

MaskRCNN_Model_Validation.rmd

2023-12-12

```
### set up
library(tidyverse)
```

```
## — Attaching core tidyverse packages — tidyverse 2.0.0 —
## ✓ dplyr      1.1.4      ✓ readr      2.1.4
## ✓ forcats    1.0.0      ✓ stringr    1.5.1
## ✓ ggplot2    3.4.4      ✓ tibble     3.2.1
## ✓ lubridate  1.9.3      ✓ tidyr      1.3.0
## ✓ purrr      1.0.2
## — Conflicts — tidyverse_conflicts() —
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()     masks stats::lag()
## ⓘ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
```

```
library(ggpubr)
```

```
# Read in the hand-traced data for the GC and SC, extract from COCO.Json file
df_gc_train <- read_csv('./Train_valid_GCarea2.csv') # GC Hand-traced data
```

```
## Rows: 82 Columns: 5
## — Column specification —
## Delimiter: ","
## chr (1): Genotype
## dbl (4): gc_area, gc_length, gc_width, gc_coco_area
##
## ⓘ Use `spec()` to retrieve the full column specification for this data.
## ⓘ Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
df_sc_train <- read_csv('./Train_valid_SCarea2.csv') # SC Hand-traced data
```

```
## Rows: 164 Columns: 5
## — Column specification —————
## Delimiter: ","
## chr (1): Genotype
## dbl (4): sc_area, sc_length, sc_width, sc_coco_area
##
## 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.
```

```
# Read in the model-predicted data for the GC and SC
df_gc <- read_csv('./Predicted_valid_GCArea2.csv')    # GC Model-predicted data
```

```
## Rows: 83 Columns: 5
## — Column specification —————
## Delimiter: ","
## chr (2): Genotype, date
## dbl (3): gc_area, gc_length, gc_width
##
## 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.
```

```
df_sc <- read_csv('./Predicted_valid_SCArea2.csv')    # SC Model-predicted data
```

```
## Rows: 165 Columns: 5
## — Column specification —————
## Delimiter: ","
## chr (2): Genotype, date
## dbl (3): sc_area, sc_length, sc_width
##
## 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.
```

Merge the GC and SC datasets and indicate the source of each data type

```

df_sc_train.2 <- df_sc_train %>%
  select(c("Genotype", "sc_length", "sc_width", "sc_area")) %>%
  mutate(source = "hand_trace")
df_sc.2 <- df_sc %>%
  select(c("Genotype", "sc_length", "sc_width", "sc_area")) %>%
  mutate(source = "prediction")

SC_merged_hand_trace_prediction <- rbind(df_sc_train.2, df_sc.2)

df_gc_train.2 <- df_gc_train %>%
  mutate(gc_width = gc_width/2) %>% ###GC identified as a complex
  select(c("Genotype", "gc_length", "gc_width", "gc_area")) %>%
  mutate(source = "hand_trace")
df_gc.2 <- df_gc %>%
  mutate(gc_width = gc_width/2) %>%
  select(c("Genotype", "gc_length", "gc_width", "gc_area")) %>%
  mutate(source = "prediction")

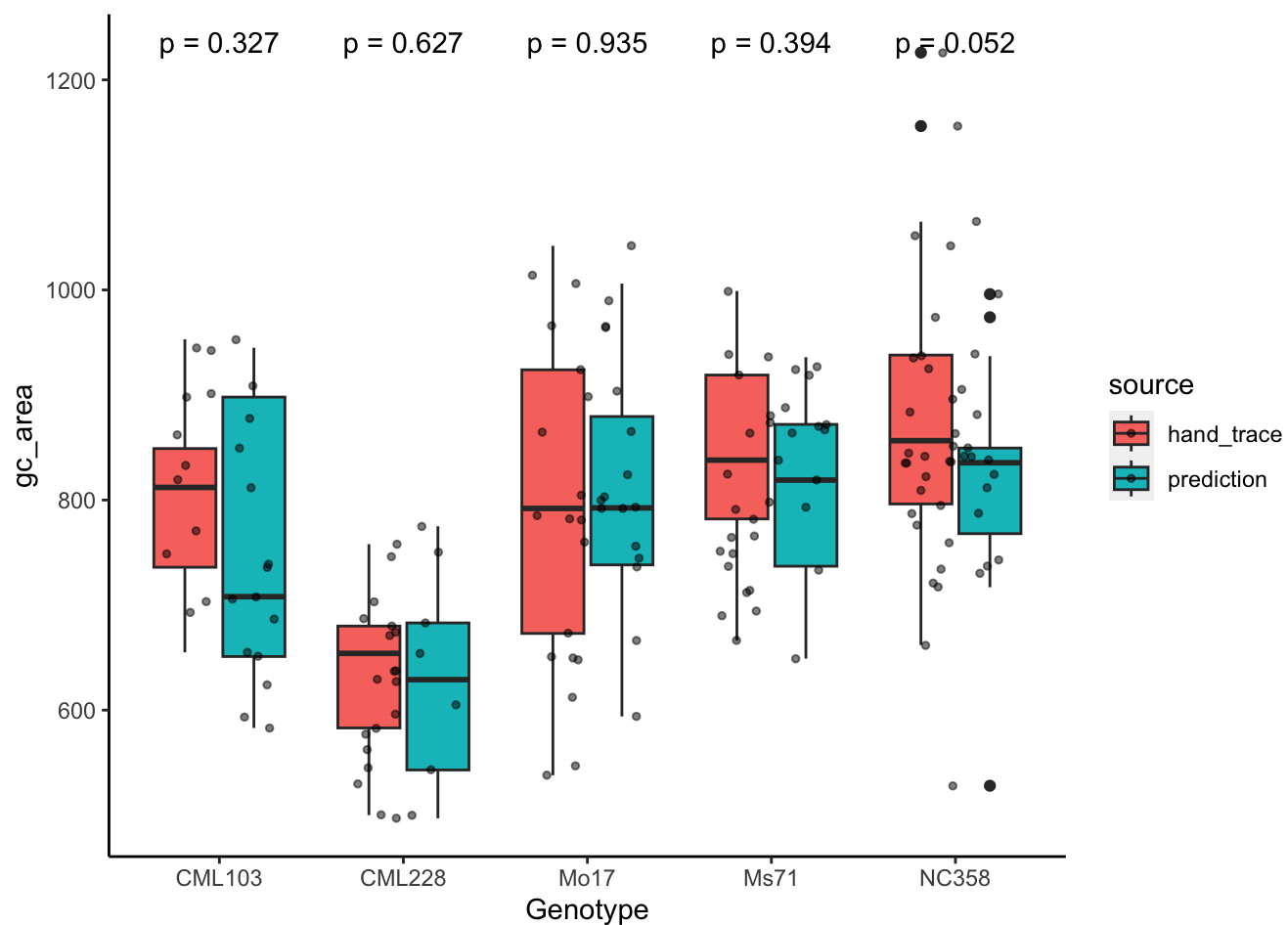
```

###Visulization

```

GC_merged_hand_trace_prediction <- rbind(df_gc_train.2, df_gc.2)
PGC_area <- ggplot(GC_merged_hand_trace_prediction, aes(x = Genotype, y = gc_area, fill = source)) +
  geom_boxplot() +
  geom_jitter(position = position_jitter(width = 0.3), size = 1, alpha = 0.5) + # Add jittered points
  stat_compare_means(method = "t.test", label = "p.format") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"))
# Print the plot
print(PGC_area)

```



```
PSC_area <- ggplot(SC_merged_hand_trace_prediction, aes(x = Genotype, y = sc_area, fill = source)) +
  geom_boxplot() +
  geom_jitter(position = position_jitter(width = 0.3), size = 1, alpha = 0.5) + # Add jittered points
  stat_compare_means(method = "t.test", label = "p.format") +
  theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black"))
# Print the plot
print(PSC_area)
```

