

09.23.2024 Gcamp Analysis for FB Video Aligned

Calvin Cho

2025-03-18

```
library(ggplot2)
```

Warning: package 'ggplot2' was built under R version 4.3.1

```
library(tidyr)
```

Warning: package 'tidyr' was built under R version 4.3.1

```
library(dplyr)
```

Warning: package 'dplyr' was built under R version 4.3.1

Attaching package: 'dplyr'

The following objects are masked from 'package:stats':

filter, lag

The following objects are masked from 'package:base':

intersect, setdiff, setequal, union

```

data_files <- list.files("video", full.names = TRUE)

data_list <- lapply(data_files, read.csv)
names(data_list) <- gsub(".csv", "", basename(data_files))

process_dataset <- function(data, dataset_name) {
  library(dplyr)

  filtered_data <- data[, seq(1, ncol(data), by = 2)]

  if (!"x" %in% names(filtered_data)) {
    names(filtered_data)[1] <- "x"
  }

  neuron_columns <- grep("^RawIntDen", names(filtered_data), value = TRUE)
  new_names <- paste0("Neuron ", seq_along(neuron_columns))
  names(filtered_data)[names(filtered_data) %in% neuron_columns] <- new_names

  if (10 %in% filtered_data$x) {
    reference_values <- filtered_data %>%
      filter(x == 10) %>%
      select(all_of(new_names)) %>%
      unlist(use.names = FALSE)

    if (length(reference_values) == length(new_names)) {
      names(reference_values) <- new_names

      filtered_data <- filtered_data %>%
        mutate(across(all_of(new_names), ~
          (. - reference_values[cur_column()]) / reference_values[cur_column()])))
    } else {
      warning("Reference values for x == 10 are inconsistent with neuron columns.
        Skipping normalization.")
    }
  }

  total_average_value <- filtered_data %>%
    mutate(average = rowMeans(select(., all_of(new_names)), na.rm = TRUE)) %>%
    select(x, average)

  selected_columns <- filtered_data %>%

```

```

    filter(x > 10) %>%
    select(all_of(new_names)) %>%
    select(where(~ any(. > 0.1, na.rm = TRUE)))

conditional_average_value <- filtered_data %>%
  mutate(conditional_average =
    rowMeans(select(., all_of(names(selected_columns))), na.rm = TRUE)) %>%
  select(x, conditional_average)

names(total_average_value)[2] <- paste0(dataset_name, "_average")
names(conditional_average_value)[2] <- paste0(dataset_name, "_caverage")

result <- left_join(total_average_value, conditional_average_value, by = "x")

return(result)
}

processed_data_list <- lapply(names(data_list), function(name) {
  process_dataset(data_list[[name]], name)
})

merged_data <-
  Reduce(function(x, y)
    { merge(x, y, by = "x", all = TRUE) }, processed_data_list)

x_column <- merged_data[, 1, drop = FALSE]

totalaverage_columns <-
totalaverage_columns <- merged_data[, seq(2, ncol(merged_data), by = 2)]
conditionalaverage_columns <- merged_data[, seq(3, ncol(merged_data), by = 2)]

totalaveragedata <- cbind(x_column, totalaverage_columns)
conditionalaveragedata <- cbind(x_column, conditionalaverage_columns)

write.csv(totalaveragedata, "09232024data_total.csv",
  row.names = FALSE)
write.csv(conditionalaveragedata, "09232024_data_conditional.csv",
  row.names = FALSE)

```