

Transporter (P-gp) Colocalization Analysis (ImageJ Coloc2)

CT Berezin

2022-08-18

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)
}

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,}")),
         image = str_extract(sample, "[:alpha:]{2,}.*"), .before=m1)

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	sd_rand	coloc
## 2	F	87	wm	wm-20x-1	0.010	0.047	0.21	1	0.01	0	TRUE
## 3	F	87	wm	wm-20x-2	0.021	0.044	0.28	1	-0.01	0	TRUE
## 4	F	87	wm	wm-20x-3	0.655	0.585	0.33	1	0.00	0	TRUE
## 5	F	87	wm	wm-20x-4	0.719	0.479	0.38	1	-0.01	0	TRUE
## 6	F	87	wm	wm-20x-5	0.674	0.331	0.30	1	0.00	0	TRUE
## 7	M	10	wm	GFP-wm-20x-1	0.012	0.040	0.13	1	-0.01	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 0.2532927 0.294287 0.195439 0.2355011 0.2804878 0.1387795 41 41
```

```
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: 8 x 11
## # Groups:   mouseID, sex [8]
##   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 80      M      section 0.320 0.282 0.246 0.290 0.37 0.140      6
## 4 81      M      section 0.283 0.251 0.204 0.286 0.377 0.0838    6
## 5 82      M      section 0.381 0.325 0.208 0.270 0.352 0.183      6
## 6 83      M      wm      0.430 0.373 0.368 0.274 0.328 0.0804    5
## 7 87      F      wm      0.416 0.366 0.297 0.247 0.3 0.0628     5
## 8 9       M      wm      0.0265 0.0155 0.0812 0.0538 0.138 0.0419   4
## # ... 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.

```
mrp2_sumstats_animals
```

```
## # A tibble: 15 x 11
## # Groups:   mouseID, sex [8]
##   mouseID sex coloc m1_mean m1_sd m2_mean m2_sd r_mean r_sd n_total
##   <fct> <fct> <lgl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <int>
## 1 22      F FALSE 0.0234 0.0497 0.202 0.118 -0.0262 0.0421 16
## 2 22      F TRUE 0.04 0.0325 0.244 0.164 0.08 0.0566 2
## 3 40      F FALSE 0.0543 0.0658 0.115 0.117 0.0122 0.0109 9
## 4 40      F TRUE 0.0796 0.0552 0.136 0.0669 0.0486 0.0177 7
## 5 62      M FALSE 0.0655 0.0959 0.0723 0.0747 -0.0174 0.0254 19
## 6 77      M FALSE 0.035 0.0492 0.135 0.0385 0.00933 0.0249 15
## 7 77      M TRUE 0.242 0.142 0.200 0.0504 0.08 0.0486 7
## 8 81      M FALSE 0.00646 0.00785 0.434 0.239 -0.00385 0.0112 13
## 9 81      M TRUE 0.029 0.0428 0.420 0.291 0.044 0.0313 5
## 10 82     M FALSE 0.0468 0.0604 0.143 0.0994 0.024 0.0397 5
## 11 82     M TRUE 0.326 0.170 0.218 0.116 0.128 0.0669 15
## 12 92     M FALSE 0.0406 0.0896 0.198 0.103 -0.0167 0.0424 9
## 13 92     M TRUE 0.306 0.0802 0.306 0.0831 0.115 0.03 4
## 14 98     F FALSE 0.056 0.0752 0.251 0.0982 0.005 0.0404 4
## 15 98     F TRUE 0.253 0.124 0.262 0.0856 0.131 0.0752 15
## # ... with 1 more variable: 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      0.416 0.366   0.297 0.247  0.3    0.0628     5     5
## 2 M      0.231 0.282   0.181 0.234  0.278  0.147    36    36

```

```

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  0.328 0.273   0.219 0.266  0.366  0.133    18     1
## 2 wm      0.195 0.303   0.177 0.213  0.213  0.103    23     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_coloc2$coloc <- as.factor(pgp_coloc2$coloc)

coin::wilcox_test(pgp_coloc2$r_pearson ~ pgp_coloc2$sex, conf.int=TRUE, distribution="exact")

```

```

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_coloc2$r_pearson by pgp_coloc2$sex (F, M)
## Z = 0.51834, p-value = 0.6214
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.11  0.16
## sample estimates:
## difference in location
##                0.03

```

```

coin::wilcox_test(pgp_coloc2$m1 ~ pgp_coloc2$sex, conf.int=TRUE, distribution="exact")

```

```

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_coloc2$m1 by pgp_coloc2$sex (F, M)
## Z = 0.67733, p-value = 0.5195
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.092  0.632
## sample estimates:
## difference in location
##                0.029

```

```

coin::wilcox_test(pgp_coloc2$m2 ~ pgp_coloc2$sex, conf.int=TRUE, distribution="exact")

```

```

##

```

```

## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_coloc2$m2 by pgp_coloc2$sex (F, M)
## Z = 1.1954, p-value = 0.2448
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.062  0.427
## sample estimates:
## difference in location
##                0.0355

coin::wilcox_test(pgp_coloc2$r_pearson ~ pgp_coloc2$prep, conf.int=TRUE, distribution="exact")

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_coloc2$r_pearson by pgp_coloc2$prep (section, wm)
## Z = 3.4573, p-value = 0.0003284
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  0.09 0.23
## sample estimates:
## difference in location
##                0.16

coin::wilcox_test(pgp_coloc2$m1 ~ pgp_coloc2$prep, conf.int=TRUE, distribution="exact")

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_coloc2$m1 by pgp_coloc2$prep (section, wm)
## Z = 3.1, p-value = 0.001469
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  0.064 0.200
## sample estimates:
## difference in location
##                0.1435

coin::wilcox_test(pgp_coloc2$m2 ~ pgp_coloc2$prep, conf.int=TRUE, distribution="exact")

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_coloc2$m2 by pgp_coloc2$prep (section, wm)
## Z = -1.0904, p-value = 0.2818
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.048  0.128
## sample estimates:
## difference in location
##                -0.0235

```



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

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

```
##  
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: pgp_sumstats_animals$r_mean by pgp_sumstats_animals$sex (F, M)  
## Z = -0.21822, p-value = 1  
## alternative hypothesis: true mu is not equal to 0  
## 95 percent confidence interval:  
## NA NA  
## sample estimates:  
## difference in location  
## -0.028
```

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

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

```
##  
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: pgp_sumstats_animals$m1_mean by pgp_sumstats_animals$sex (F, M)  
## Z = 1.0911, p-value = 0.5  
## alternative hypothesis: true mu is not equal to 0  
## 95 percent confidence interval:  
## NA NA  
## sample estimates:  
## difference in location  
## 0.1333
```

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

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

```
##  
## Exact Wilcoxon-Mann-Whitney Test  
##  
## data: pgp_sumstats_animals$m2_mean by pgp_sumstats_animals$sex (F, M)  
## Z = 1.0911, p-value = 0.5  
## alternative hypothesis: true mu is not equal to 0  
## 95 percent confidence interval:  
## NA NA  
## sample estimates:  
## difference in location  
## 0.0932
```

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

```

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sumstats_animals$r_mean by pgp_sumstats_animals$prep (section, wm)
## Z = 2.2361, p-value = 0.03571
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  0.02366667 0.24866667
## sample estimates:
## difference in location
##           0.2141667

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

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sumstats_animals$m1_mean by pgp_sumstats_animals$prep (section, wm)
## Z = 0.44721, p-value = 0.7857
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.1473  0.3710
## sample estimates:
## difference in location
##           0.25675

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

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  pgp_sumstats_animals$m2_mean by pgp_sumstats_animals$prep (section, wm)
## Z = 0.44721, p-value = 0.7857
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.1636000 0.2033333
## sample estimates:
## difference in location
##           0.1264167

mrp2_coloc2$coloc <- as.factor(mrp2_coloc2$coloc)
mrp2_sumstats_animals$coloc <- as.factor(mrp2_sumstats_animals$coloc)
mrp2_sumstats_image$coloc <- as.factor(mrp2_sumstats_image$coloc)

coin::wilcox_test(mrp2_coloc2$r_pearson ~ mrp2_coloc2$coloc, conf.int=TRUE, distribution="exact")

##
## Exact Wilcoxon-Mann-Whitney Test
##
## data:  mrp2_coloc2$r_pearson by mrp2_coloc2$coloc (FALSE, TRUE)
## Z = -9.665, p-value < 2.2e-16
## alternative hypothesis: true mu is not equal to 0

```

```
## 95 percent confidence interval:
##  -0.11 -0.08
## sample estimates:
## difference in location
##          -0.1
```

```
coin::wilcox_test(mrp2_coloc2$m1 ~ mrp2_coloc2$coloc, conf.int=TRUE, distribution="exact")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_coloc2$m1 by mrp2_coloc2$coloc (FALSE, TRUE)
## Z = -7.9711, p-value < 2.2e-16
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.226 -0.125
## sample estimates:
## difference in location
##          -0.183
```

```
coin::wilcox_test(mrp2_coloc2$m2 ~ mrp2_coloc2$coloc, conf.int=TRUE, distribution="exact")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_coloc2$m2 by mrp2_coloc2$coloc (FALSE, TRUE)
## Z = -3.0646, p-value = 0.002026
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.111 -0.028
## sample estimates:
## difference in location
##          -0.071
```

```
coin::wilcox_test(mrp2_coloc2$r_pearson ~ mrp2_coloc2$sex, conf.int=TRUE, distribution="exact")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_coloc2$r_pearson by mrp2_coloc2$sex (F, M)
## Z = 0.98923, p-value = 0.3242
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
##  -0.01  0.03
## sample estimates:
## difference in location
##          0.01
```

```
coin::wilcox_test(mrp2_coloc2$m1 ~ mrp2_coloc2$sex, conf.int=TRUE, distribution="exact")
```

```
##
```

```
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_coloc2$m1 by mrp2_coloc2$sex (F, M)
## Z = 0.25773, p-value = 0.798
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.006 0.023
## sample estimates:
## difference in location
## 0
```

```
coin::wilcox_test(mrp2_coloc2$m2 ~ mrp2_coloc2$sex, conf.int=TRUE, distribution="exact")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_coloc2$m2 by mrp2_coloc2$sex (F, M)
## Z = 0.73704, p-value = 0.463
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.030 0.062
## sample estimates:
## difference in location
## 0.017
```

```
coin::wilcox_test(mrp2_sumstats_image$r_mean ~ mrp2_sumstats_image$coloc, conf.int=TRUE, distribution="exact")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_image$r_mean by mrp2_sumstats_image$coloc (FALSE, TRUE)
## Z = -6.2203, p-value = 1.005e-13
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.1258333 -0.0675000
## sample estimates:
## difference in location
## -0.0975
```

```
coin::wilcox_test(mrp2_sumstats_image$m1_mean ~ mrp2_sumstats_image$coloc, conf.int=TRUE, distribution="exact")
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_image$m1_mean by mrp2_sumstats_image$coloc (FALSE, TRUE)
## Z = -4.7939, p-value = 2.868e-07
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.23385 -0.09850
## sample estimates:
## difference in location
## -0.1665
```

```
coin::wilcox_test(mrp2_sumstats_image$m2_mean ~ mrp2_sumstats_image$coloc, conf.int=TRUE, distribution="ex
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_image$m2_mean by mrp2_sumstats_image$coloc (FALSE, TRUE)
## Z = -2.5152, p-value = 0.01124
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.1379167 -0.0210000
## sample estimates:
## difference in location
## -0.084
```

```
coin::wilcox_test(mrp2_sumstats_image$r_mean ~ mrp2_sumstats_image$sex, conf.int=TRUE, distribution="ex
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_image$r_mean by mrp2_sumstats_image$sex (F, M)
## Z = 0.55663, p-value = 0.5833
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.01666667 0.03666667
## sample estimates:
## difference in location
## 0.01
```

```
coin::wilcox_test(mrp2_sumstats_image$m1_mean ~ mrp2_sumstats_image$sex, conf.int=TRUE, distribution="ex
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_image$m1_mean by mrp2_sumstats_image$sex (F, M)
## Z = 0.20389, p-value = 0.8428
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.0565 0.0365
## sample estimates:
## difference in location
## 0.001458333
```

```
coin::wilcox_test(mrp2_sumstats_image$m2_mean ~ mrp2_sumstats_image$sex, conf.int=TRUE, distribution="ex
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_image$m2_mean by mrp2_sumstats_image$sex (F, M)
## Z = -0.20378, p-value = 0.8429
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
```

```
## -0.082 0.050
## sample estimates:
## difference in location
## -0.01
```

```
coin::wilcox_test(mrp2_sumstats_animals$r_mean ~ mrp2_sumstats_animals$coloc, conf.int=TRUE, distributi
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$r_mean by
## mrp2_sumstats_animals$coloc (FALSE, TRUE)
## Z = -3.2433, p-value = 0.0003108
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.1316667 -0.0560000
## sample estimates:
## difference in location
## -0.0966667
```

```
coin::wilcox_test(mrp2_sumstats_animals$m1_mean ~ mrp2_sumstats_animals$coloc, conf.int=TRUE, distribut
```

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$m1_mean by
## mrp2_sumstats_animals$coloc (FALSE, TRUE)
## Z = -1.9674, p-value = 0.05408
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.264944444 0.000555556
## sample estimates:
## difference in location
## -0.191719
```

```
coin::wilcox_test(mrp2_sumstats_animals$m1_mean ~ mrp2_sumstats_animals$coloc, conf.int=TRUE)
```

```
##
## Asymptotic Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$m1_mean by
## mrp2_sumstats_animals$coloc (FALSE, TRUE)
## Z = -1.9674, p-value = 0.04914
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.26099299 -0.00499999
## sample estimates:
## difference in location
## -0.193544
```

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

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$m2_mean by
## mrp2_sumstats_animals$coloc (FALSE, TRUE)
## Z = -1.5045, p-value = 0.152
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.17138333 0.03308333
## sample estimates:
## difference in location
## -0.06502698
```

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

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$r_mean by mrp2_sumstats_animals$sex (F, M)
## Z = 0.17693, p-value = 0.8881
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.06777778 0.06593985
## sample estimates:
## difference in location
## 0.003111111
```

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

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$m1_mean by mrp2_sumstats_animals$sex (F, M)
## Z = 0, p-value = 1
## alternative hypothesis: true mu is not equal to 0
## 95 percent confidence interval:
## -0.21884821 0.04457143
## sample estimates:
## difference in location
## 0.002222222
```

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

```
##
## Exact Wilcoxon-Mann-Whitney Test
##
## data: mrp2_sumstats_animals$m2_mean by mrp2_sumstats_animals$sex (F, M)
## Z = -0.11785, p-value = 0.9546
## alternative hypothesis: true mu is not equal to 0
```

```
## 95 percent confidence interval:
## -0.17205128  0.06725833
## sample estimates:
## difference in location
## -0.01158512
```

Plotting the data

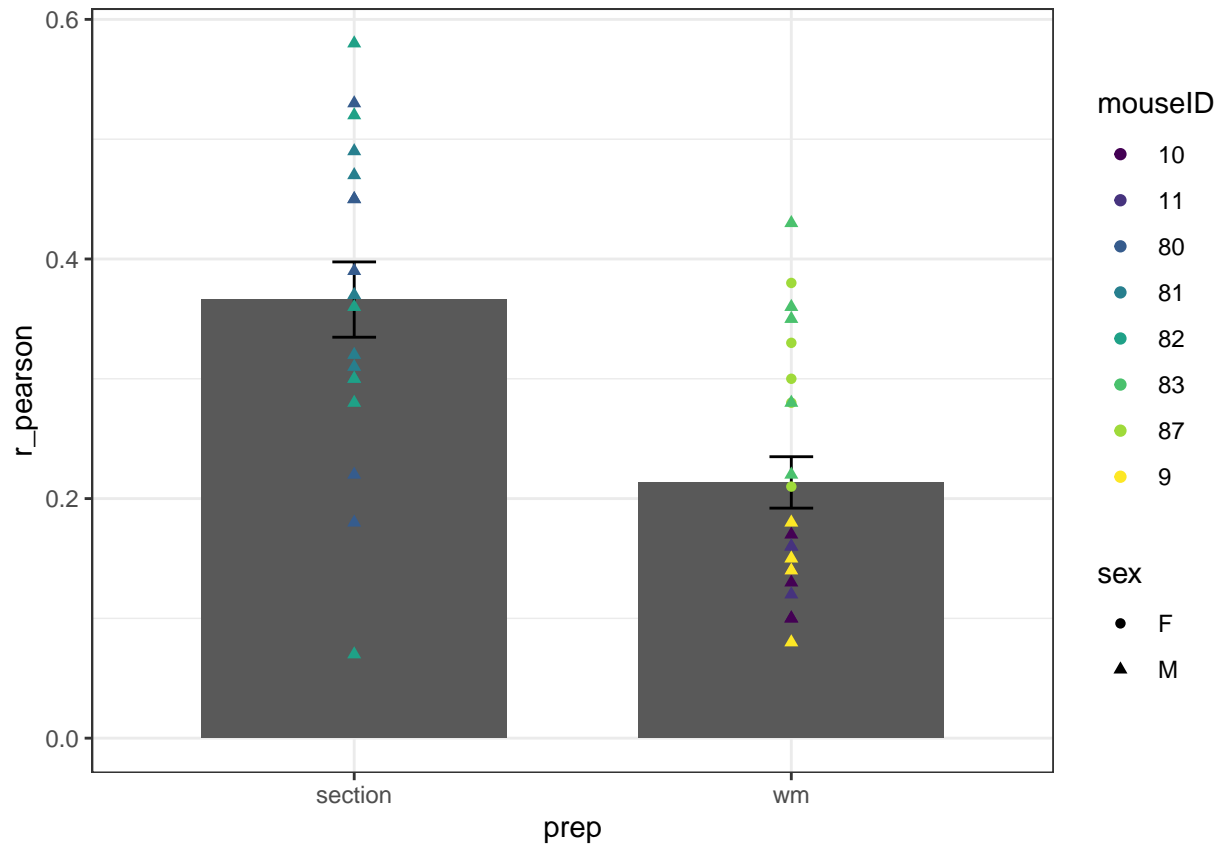
```
pgp_coloc2$coloc <- as.logical(pgp_coloc2$coloc)
pgp_coloc2$mouseID <- as.factor(pgp_coloc2$mouseID)

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

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.
```

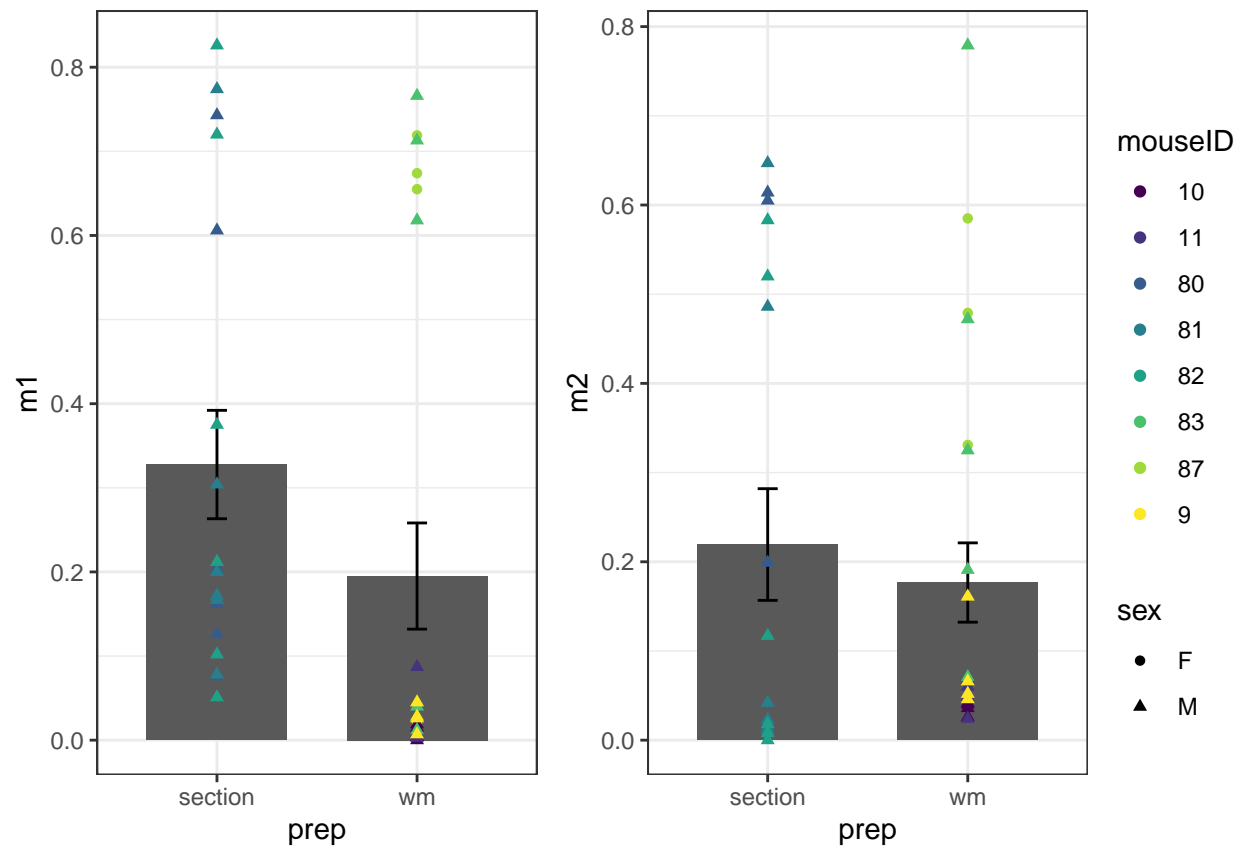
```
#per image
pgp_coloc2 %>% ggplot(aes(x=prep, y=r_pearson)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
              geom = "errorbar", width=0.1) +
  geom_point(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()
```

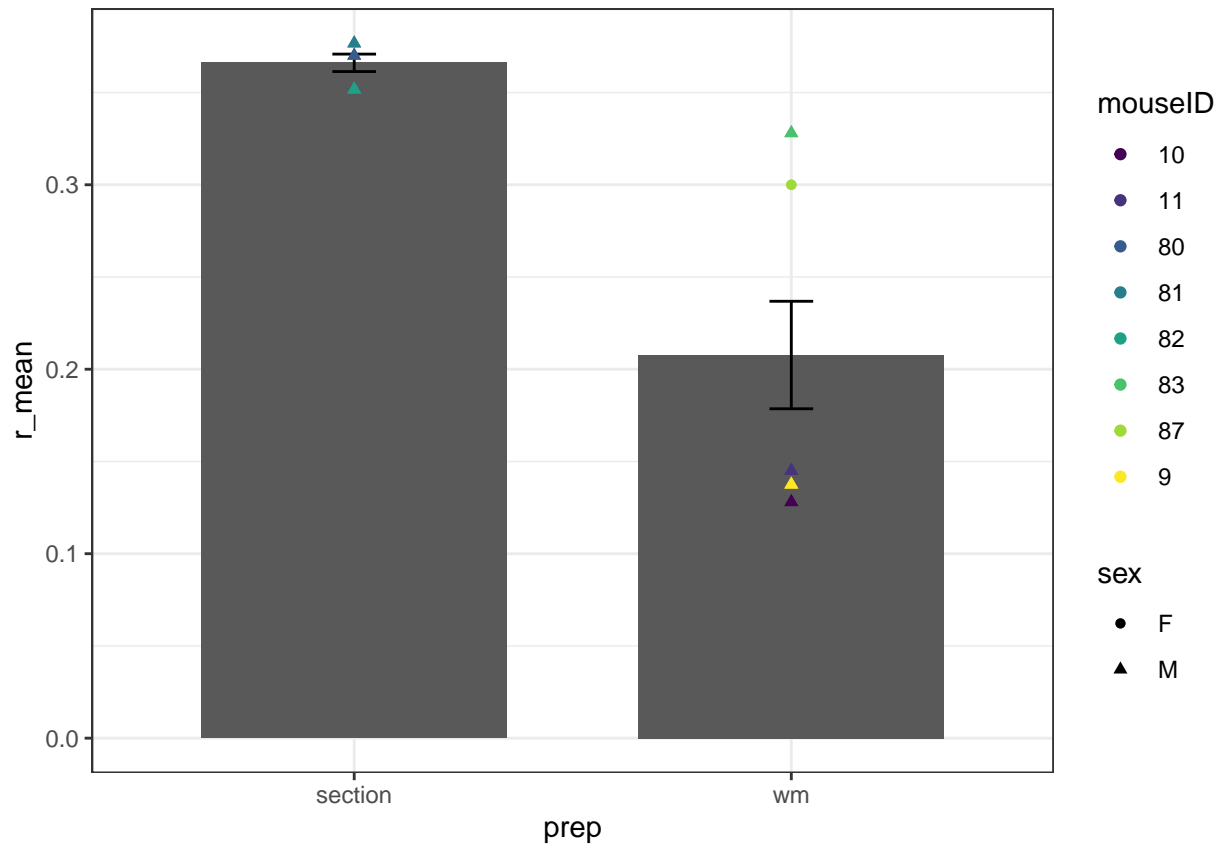
```
pgp_m1_plot <- ggplot(pgp_coloc2, aes(x=prep, y=m1)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
    geom = "errorbar", width=0.1) +
  geom_point(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()

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(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()

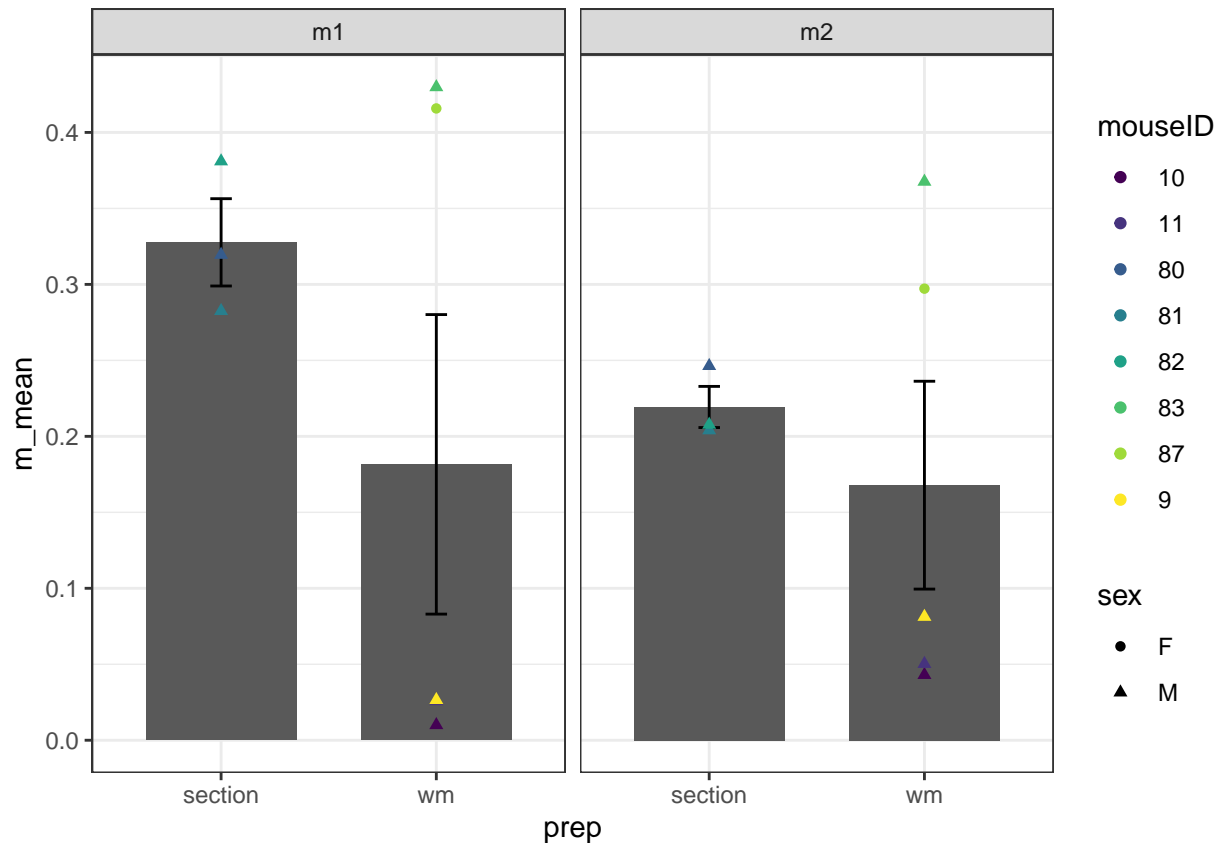
ggarrange(pgp_m1_plot, pgp_m2_plot, common.legend=TRUE, legend="right")
```



```
#per animal
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(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()
```



```
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(aes(color=mouseID, shape=sex)) +
  facet_wrap(~m_stat_type) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()
```



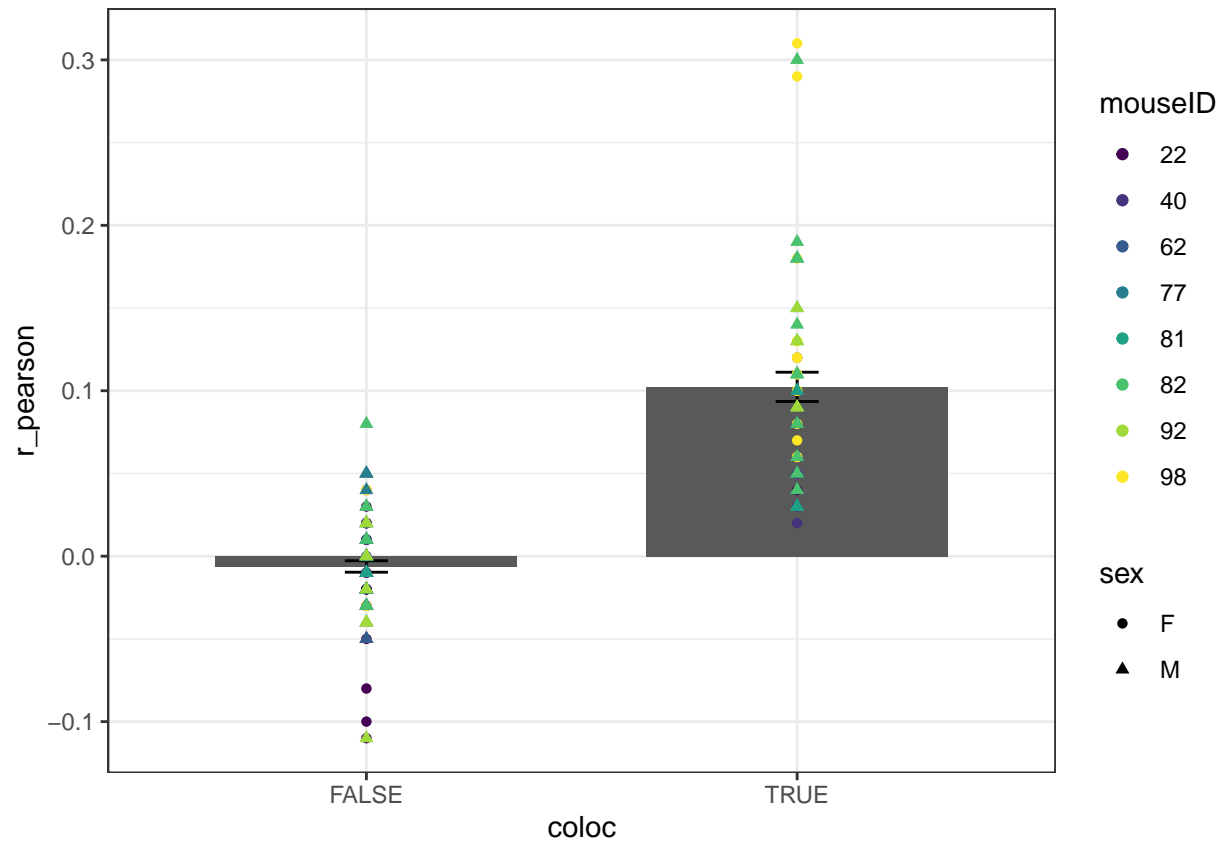
```
mrp2_coloc2$coloc <- as.logical(mrp2_coloc2$coloc)

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.
```

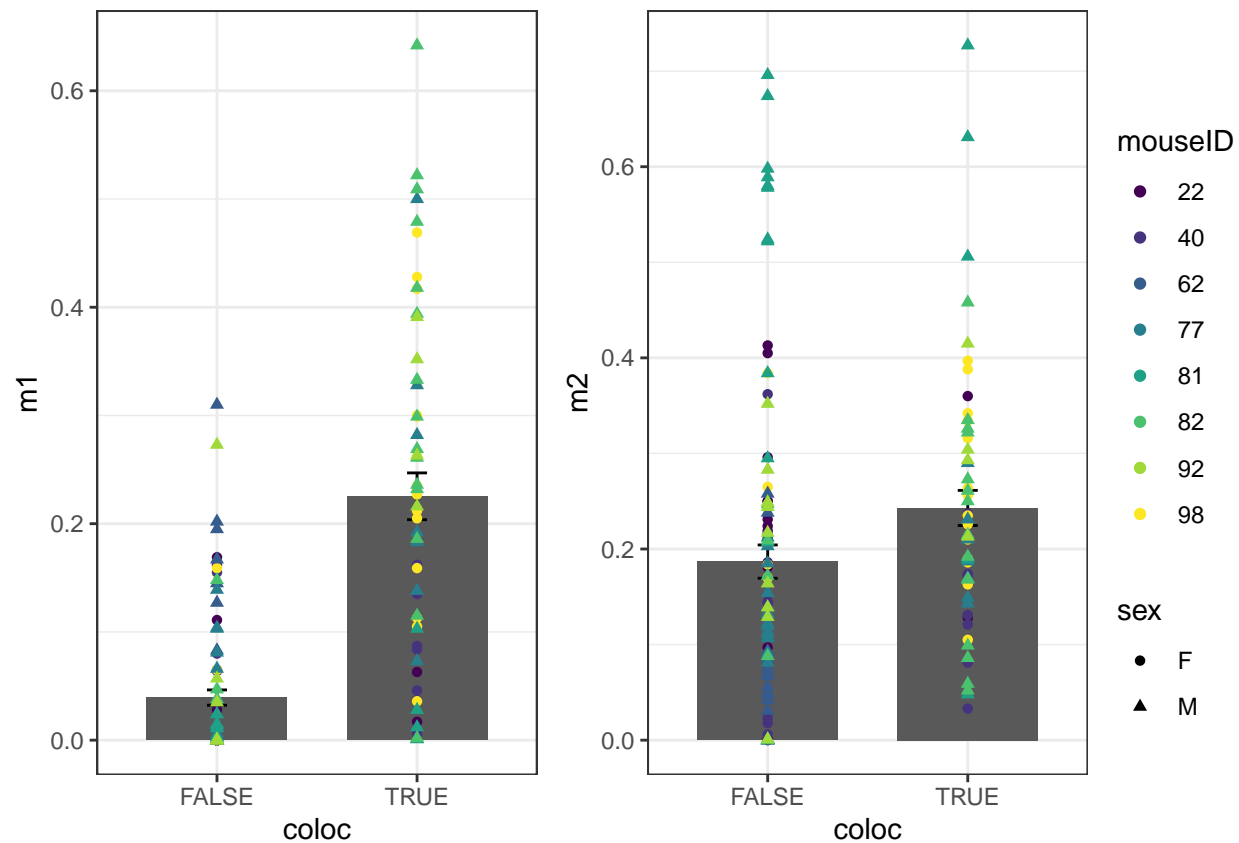
```
#per roi
mrp2_coloc2 %>% ggplot(aes(x=coloc, y=r_pearson)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
              geom = "errorbar", width=0.1) +
  geom_point(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()
```



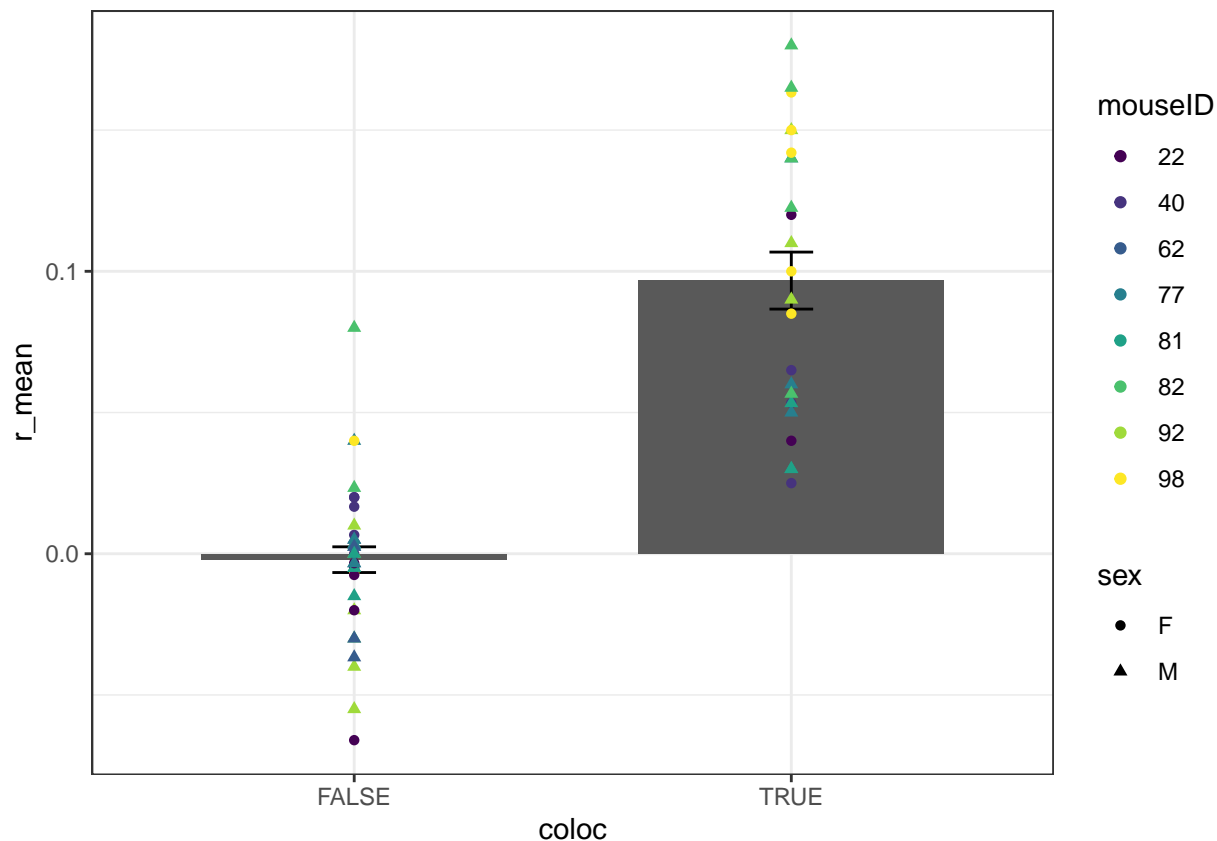
```
mrp2_m1_plot <- ggplot(mrp2_coloc2, aes(x=coloc, y=m1)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()

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(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()

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



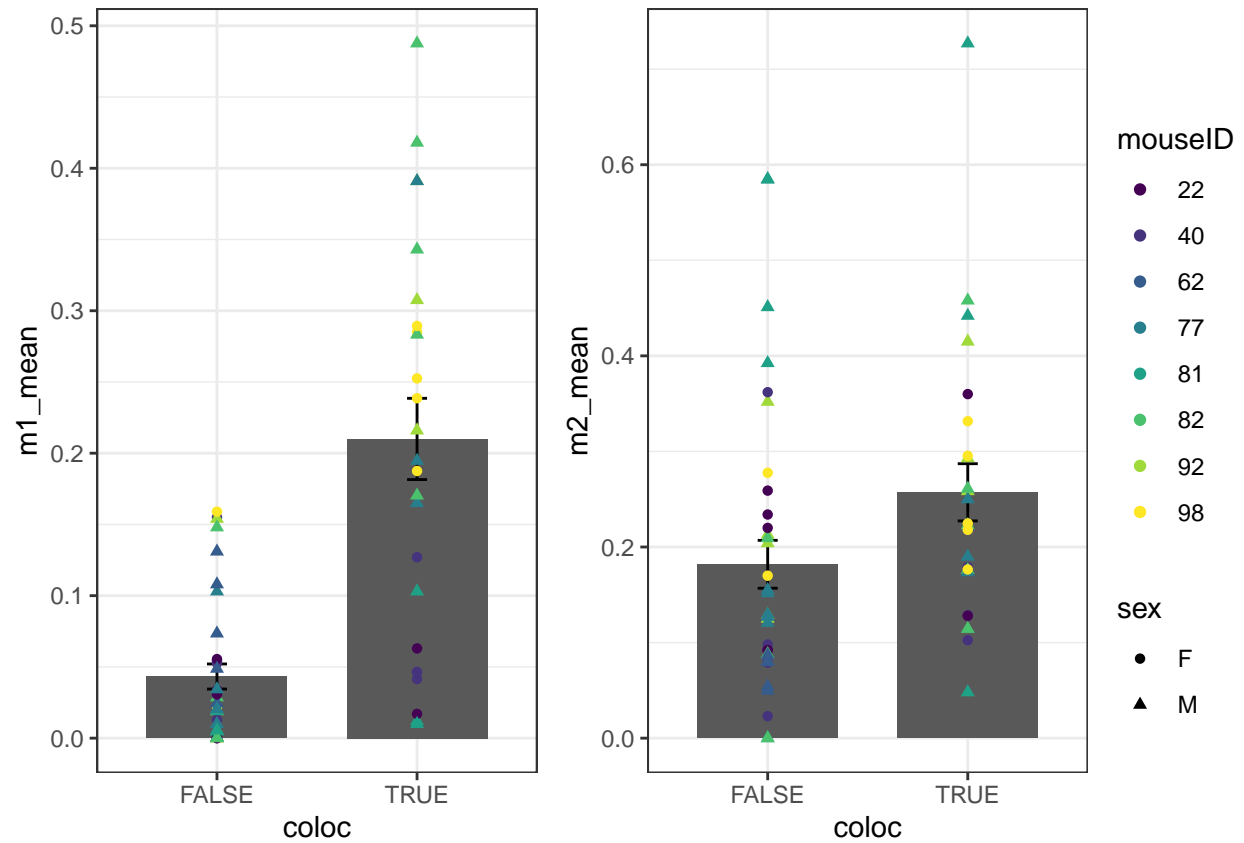
```
#per image
mrp2_sumstats_image %>% ggplot(aes(x=coloc, y=r_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()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



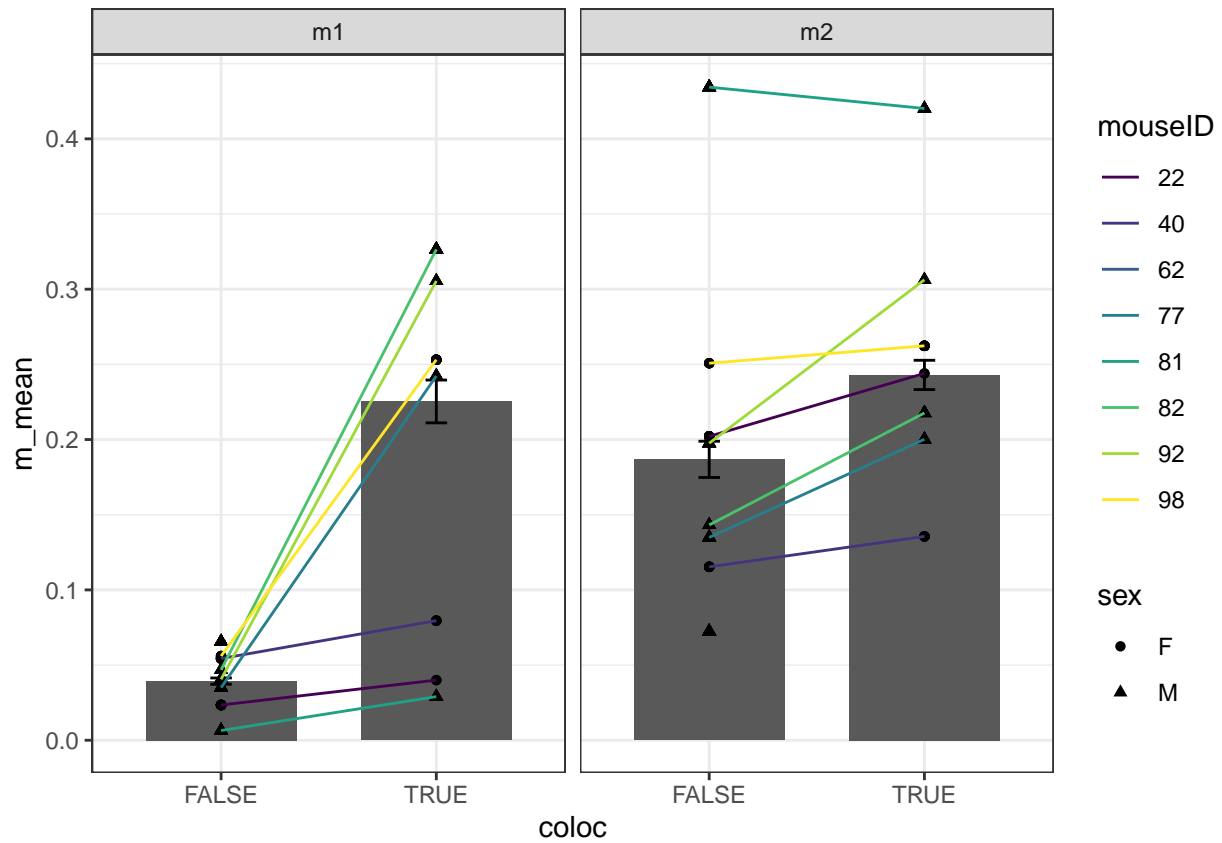
```
mrp2_m1_plot_image <- mrp2_sumstats_image %>% ggplot(aes(x=coloc, y=m1_mean)) +
  stat_summary(fun = 'mean', geom="bar", width=0.7) +
  stat_summary(fun.data = mean_se,
               geom = "errorbar", width=0.1) +
  geom_point(aes(color=mouseID, shape=sex)) +
  scale_color_viridis(discrete="TRUE")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

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



```
#per animal
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(aes(shape=sex)) +
  geom_line(aes(group=mouseID, color=mouseID)) +
  facet_wrap(~m_stat_type) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()
```

```
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(aes(shape=sex)) +
  geom_line(aes(group=mouseID, color=mouseID)) +
  scale_color_viridis(discrete="TRUE") +
  theme_bw()
```

