

# Plot Benchmark Results

Anthony Duncan

```
library(tidyverse)
library(ggplot2)
library(patchwork)
library(ggpubr)
```

## Figures for LorDist and CTF benchmarks

```
df_ctf = read_delim("results/simulated_new/ctf.tsv")
```

```
## New names:
## Rows: 8000 Columns: 10
## — Column specification
## _____
## _____ Delimiter: "\t" chr
## (2): matrix, output dbl (8): ...1, sample_f, sample_p, subject_f, subject_p, requested_sparsity,...
## 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.
## • `` -> `...1`
```

```
df_lordist = read_delim("results/simulated_new/lordist.tsv")
```

```
## Rows: 8000 Columns: 6
## — Column specification _____
## _____
## Delimiter: "\t"
## chr (1): output
## dbl (5): i, requested_sparsity, achieved_sparsity, pval, fval
##
## 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_bc = read_delim("results/simulated_new/bc.tsv")
```

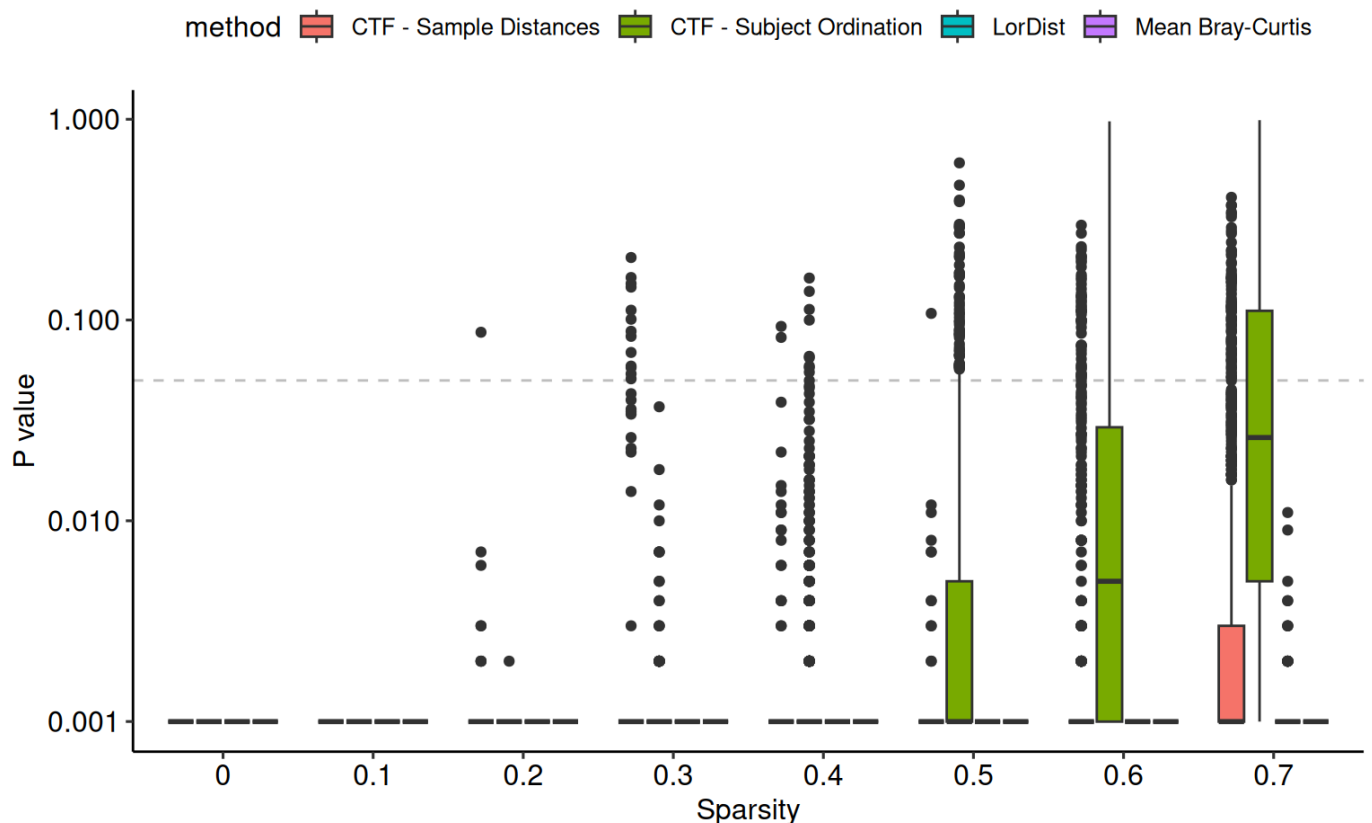
```
## Rows: 8000 Columns: 6
## — Column specification —————
## Delimiter: "\t"
## chr (1): output
## dbl (5): i, requested_sparsity, achieved_sparsity, pval, fval
##
## i Use `spec()` to retrieve the full column specification for this data.
## i Specify the column types or set `show_col_types = FALSE` to quiet this message.
```

```
# Harmonise for plotting
df_ctf_sample_long <- df_ctf |>
  select(sample_f, sample_p, output, requested_sparsity, achieved_sparsity) |>
  rename(pval = sample_p, fval = sample_f) |>
  mutate(method = "CTF - Sample Distances")
df_ctf_subject_long <- df_ctf |>
  select(subject_f, subject_p, output, requested_sparsity, achieved_sparsity) |>
  rename(pval = subject_p, fval = subject_f) |>
  mutate(method = "CTF - Subject Ordination")
df_bc_long <- df_bc |>
  select(fval, pval, output, requested_sparsity, achieved_sparsity) |>
  mutate(method = "Mean Bray-Curtis")
df_lordist_long <- df_lordist |>
  select(fval, pval, output, requested_sparsity, achieved_sparsity) |>
  mutate(method = "LorDist")
df_long <- rbind(
  df_ctf_sample_long, df_ctf_subject_long, df_lordist_long, df_bc_long
) |>
  mutate(requested_sparsity = as.factor(requested_sparsity))
df_long |> head()
```

```
## # A tibble: 6 × 6
##   fval    pval output requested_sparsity achieved_sparsity method
##   <dbl>  <dbl> <chr>          <fct>                <dbl> <chr>
## 1  90.9 0.000999 ./data/simulated_n... 0                0.00422 CTF -...
## 2 160. 0.000999 ./data/simulated_n... 0.1              0.104   CTF -...
## 3  88.0 0.000999 ./data/simulated_n... 0.2              0.203   CTF -...
## 4  81.7 0.000999 ./data/simulated_n... 0.3              0.303   CTF -...
## 5  78.7 0.000999 ./data/simulated_n... 0.4              0.402   CTF -...
## 6  68.1 0.000999 ./data/simulated_n... 0.5              0.502   CTF -...
```

Make boxplots comparing p values

```
plt_p <- ggplot(
  df_long,
  aes(x = requested_sparsity, y = pval, fill = method)
) +
  geom_hline(yintercept = 0.05, linetype = "dashed", color = "grey") +
  xlab("Sparsity") +
  ylab("P value") +
  geom_boxplot() +
  scale_y_log10() +
  theme_pubr()
plt_p
```



Recreate plots from LorDist paper

```
# Point at the mean, error bars are range.
# Caption on original figure is unclear whether point is at mean and what error
# bars mean
df_plot <- df_long |>
  group_by(requested_sparsity, method) |>
  summarise(
    median_pval = median(pval), median_fval = median(fval),
    mean_pval = mean(pval), mean_fval = mean(fval),
    min_fval = min(fval), min_pval = min(pval),
    max_fval = max(fval), max_pval = max(pval),
    quant_fval_0.1 = quantile(fval, 0.1),
    quant_fval_0.9 = quantile(fval, 0.9),
    quant_pval_0.1 = quantile(pval, 0.1),
    quant_pval_0.9 = quantile(pval, 0.9),
    mean_achieved_sparsity = mean(achieved_sparsity)
  )
```

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

```
scale_lordist <- scale_color_manual(
  name = "Method",
  values = c(
    "LorDist" = "#F8766D",
    "CTF - Sample Distances" = "#03408fff",
    "CTF - Subject Ordination" = "#00BFC4",
    "Mean Bray-Curtis" = "#4c5152ff"
  )
)
plt_pval <- df_plot |>
  ggplot(
    aes(
      x = requested_sparsity,
      y = median_pval,
      ymin = quant_pval_0.1,
      ymax = quant_pval_0.9,
      color = method
    )
  ) +
  geom_point(
    size = 2,
    position = position_dodge(width = 0.3)
  ) +
  geom_errorbar(
    width = 0.25,
    position = position_dodge(width = 0.3)
  ) +
  geom_line(
    aes(group = method),
    position = position_dodge(width = 0.3)
  ) +
  scale_lordist +
  xlab("Sparsity") +
  ylab("P value") +
  geom_hline(yintercept = 0.05, color = "grey", linetype = "dashed") +
  theme_pubr()

ggsave("results/simulated/pval.png", plt_pval)
```

```
## Saving 8 x 5 in image
```

```

plt_fval <- df_plot |>
  ggplot(
    aes(
      x = requested_sparsity,
      y = median_fval,
      ymin = quant_fval_0.1,
      ymax = quant_fval_0.9,
      color = method
    )
  ) +
  geom_point(
    size = 2,
    position = position_dodge(width=0.3)
  ) +
  geom_errorbar(
    width = 0.25,
    position = position_dodge(width=0.3)
  ) +
  geom_line(
    aes(group = method),
    position = position_dodge(width=0.3)
  ) +
  scale_x_dist +
  scale_y_log10(n.breaks = 6) +
  xlab("Sparsity") +
  ylab("Pseudo-F statistic") +
  theme_pubr()

ggsave("results/simulated/fval.png", plt_fval)

```

## Saving 8 x 5 in image

```

plt_combined <- plt_fval +
  plt_pval +
  plot_layout(guides = 'collect') &
  theme(legend.position = "top")
ggsave(
  "results/simulated/combined_fval_pval.png",
  plt_combined,
  scale = 1.5,
  height = 3,
  width = 6
)
plt_combined

```

