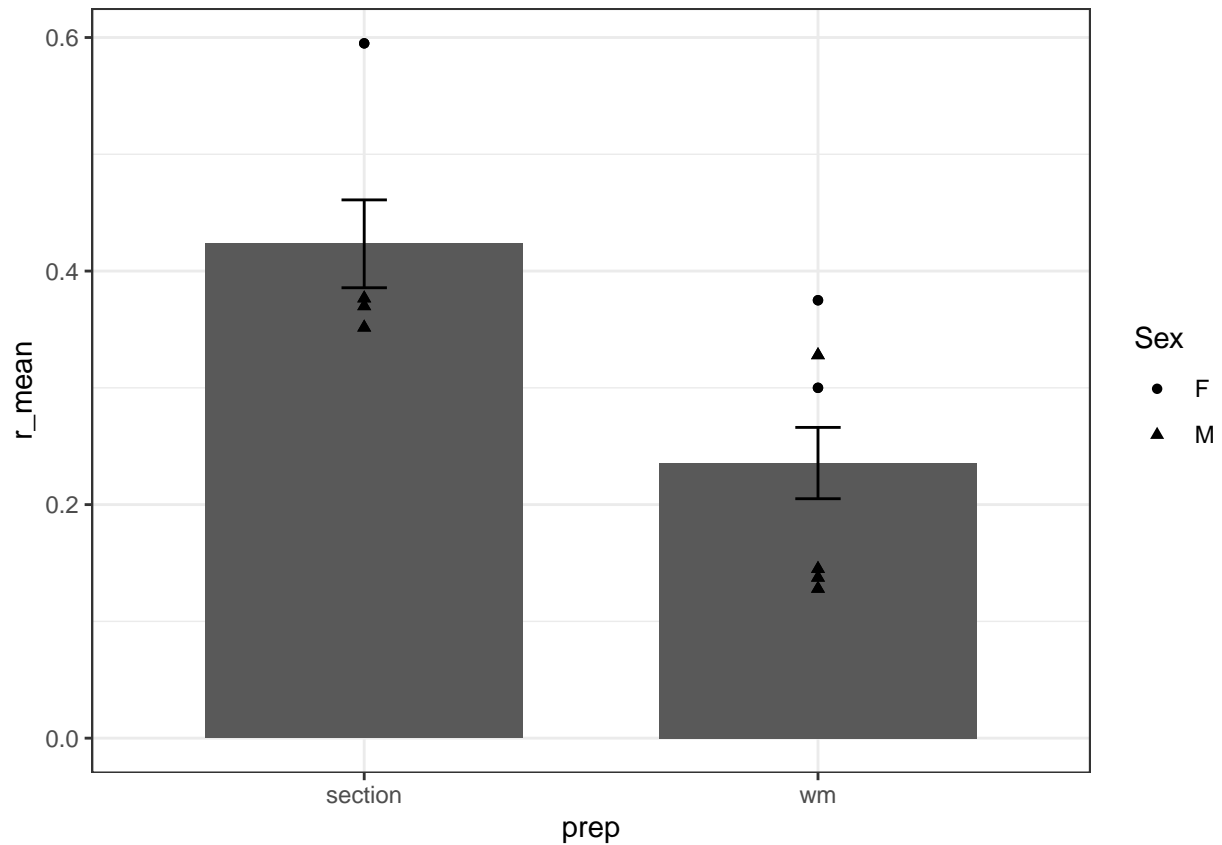
pgp_m2_plot <- ggplot(pgp_coloc2, aes(x=prep, y=m2)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
             geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

ggarrange(pgp_m1_plot, pgp_m2_plot, common.legend=TRUE, legend="right")
```



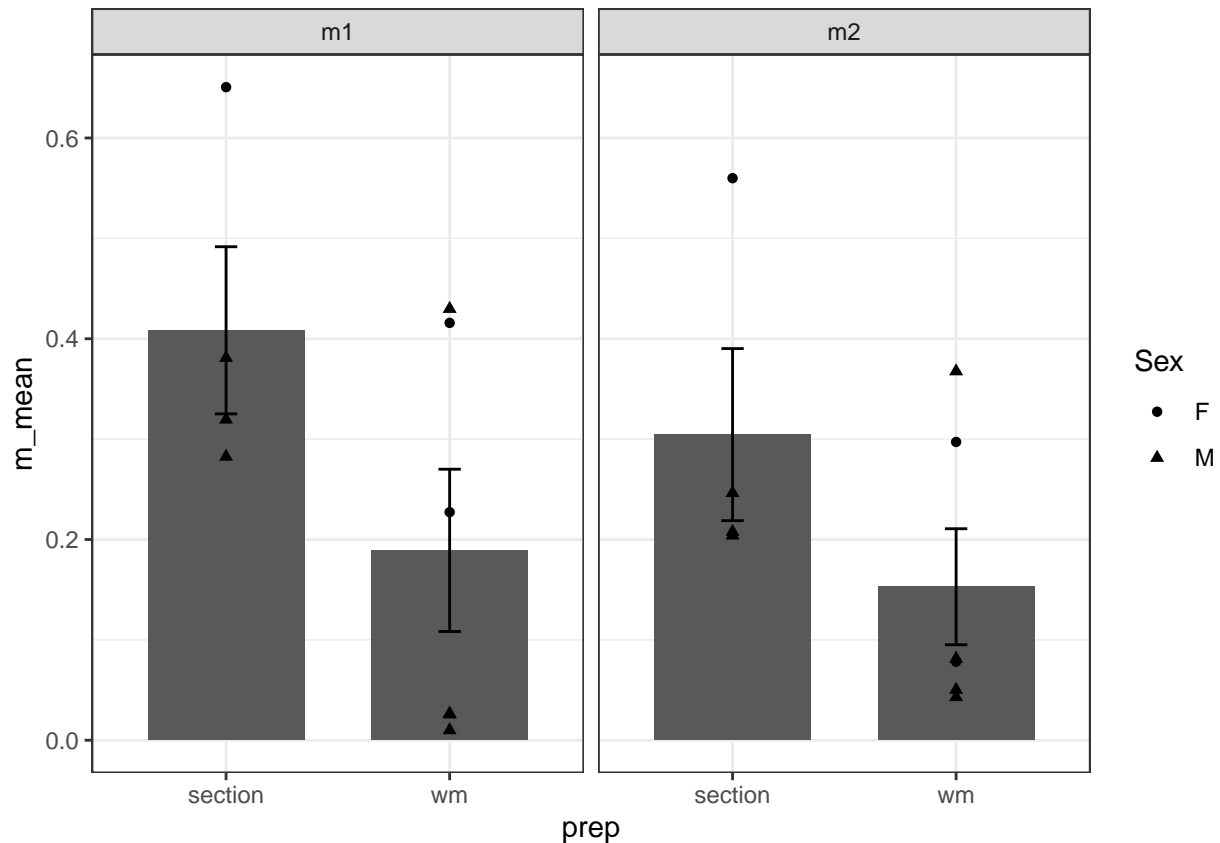
```
#per animal
pgp_animal_r_plot <- pgp_coloc2_long %>% ggplot(aes(x=prep, y=r_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw() +
  guides(shape=guide_legend("Sex"))

pgp_animal_r_plot
```



```
pgp_animal_m_plot <- ggplot(pgp_coloc2_long, aes(x=prep, y=m_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  facet_wrap(~m_stat_type) +
  theme_bw() +
  guides(shape=guide_legend("Sex"))

pgp_animal_m_plot
```

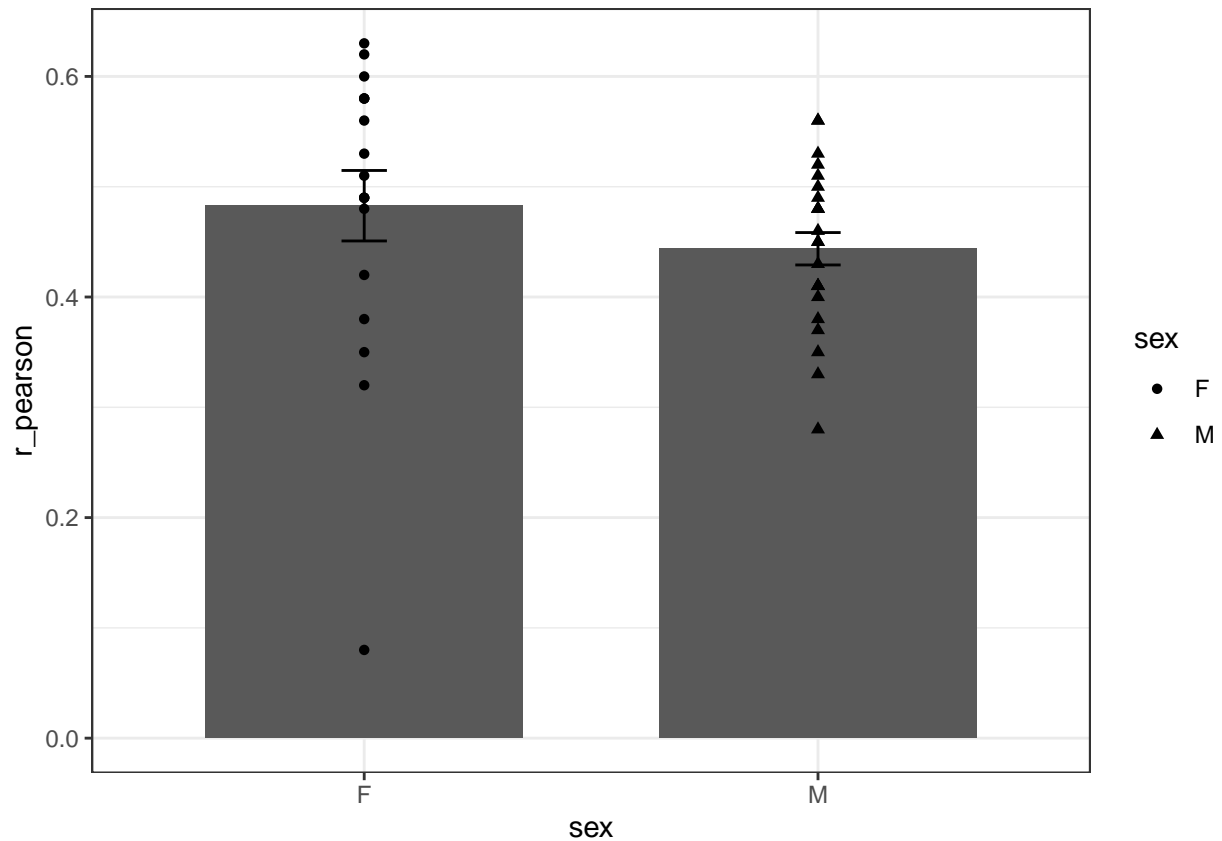


```
bcrp_coloc2_long <- bcrp_coloc2 %>% pivot_longer(cols = m1:m2, names_to = "m_stat_type", values_to = "m_stat_val")

bcrp_coloc2_long <- bcrp_coloc2_long %>% group_by(mouseID, m_stat_type, sex) %>%
  summarise(m_mean = mean(m_stat_val),
            m_sd = sd(m_stat_val),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc))
```

'summarise()' has grouped output by 'mouseID', 'm_stat_type'. You can override
using the '.groups' argument.

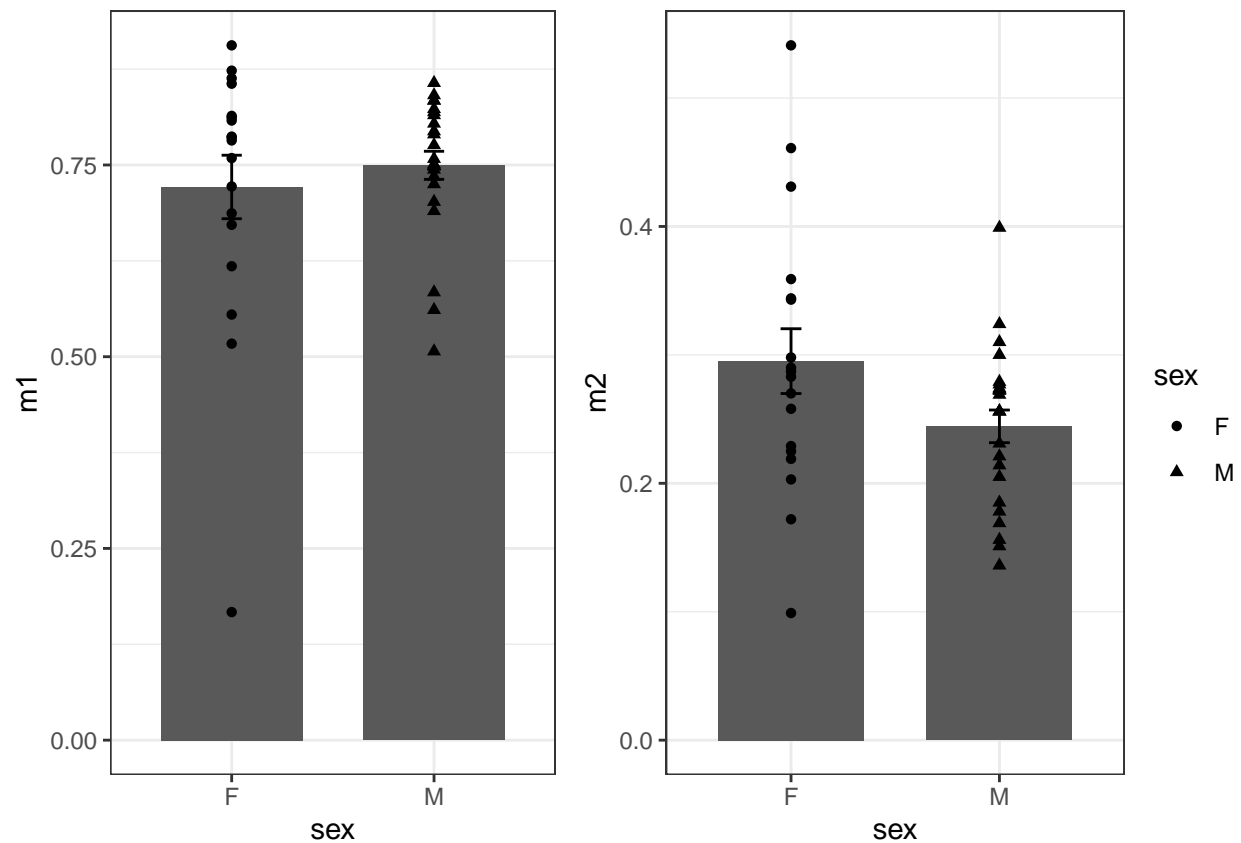
```
#per image
bcrp_coloc2 %>% ggplot(aes(x=sex, y=r_pearson)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
              geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()
```



```
bcrp_m1_plot <- ggplot(bcrp_coloc2, aes(x=sex, y=m1)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
    geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

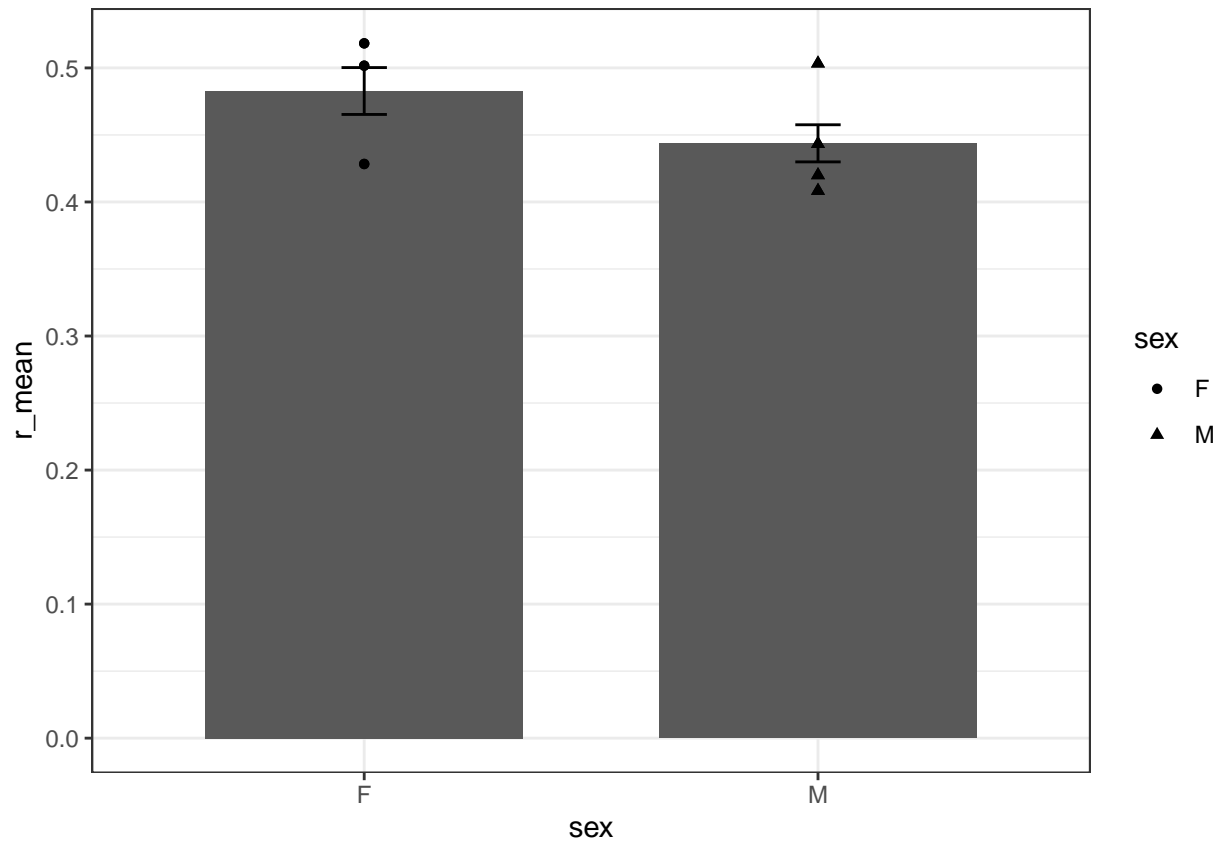
bcrp_m2_plot <- ggplot(bcrp_coloc2, aes(x=sex, y=m2)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
    geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

ggarrange(bcrp_m1_plot, bcrp_m2_plot, common.legend=TRUE, legend="right")
```

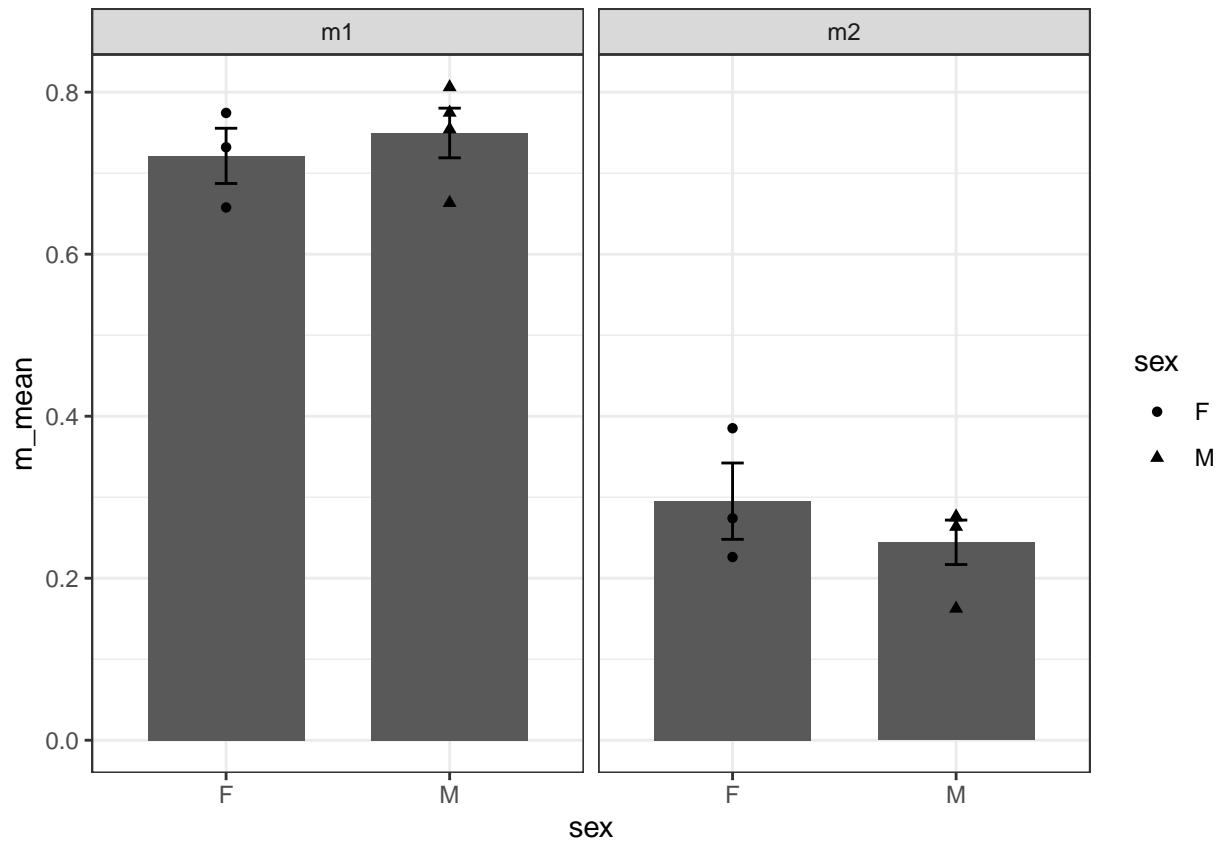


```
#per animal
bcrp_animal_r_plot <- bcrp_coloc2_long %>% ggplot(aes(x=sex, y=r_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

bcrp_animal_r_plot
```



```
bcrp_animal_m_plot <- ggplot(bcrp_coloc2_long, aes(x=sex, y=m_mean)) +  
  stat_summary(fun = 'mean', geom="bar", width=0.7) +  
  stat_summary(fun.data = mean_se,  
              geom = "errorbar", width=0.1) +  
  geom_point(color="black", aes(shape=sex)) +  
  facet_wrap(~m_stat_type) +  
  theme_bw()  
  
bcrp_animal_m_plot
```

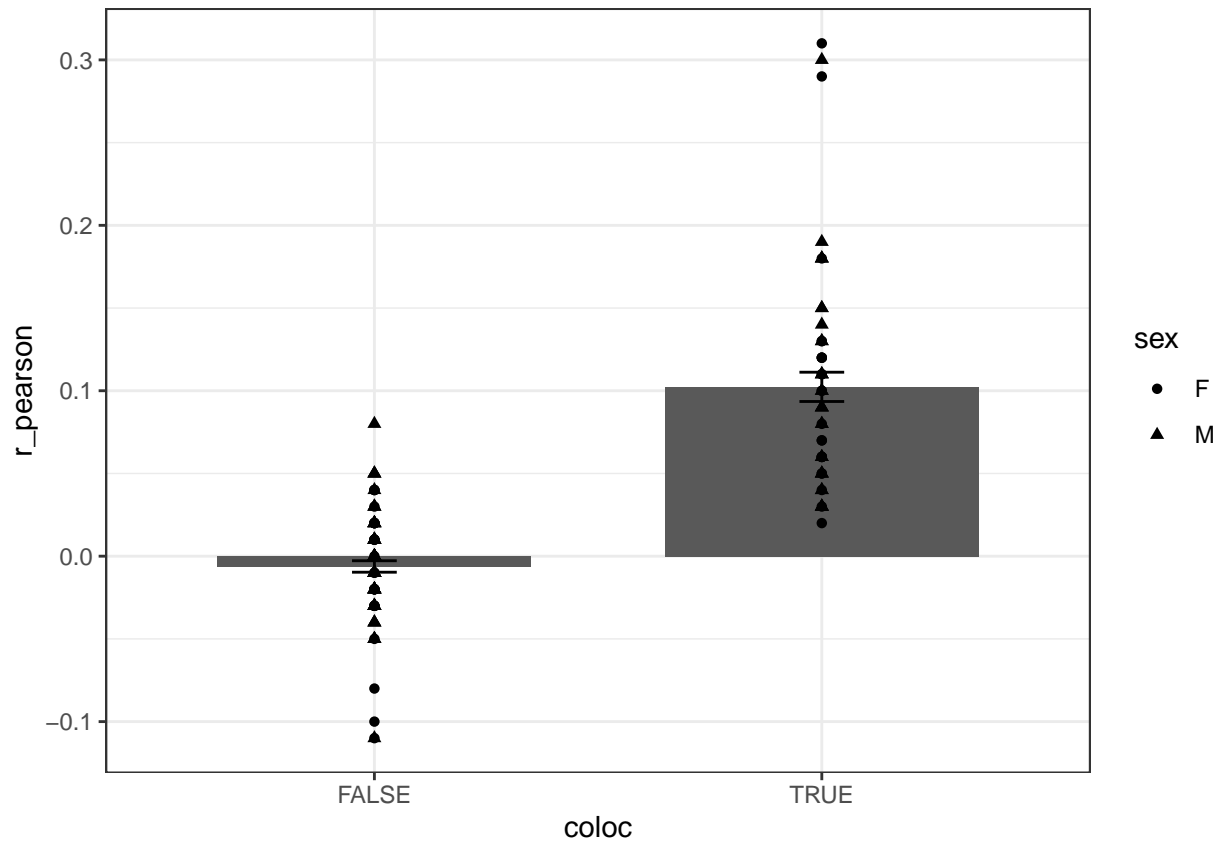
```
mrp2_coloc2$coloc <- as.logical(mrp2_coloc2$coloc)

mrp2_coloc2_long <- mrp2_coloc2 %>% pivot_longer(cols = m1:m2, names_to = "m_stat_type", values_to = "m_stat_val")

mrp2_coloc2_long <- mrp2_coloc2_long %>% group_by(mouseID, coloc, m_stat_type, sex) %>%
  summarise(m_stat_type = as.factor(m_stat_type),
            m_mean = mean(m_stat_val),
            m_sd = sd(m_stat_val),
            r_mean = mean(r_pearson),
            r_sd = sd(r_pearson),
            n_total = n(),
            n_coloc = sum(coloc)) %>% ungroup()

## 'summarise()' has grouped output by 'mouseID', 'coloc', 'm_stat_type', 'sex'.
## You can override using the '.groups' argument.
```

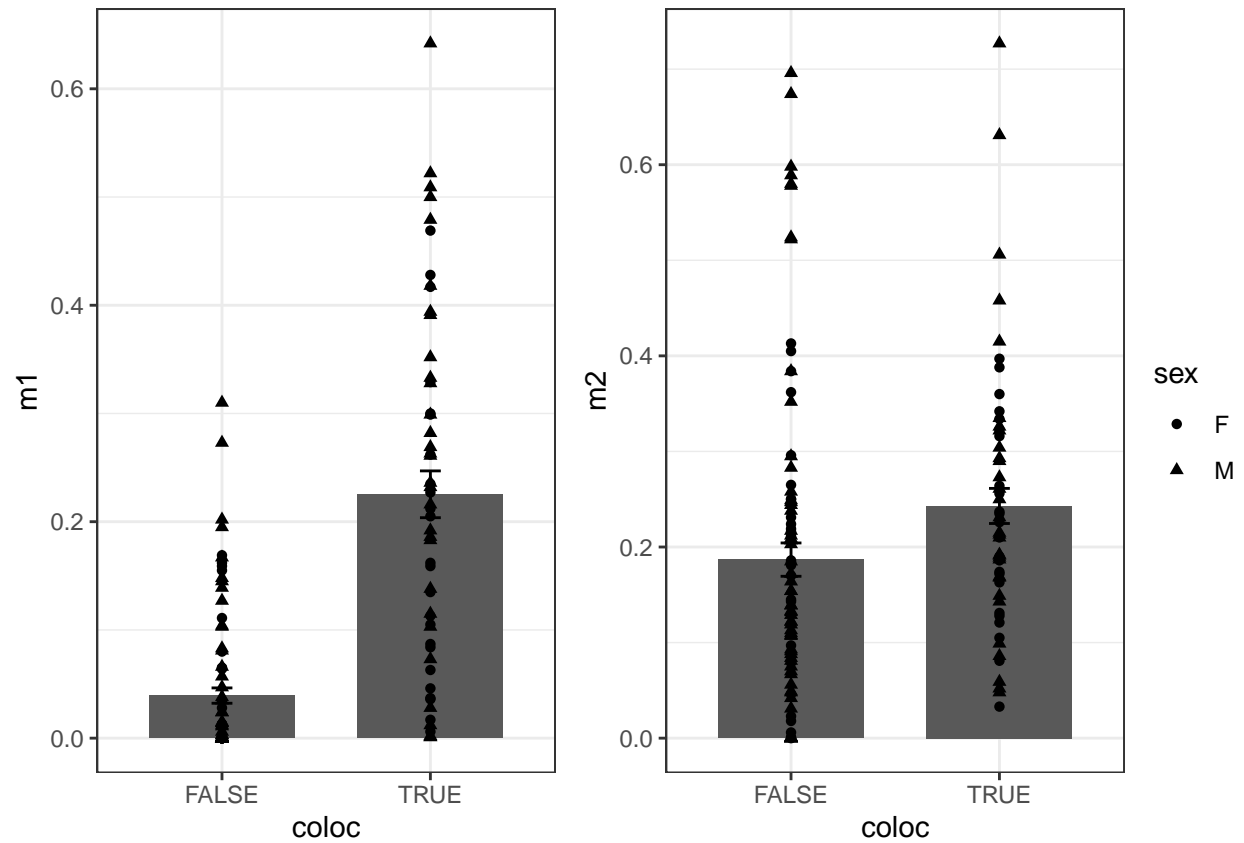
```
#per roi
mrp2_coloc2 %>% ggplot(aes(x=coloc, y=r_pearson)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
              geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()
```



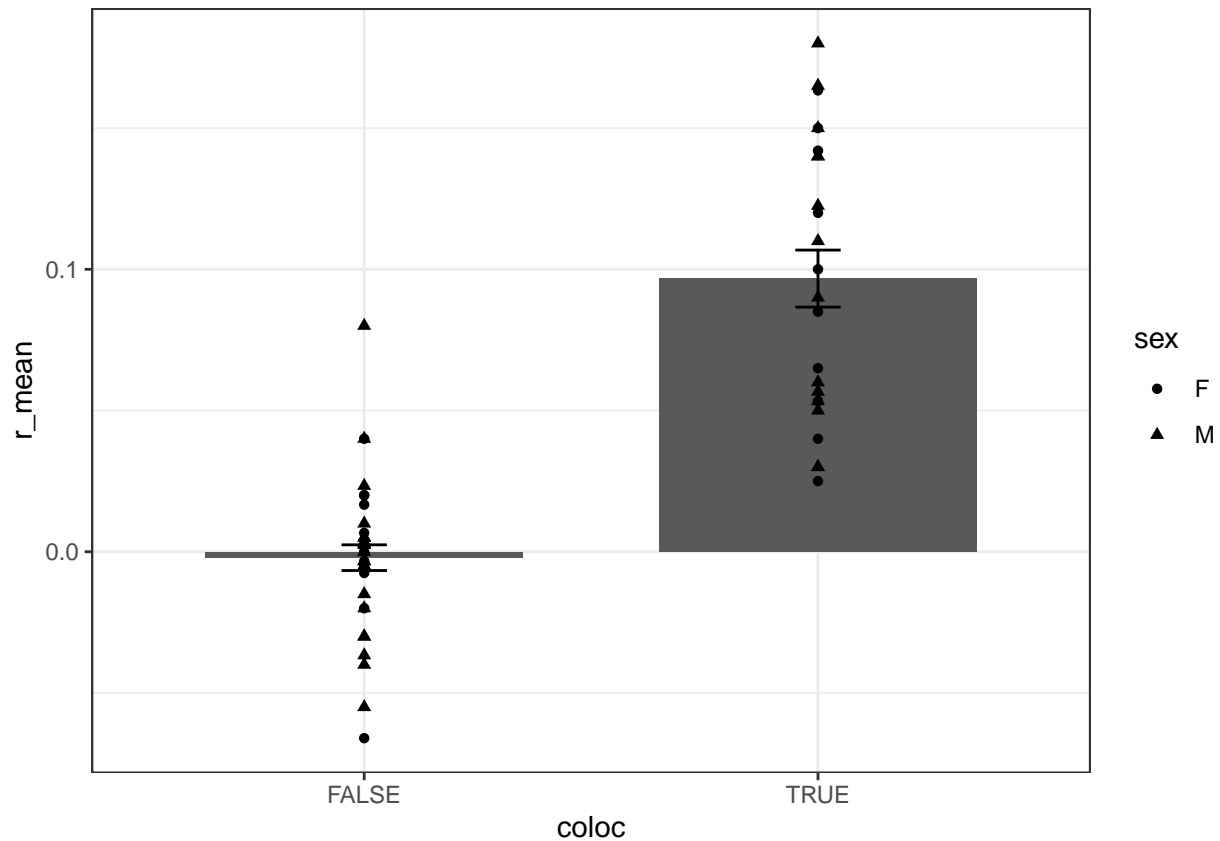
```
mrp2_m1_plot <- ggplot(mrp2_coloc2, aes(x=coloc, y=m1)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

mrp2_m2_plot <- ggplot(mrp2_coloc2, aes(x=coloc, y=m2)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

ggarrange(mrp2_m1_plot, mrp2_m2_plot, common.legend=TRUE, legend="right")
```



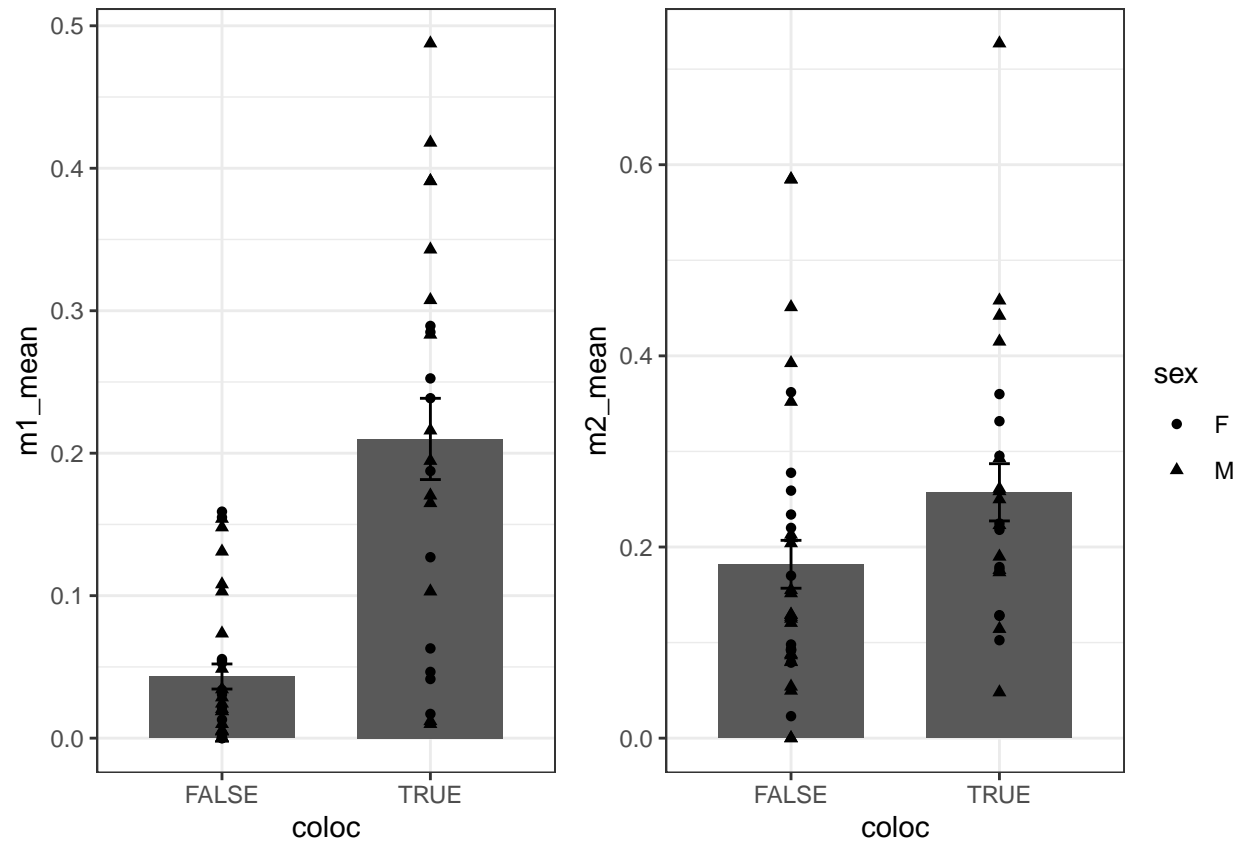
```
#per image
mrp2_sumstats_image %>% ggplot(aes(x=coloc, y=r_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()
```



```
mrp2_m1_plot_image <- mrp2_sumstats_image %>% ggplot(aes(x=coloc, y=m1_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

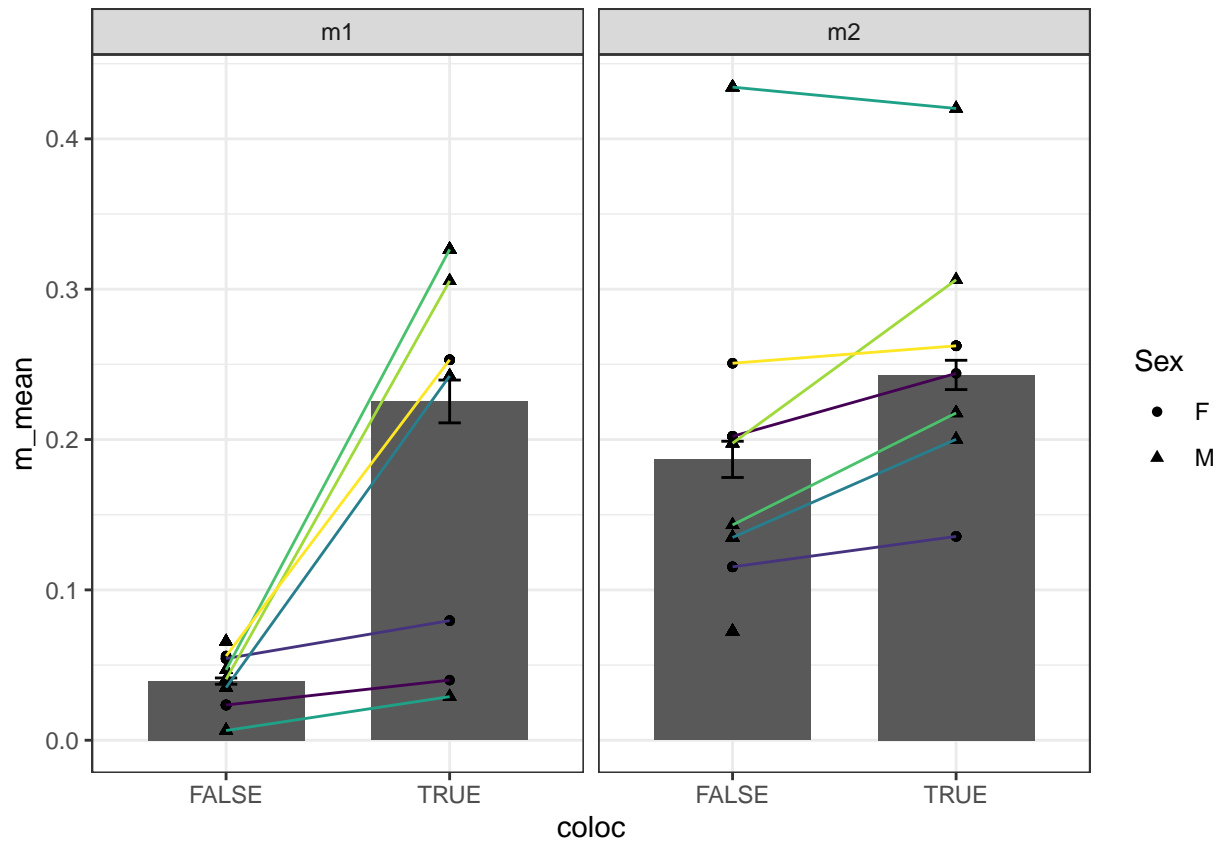
mrp2_m2_plot_image <- mrp2_sumstats_image %>% ggplot(aes(x=coloc, y=m2_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw()

ggarrange(mrp2_m1_plot_image, mrp2_m2_plot_image, common.legend=TRUE, legend="right")
```



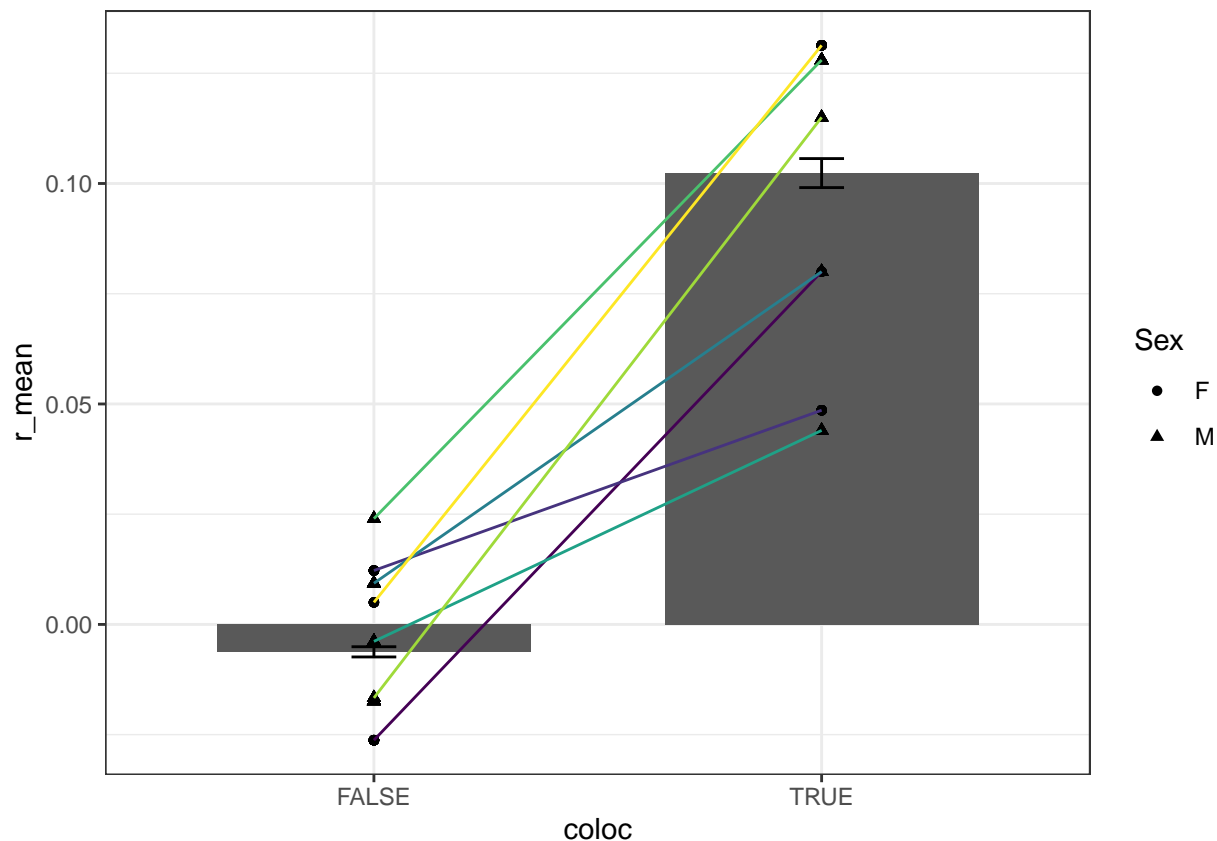
```
#per animal
mrp2_animal_m_plot <- ggplot(mrp2_coloc2_long, aes(x=coloc, y=m_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  geom_line(aes(group=mouseID, color=mouseID)) +
  facet_wrap(~m_stat_type) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw() +
  guides(shape=guide_legend("Sex"), color = "none")

mrp2_animal_m_plot
```



```
mrp2_animal_r_plot <- ggplot(mrp2_coloc2_long, aes(x=coloc, y=r_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  geom_line(aes(group=mouseID, color=mouseID)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw() +
  guides(shape=guide_legend("Sex"), color = "none")

mrp2_animal_r_plot
```



```
pgp_animal_r_plot1 <- pgp_animal_r_plot + ylim(-0.05,0.65) + ggtitle("P-gp")
bcrp_animal_r_plot1 <- bcrp_animal_r_plot + ylim(-0.05,0.65) + ggtitle("Bcrp")
mrp2_animal_r_plot1 <- mrp2_animal_r_plot + ylim(-0.05,0.65) + ggtitle("Mrp2")

all_r_plot <- ggarrange(pgp_animal_r_plot1, bcrp_animal_r_plot1, mrp2_animal_r_plot1, ncol=3, common.legend=TRUE)
ggsave("../figures/all_r_plot.png", plot=all_r_plot, width=7, height=4)
ggsave("../figures/all_r_plot.svg", plot=all_r_plot, width=7, height=4)

pgp_animal_m1_plot <- ggplot(pgp_sumstats_animals, aes(x=prep, y=m1_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw(12) +
  ylim(0,0.9) +
  ggtitle("P-gp") +
  guides(shape=guide_legend("Sex"))

bcrp_animal_m1_plot <- ggplot(bcrp_sumstats_animals, aes(x=sex, y=m1_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw(12) +
  ylim(0,0.9) +
```

```

ggtitle("Bcrp") +
guides(shape=guide_legend("Sex"))

mrp2_animal_m1_plot <- ggplot(mrp2_sumstats_animals, aes(x=coloc, y=m1_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  geom_line(aes(group=mouseID, color=mouseID)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw(12) +
  ylim(0,0.9) +
  ggtitle("Mrp2") +
  guides(shape=guide_legend("Sex"), color = "none")

all_m1_plot <- ggarrange(pgp_animal_m1_plot, bcrp_animal_m1_plot, mrp2_animal_m1_plot, ncol=3, common.l

ggsave("../figures/all_m1_plot.png", plot=all_m1_plot, width=7, height=4)
ggsave("../figures/all_m1_plot.svg", plot=all_m1_plot, width=7, height=4)

pgp_animal_m2_plot <- ggplot(pgp_sumstats_animals, aes(x=prep, y=m2_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw() +
  ylim(0,0.6) +
  ggtitle("P-gp") +
  guides(shape=guide_legend("Sex"))

bcrp_animal_m2_plot <- ggplot(bcrp_sumstats_animals, aes(x=sex, y=m2_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  theme_bw() +
  ylim(0,0.6) +
  ggtitle("Bcrp") +
  guides(shape=guide_legend("Sex"),)

mrp2_animal_m2_plot <- ggplot(mrp2_sumstats_animals, aes(x=coloc, y=m2_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(color="black", aes(shape=sex)) +
  geom_line(aes(group=mouseID, color=mouseID)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw() +
  ylim(0,0.6) +
  ggtitle("Mrp2") +

```



```
guides(shape=guide_legend("Sex"), color = "none")

all_m2_plot <- ggarrange(pgp_animal_m2_plot, bcrp_animal_m2_plot, mrp2_animal_m2_plot, ncol=3, common.l
ggsave("../figures/all_m2_plot.png", plot=all_m2_plot, width=7, height=4)
ggsave("../figures/all_m2_plot.svg", plot=all_m2_plot, width=7, height=4)
```