sldsc_jmc_heatmap_suppfig_s16

Kate Isaac

2022-12-13

Making a heatmap/dotplot hybrid

The desired order of the JMC/metacluster axis

1 4 12 9 8 10 2 5 3 15 7 11 14 6 13 All traits that were found as significant in at least one metacluster

Enrichment score to be the color of the circle and then we want to overlay a circle whose size reflects the significance don't plot a circle for a trait that is not significant for that given metacluster

```
library(tidyverse)
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6      v purrr   0.3.4
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.2      v forcats 0.5.2
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(dplyr)
```

read in the metacluster enrichment data read in data that labels the trait type for all of the traits

```
metacluster_enrichments <- read.delim("~/Downloads/cCRE_15-metaclusters_all_UKBB_traits.enrichment.txt"
trait_type_only <- read.delim("~/mccoyLab/VISION/sldsc/trait_table_description_significance_type.txt", 
```

Add the label for the type of trait to the metacluster enrichment data using a full join

And set the metacluster number as its own variable, with a factor level reflecting the order we want it plotted

```
metacluster_enrichments_withtype <- full_join(metacluster_enrichments, trait_type_only, by="trait")
metacluster_enrichments_withtype$metacluster <- factor(unlist(lapply(1:nrow(metacluster_enrichments_wit
```

fix our labels to be prettier/match the other volcano figure

```
metacluster_enrichments_withtype$label <- metacluster_enrichments_withtype$description
metacluster_enrichments_withtype <- metacluster_enrichments_withtype %>%
  mutate(across("label", str_replace, "White blood cell \\(leukocyte\\)", "WBC")) %>%
  mutate(across("label", str_replace, "Red blood cell \\(erythrocyte\\)", "RBC")) %>%
  mutate(across("label", str_replace, "haemoglobin", "Hb")) %>%
  mutate(across("label", str_replace, "Haemoglobin", "Hb")) %>%
  mutate(across("label", str_replace, "Neutrophill", "NEU")) %>%
  mutate(across("label", str_replace, "Platelet", "PLT")) %>%
  mutate(across("label", str_replace, "platelet \\(thrombocyte\\)", "PLT")) %>%
  mutate(across("label", str_replace, "Monocyte", "MON")) %>%
  mutate(across("label", str_replace, 'reticulocyte', "RTIC")) %>%
  mutate(across("label", str_replace, 'Reticulocyte', "RTIC")) %>%
  mutate(across("label", str_replace, "Lymphocyte", "LYM")) %>%
  mutate(across("label", str_replace, "Basophil", "BASO")) %>%
  mutate(across("label", str_replace, "Eosinophill", "EOS")) %>%
  mutate(across("label", str_replace, "Mean corpuscular volume", "MCV")) %>%
  mutate(across("label", str_replace, "Mean", "avg")) %>%
  mutate(across("label", str_replace, "percentage", "%")) %>%
  mutate(across("label", str_replace, "distribution", "dist")) %>%
  mutate(across("label", str_replace, "volume", "vol")) %>%
  mutate(across("label", str_replace, "concentration", "conc")) %>%
  mutate(across("label", str_replace, "count", "ct")) %>%
  mutate(across("label", str_replace, "Immature", "Im")) %>%
  mutate(across("label", str_replace, "Average", "avg")) %>%
  mutate(across("label", str_replace, "Glycated haemoglobin (mmol/mol)", "Glycated Hb")) %>%
  mutate(across("label", str_replace, "Forced expiratory vol in 1-second \\(FEV1\\), Best measure", "FE
  mutate(across("label", str_replace, "\\(left\\)", "\\(l\\)")) %>%
  mutate(across("label", str_replace, "\\(right\\)", "\\(r\\)"))
```

get female only

```
metacluster_enrichments_withtype$sex <- unlist(lapply(1:nrow(metacluster_enrichments_withtype), functio

metacluster_enrichments_withtypefem <- metacluster_enrichments_withtype[which(metacluster_enrichments_w
```

get male only

```
metacluster_enrichments_withtypemale <- metacluster_enrichments_withtype[which(metacluster_enrichments_
```

Focusing on female

Which traits are significant?

```
roi <- which(metacluster_enrichments_withtypefem$Enrichment_p_adj < 0.05)
uniq_sig_traits <- unique(metacluster_enrichments_withtypefem[roi,"trait"])
```

Let’s make some matrices that we can flatten for plotting, but they’re going to store the size of the point (reflecting significance; 0 if not significant)

and the color of the point (reflecting enrichment)

```
matSize <- matrix(NA, nrow=length(unique(metacluster_enrichments_withtypefem$metacluster)), ncol = leng

matColor <- matrix(NA, nrow=length(unique(metacluster_enrichments_withtypefem$metacluster)), ncol = len
```

```
for (metacluster_num in names(table(metacluster_enrichments_withtypefem$metacluster))){
  sig_traits <- metacluster_enrichments_withtypefem[which(metacluster_enrichments_withtypefem$metaclust
  which_cols_in_mat <- unlist(lapply(1:nrow(sig_traits), function(i) which(colnames(matSize) == sig_tra
  matSize[metacluster_num, which_cols_in_mat] = sig_traits[,"Enrichment_p_adj"]
  matColor[metacluster_num, which_cols_in_mat] = sig_traits[,"Enrichment"]
}
```

Focusing on male

Which traits are significant?

```
roi <- which(metacluster_enrichments_withtypemale$Enrichment_p_adj < 0.05)
uniq_sig_traits <- unique(metacluster_enrichments_withtypemale[roi,"trait"])
```

Let's make some matrices that we can flatten for plotting, but they're going to store the size of the point (reflecting significance; 0 if not significant)

and the color of the point (reflecting enrichment)

```
matSizeMale <- matrix(NA, nrow=length(unique(metacluster_enrichments_withtypemale$metacluster)), ncol =

matColorMale <- matrix(NA, nrow=length(unique(metacluster_enrichments_withtypemale$metacluster)), ncol
```

```
for (metacluster_num in names(table(metacluster_enrichments_withtypemale$metacluster))){
  sig_traits <- metacluster_enrichments_withtypemale[which(metacluster_enrichments_withtypemale$metaclus
  which_cols_in_mat <- unlist(lapply(1:nrow(sig_traits), function(i) which(colnames(matSizeMale) == sig
  matSizeMale[metacluster_num, which_cols_in_mat] = sig_traits[,"Enrichment_p_adj"]
  matColorMale[metacluster_num, which_cols_in_mat] = sig_traits[,"Enrichment"]
}
```

So we need to plot the matrices... we'll want to collapse the matrices into a dataframe

```
femMat <- full_join(pivot_longer(cbind(jmc=row.names(matColor), data.frame(matColor)), -c(jmc), names_t
femMat$trait <- str_replace(femMat$trait, "X", "")
femMatWithLabel <- left_join(femMat, metacluster_enrichments_withtypefem[,c("label", "trait", "type")],

maleMat <- full_join(pivot_longer(cbind(jmc=row.names(matColorMale), data.frame(matColorMale)), -c(jmc)
maleMat$trait <- str_replace(maleMat$trait, "X", "")
maleMatWithLabel <- left_join(maleMat, metacluster_enrichments_withtypemale[,c("label", "trait", "type"
```

set up colors for the tick labels to specify if the trait is a blood count (maroon/red), blood chemistry (purple), or other (black) trait.

```
femMatWithLabel <- femMatWithLabel %>%
  mutate(tick_colors = ifelse(type == "count", "maroon", ifelse(type == "chemistry", "purple", "black")

maleMatWithLabel <- maleMatWithLabel %>%
  mutate(tick_colors = ifelse(type == "count", "maroon", ifelse(type == "chemistry", "purple", "black")
```

```
tick_colors_fem <- unlist(lapply(1:length(unique(femMatWithLabel$label)), function(x) femMatWithLabel[wh
```

```
gf <- ggplot(femMatWithLabel, aes(y=factor(jmc, levels = rev(c( "1","4","12","9","8","10","2","5","3","
```

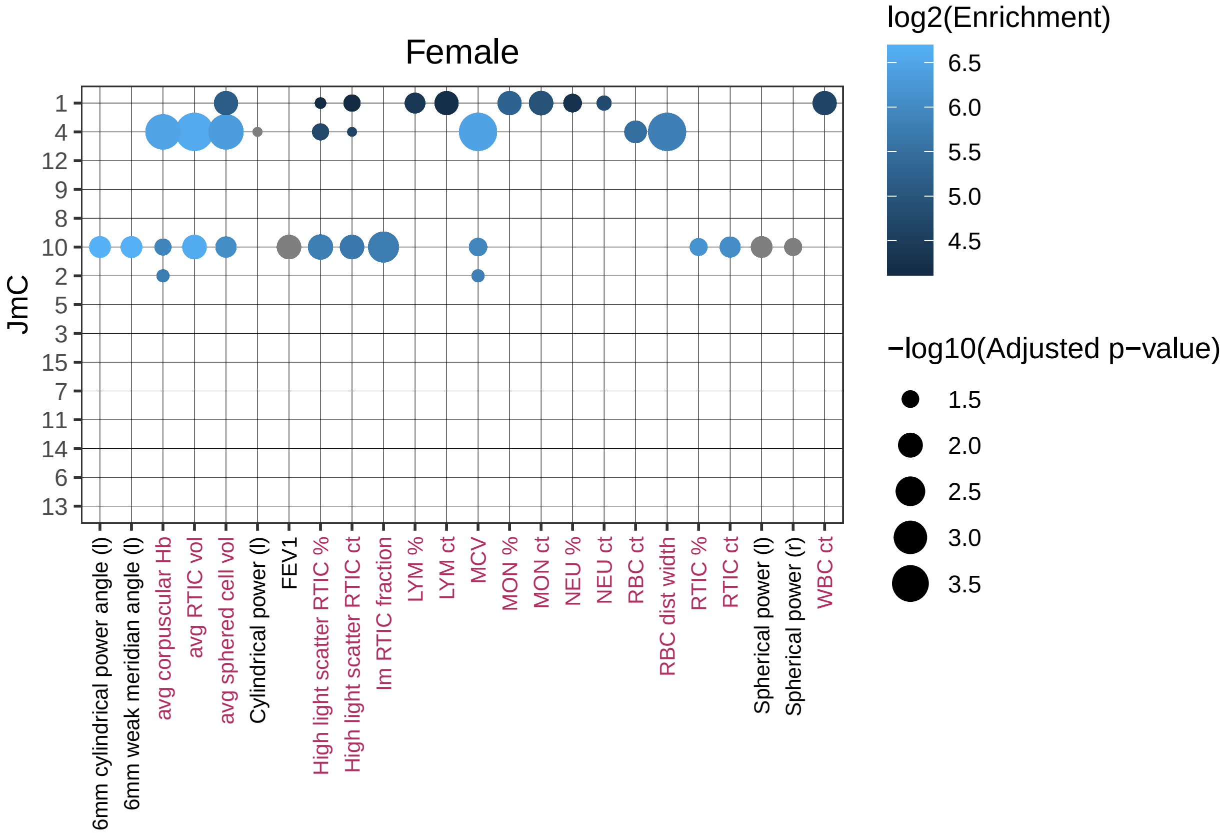
```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
gf
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```

```
## Warning in FUN(X[[i]], ...): NaNs produced
```

```
## Warning: Removed 4875 rows containing missing values (geom_point).
```



```
ggsave("~/mccoyLab/VISION/sldsc/jmc_female_straight.pdf", plot = gf)
```

```
## Saving 6.5 x 4.5 in image

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning: Removed 4875 rows containing missing values (geom_point).
```

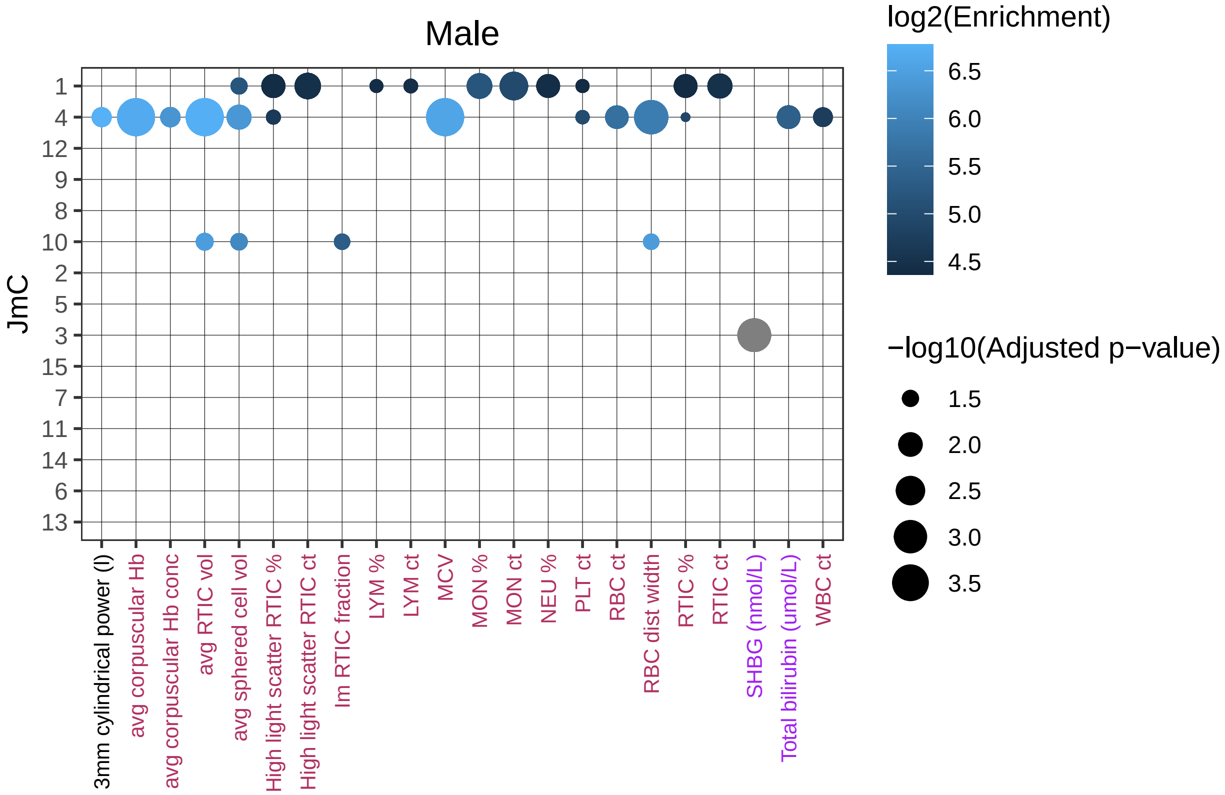
```
tick_colors_male <- unlist(lapply(1:length(unique(maleMatWithLabel$label)), function(x) maleMatWithLabel
```

```
gm <- ggplot(maleMatWithLabel, aes(y=factor(jmc, levels = rev(c( "1","4","12","9","8","10","2","5","3",
```

```
## Warning: Vectorized input to `element_text()` is not officially supported.
## Results may be unexpected or may change in future versions of ggplot2.
```

```
gm
```

```
## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning: Removed 4515 rows containing missing values (geom_point).
```

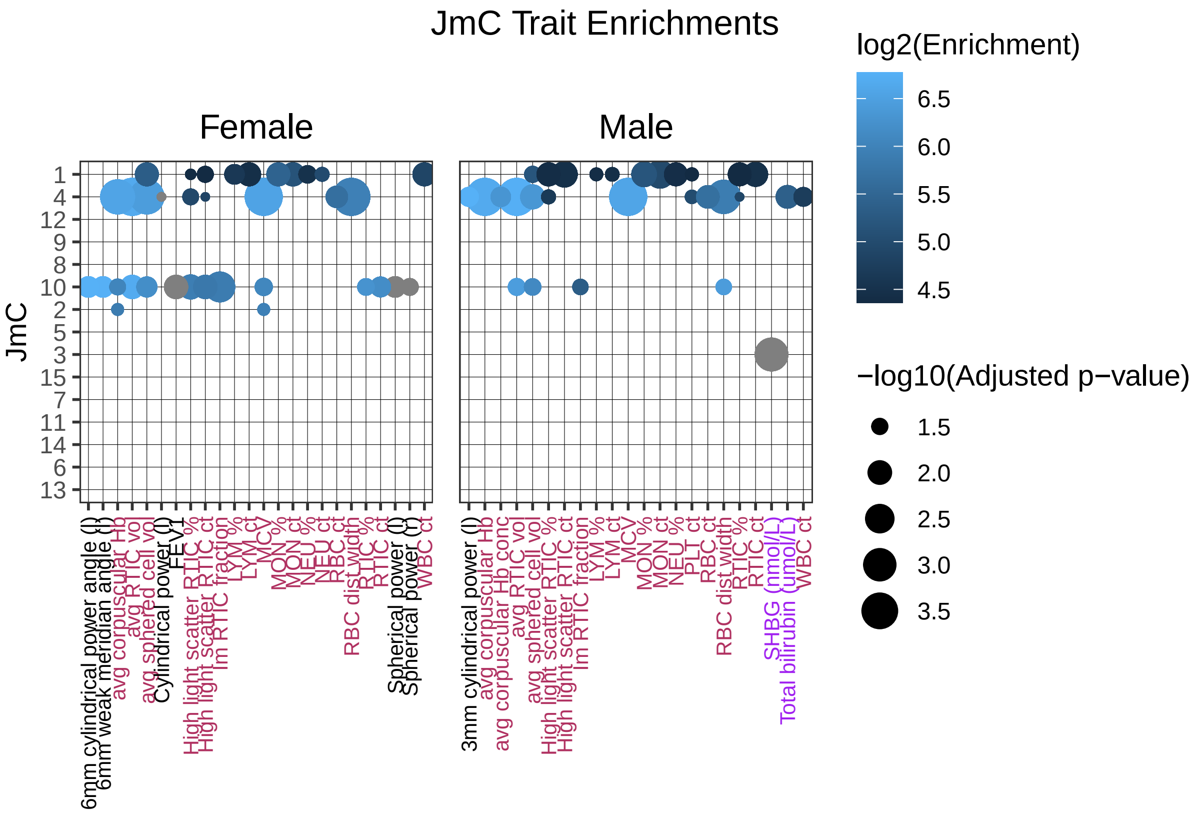


```
ggsave("~/mccoyLab/VISION/sldsc/jmc_male_straight.pdf", plot = gm)
```

```
## Saving 6.5 x 4.5 in image

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning: Removed 4515 rows containing missing values (geom_point).
```

```
library(patchwork)

combo <- gf + theme(legend.position = "none") + gm + theme(axis.text.y = element_blank(),
                                                            axis.ticks.y = element_blank(),
                                                            axis.title.y = element_blank() ) + plot_an
combo
```

```
## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning: Removed 4875 rows containing missing values (geom_point).

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning: Removed 4515 rows containing missing values (geom_point).
```



```
ggsave("~/mccoyLab/VISION/sldsc/jmc_combo_straight.pdf", plot=combo, width= 12.23, height = 6.15, units
```

```
## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning: Removed 4875 rows containing missing values (geom_point).

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning in FUN(X[[i]], ...): NaNs produced

## Warning: Removed 4515 rows containing missing values (geom_point).
```