Exact vs. approximate sceptre

Tim

2023-04-11

Exact vs. `approximate` SCEPTRE

In this writeup I compare the "approximate" version of SCEPTRE to the "exact" version of SCEPTRE (henceforth "SCEPTRE-approximate" and "SCEPTRE-exact"). SCEPTRE-approximate is the version of SCEPTRE that currently is implemented in the `sceptre` package. SCEPTRE-approximate carries out the gene precomputation using the negative control cells only. SCEPTRE-exact, by contrast, carries out the gene precomputation using the *entire* set of cells (for a given gene-gRNA group pair). As Gene pointed out, SCEPTRE-approximate technically is an approximate permutation test, while SCEPTRE-exact is an exact permutation test. SCEPTRE-approximate and SCEPTRE-exact should produce similar p-values when the regression on the control cells matches the regression on the entire set of cells (which should hold under the null hypothesis when the number of cells is large).

To explore this issue empirically, I implemented SCEPTRE-exact in the SCEPTRE package. Note that this is an experimental feature that I implemented on a development branch; "properly" integrating the "exact full" test statistic into the `sceptre` software would take about a day.

```
 # load packages
library(ondisc)
library(lowmoi)
library(ggplot2)
```

Below, I define a few functions to carry out the analysis. I sample 5,000 discovery pairs from a given dataset and apply SCEPTRE-exact and SCEPTRE-approximate to analyze the pairs.

```
# write a function to carry out the analysis for a given dataset
dirs <- c("data/papalexi/eccite_screen/",
          "data/schraivogel/enhancer_screen_chr8/")

sample_discovery_pairs <- function(response_odm, grna_odm, n_to_sample) {
  set.seed(3)
  grna_groups <- grna_odm |>
              ondisc::get_feature_covariates() |>
              dplyr::pull(target) |> unique()
  grna_groups <- grna_groups[grna_groups != "non-targeting"]
  expand.grid(response_id = response_odm |> ondisc::get_feature_ids(),
              grna_group = grna_groups) |>
  dplyr::sample_n(n_to_sample)
}

run_analysis <- function(dir, n_to_sample = 5000) {
  # 1. load data
  LOCAL_SCEPTRE2_DATA_DIR <- .get_config_path("LOCAL_SCEPTRE2_DATA_DIR")
  data_dir <- paste0(LOCAL_SCEPTRE2_DATA_DIR, dir)

  # 2. response info
  response_odm_fp <- paste0(data_dir, "gene/matrix.odm")
  response_metadata_fp <- paste0(data_dir, "gene/metadata_qc.rds")
  response_odm <- read_odm(odm_fp = response_odm_fp, metadata_fp = response_metadata_fp)

  # 3. grna info
  grna_odm_fp <- paste0(data_dir, "grna_assignment/matrix.odm")
  grna_metadata_fp <- paste0(data_dir, "grna_assignment/metadata_qc.rds")
  grna_odm <- read_odm(odm_fp = grna_odm_fp, metadata_fp = grna_metadata_fp)

  # 4. discovery pairs to analyze
  response_grna_group_pairs <- sample_discovery_pairs(response_odm = response_odm,
                                                      grna_odm = grna_odm,
                                                      n_to_sample = n_to_sample)

  approx_time <- system.time(approx_res <- lowmoi::sceptre(response_odm = response_odm,
                                                           grna_odm = grna_odm,
                                                           response_grna_group_pairs = re
sponse_grna_group_pairs,
                                                           test_stat = "full",
                                                           print_progress = FALSE))

  exact_time <- system.time(exact_res <- lowmoi::sceptre(response_odm = response_odm,
                                                         grna_odm = grna_odm,
                                                         response_grna_group_pairs = respo
nse_grna_group_pairs,
                                                         test_stat = "exact_full",
                                                         print_progress = FALSE))
  # 5. join the data frames
  res <- dplyr::left_join(x = approx_res,
                          y = exact_res,
                          by = c("response_id", "grna_group"),
```

```
                          suffix = c("_approx", "_exact")) |> na.omit()

  # 6. output the result
  list(res = res, approx_time = approx_time, exact_time = exact_time)
}

plot_analysis_output <- function(res) {
  p1 <- ggplot(data = res, mapping = aes(x = p_value_exact, y = p_value_approx)) +
    geom_point() + theme_bw() + xlab("P-value (exact)") + ylab("P-value (approx)") +
    ggtitle("Untransformed scale") + geom_abline(slope = 1, intercept = 0, col = "blue")
  p2 <- ggplot(data = res, mapping = aes(x = p_value_exact, y = p_value_approx)) +
    geom_point() + theme_bw() + xlab("P-value (exact)") + ylab("P-value (approx)") +
    ggtitle("Transformed scale") + ggplot2::scale_x_continuous(trans = sceptre:::revlog_
trans(10)) +
        ggplot2::scale_y_continuous(trans = sceptre:::revlog_trans(10)) +
    geom_abline(slope = 1, intercept = 0, col = "blue")
  return(cowplot::plot_grid(p1, p2, nrow = 1))
}
```

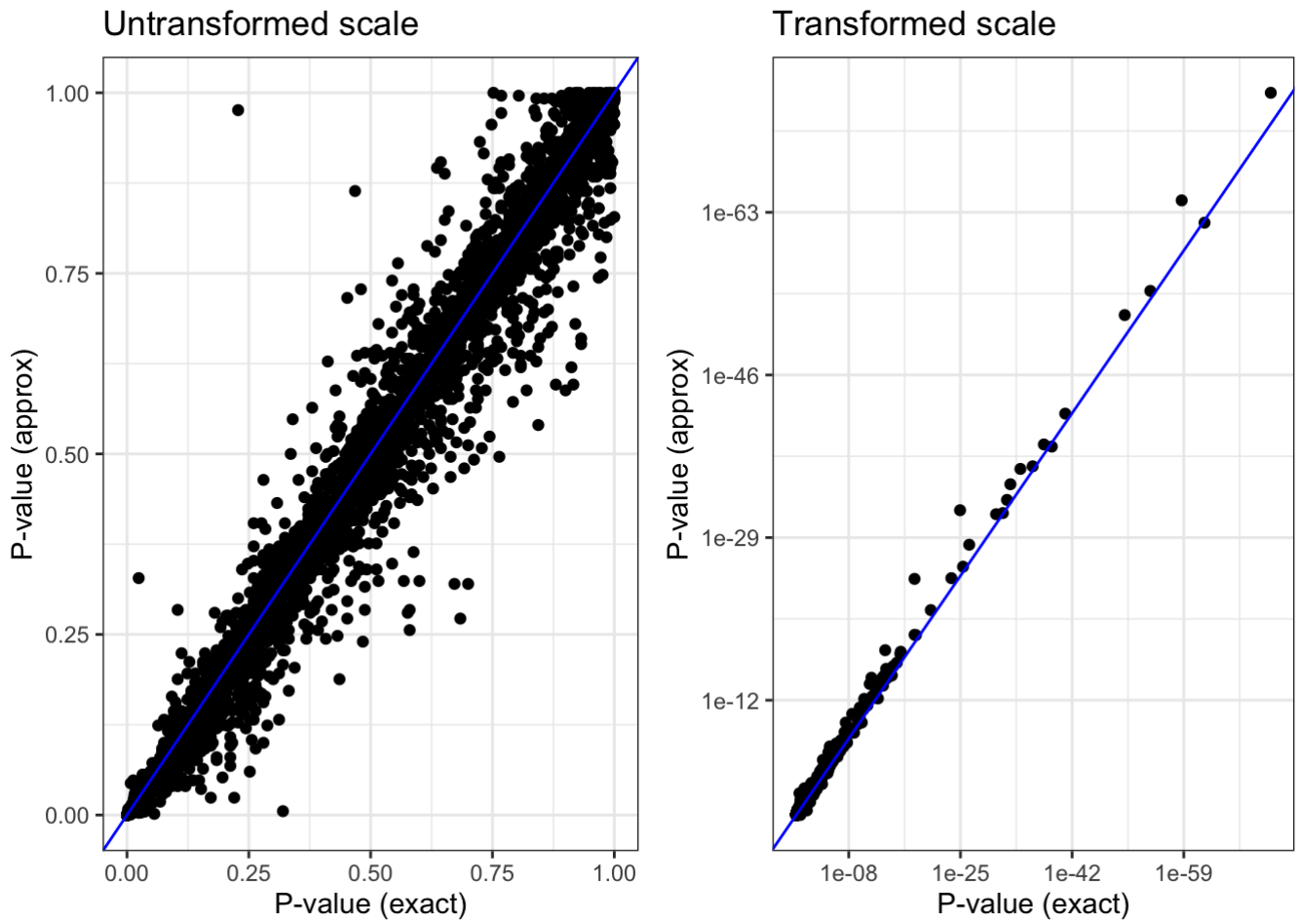
First, I analyze the Papalexi data.

```
# papalexi
out <- run_analysis("data/papalexi/eccite_screen/")
```

```
## Running setup.  ✓
## Generating permutation resamples. ✓
## Running differential expression analyses.
## Running setup.  ✓
## Generating permutation resamples. ✓
## Running differential expression analyses.
```

I plot the SCEPTRE-approximate vs. SCEPTRE-exact p-values on an untransformed scale and negative log-10 transformed scale.

```
plot_analysis_output(out$res)
```



We see that the p-values coincide, especially in the tail. The SCEPTRE-approximate p-values tend to be slightly smaller (i.e., more significant) than their SCEPTRE-exact counterparts. The correlation between the p-values high high on both scales.

```
cor(out$res$p_value_approx, out$res$p_value_exact)
```

```
## [1] 0.9854812
```

```
cor(-log(out$res$p_value_approx, base = 10),
    -log(out$res$p_value_exact, base = 10))
```

```
## [1] 0.997687
```

Finally, I compare the execution time of SCEPTRE-exact and SCEPTRE-approximate.

```
out$exact_time[[3]]/out$approx_time[[3]]
```

```
## [1] 1.179087
```

SCEPTRE-exact is about 10% slower than SCEPTRE-approximate, which is not bad at all.

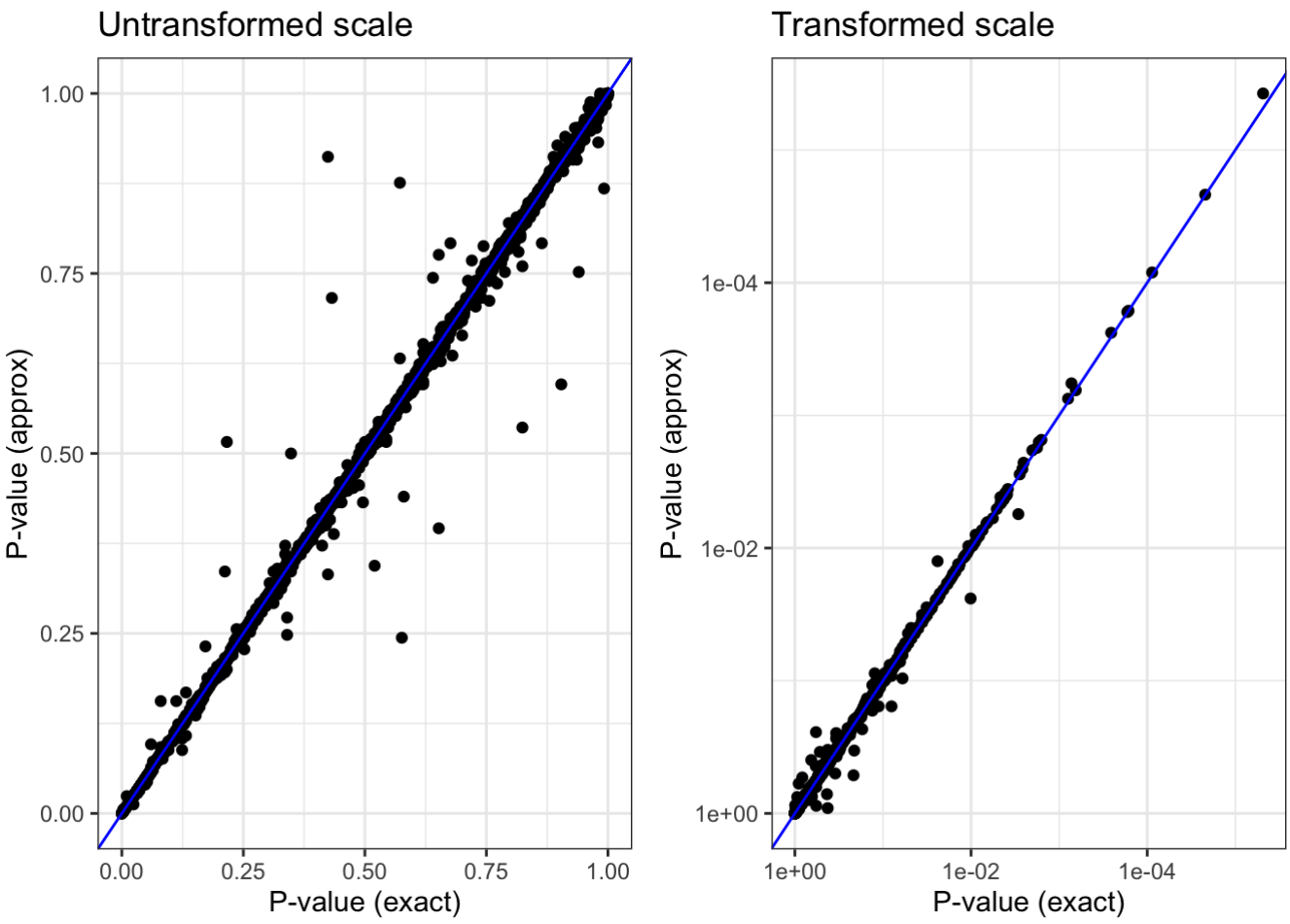
I repeat this analysis for the Frangieh IFN-gamma data.

```
# frangieh
out <- run_analysis("data/frangieh/ifn_gamma/")
```

```
## Running setup.  ✓
## Generating permutation resamples. ✓
## Running differential expression analyses.
## Running setup.  ✓
## Generating permutation resamples. ✓
## Running differential expression analyses.
```

Plotting the p-values, we again see that they coincide closely, especially in the tail.

```
plot_analysis_output(out$res |>
                       dplyr::filter(p_value_approx > 1e-50))
```




```
out$exact_time[[3]]/out$approx_time[[3]]
```

```
## [1] 1.173328
```

SCEPTRE-exact is within 10% of the running time of SCEPTRE-approximate.

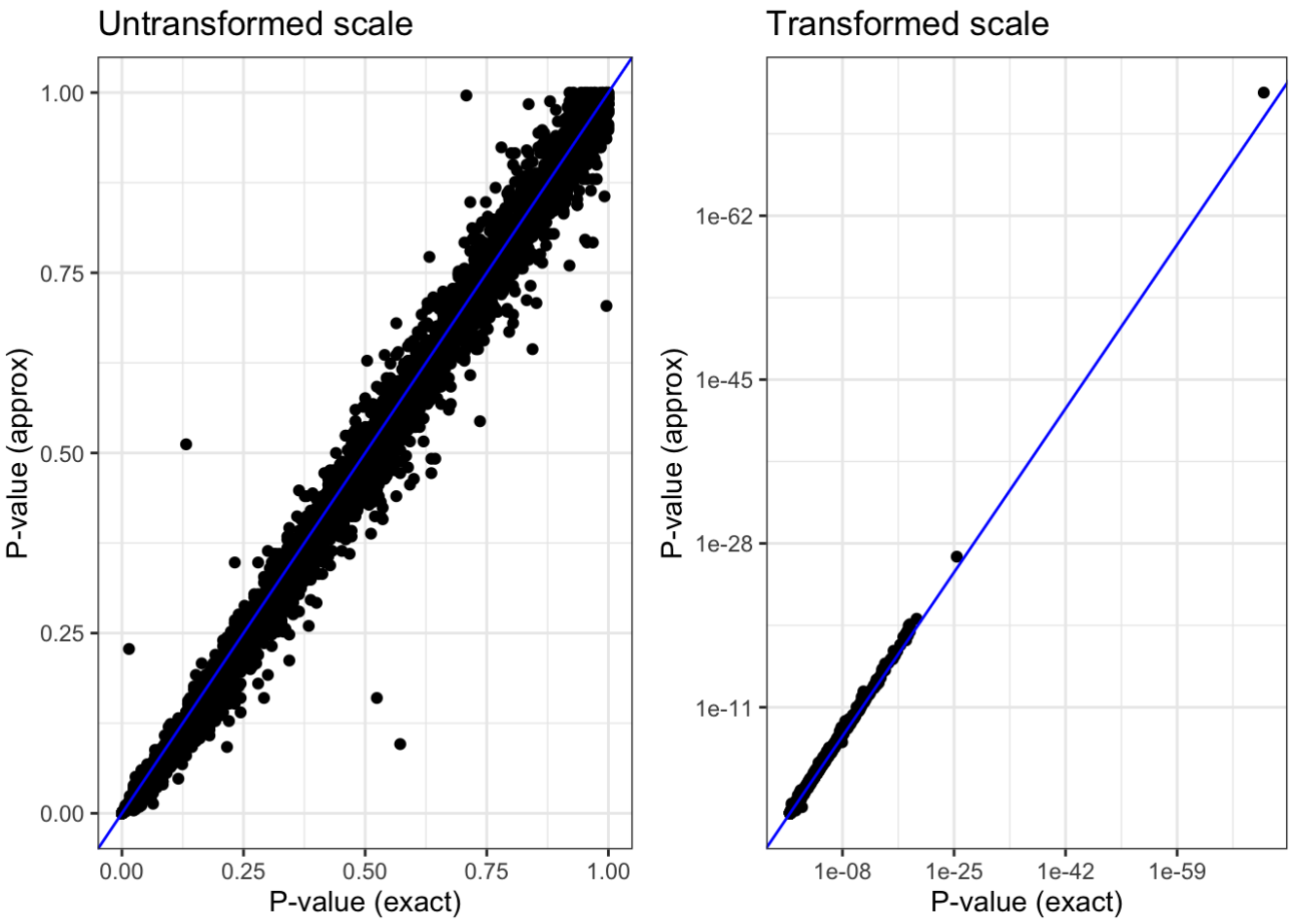
Finally, I carry out this analysis on the Schraivogel chromosome 8 data. I sample twice as many pairs so that we can look further out into the tail.

```
# schraivogel
out <- run_analysis("data/schraivogel/enhancer_screen_chr8/", n_to_sample = 10000)
```

```
## Running setup.  ✓
## Generating permutation resamples. ✓
## Running differential expression analyses.
## Running setup.  ✓
## Generating permutation resamples. ✓
## Running differential expression analyses.
```

The p-values again coincide. The correlation between the two sets of p-values on the negative log-10 transformed scale is 0.999.

```
plot_analysis_output(out$res)
```




```
cor(-log(out$res$p_value_approx),
    -log(out$res$p_value_exact))
```

```
## [1] 0.9993662
```

SCEPTRE-exact is about 7.5 times slower than SCEPTRE-approximate. SCEPTRE-exact is relatively slow on this dataset because each gene is paired to a relatively large number of gRNA groups. Thus, we save more compute by factoring out the precomputation at the level of the gene.

In summary SCEPTRE-exact and SCEPTRE-approximate do not seem to differ too much statistically. We should decide whether we want to include SCEPTRE-exact as an option in the package. Adding this functionality would take about a day of work.