Exact vs. approximate sceptre

Tim

2023-04-11

Comparing exact SCEPTRE and 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 yields a similar result to 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 this functionality 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
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, formula_obj = NULL) {
  # 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")
```

```r
  response_odm <- read_odm(odm_fp = response_odm_fp, metadata_fp = response_metadata_fp)
  if (!is.null(formula_obj)) {
    response_odm@misc$sceptre_formula <- formula_obj
  }

  # 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 = response_grna_gr
                                                           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 = response_grna_grou
                                                         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) {
  res <- res |> dplyr::mutate(p_value_approx = ifelse(p_value_approx == 0, 1e-200, p_value_approx))
  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") + ggplot2::scal
  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") + scale_x_continuous(trans = sceptre:::revlog_trans(10)) + scale_y_con
    geom_abline(slope = 1, intercept = 0, col = "blue")
  return(cowplot::plot_grid(p1, p2, nrow = 1))
}
```

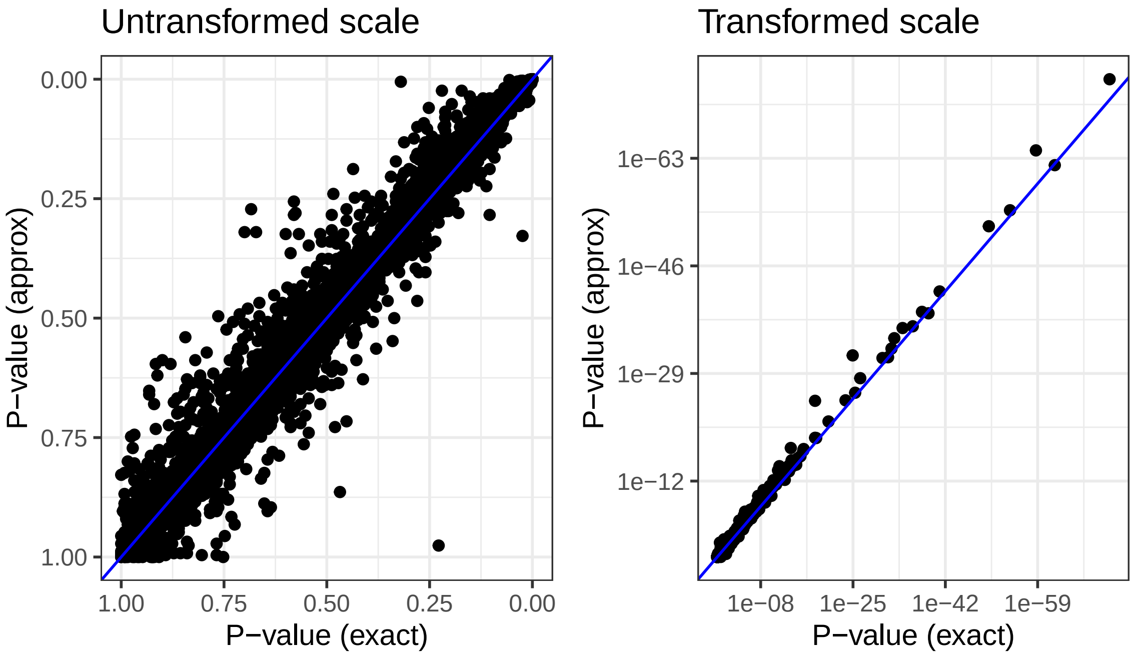
Papalexi

First, I analyze the Papalexi data.

```r
# papalexi
fp <- "papa_out.rds"
if (!file.exists("papa_out.rds")) {
  papa_out <- run_analysis("data/papalexi/eccite_screen/")
  saveRDS(papa_out, file = fp)
} else {
  papa_out <- readRDS(fp)
}
```

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

```r
plot_analysis_output(papa_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 is high on both scales.

```r
cor(papa_out$res$p_value_approx, papa_out$res$p_value_exact)
```

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

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

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

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

```r
papa_out$exact_time[[3]]/papa_out$approx_time[[3]]
```

```
## [1] 0.9632643
```

SCEPTRE-exact has a similar running time to SCEPTRE-approximate.

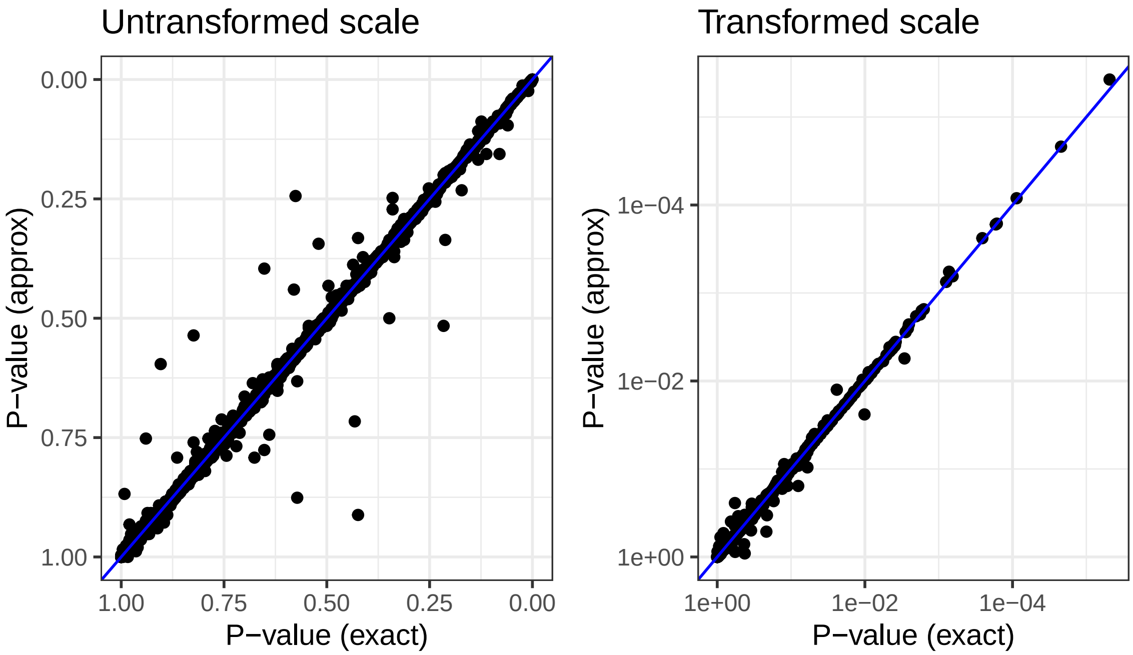
Frangieh IFN-gamma

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

```
# frangieh
fp <- "frangieh_out.rds"
if (!file.exists(fp)) {
 frangieh_out <- run_analysis("data/frangieh/ifn_gamma/")
 saveRDS(object = frangieh_out, file = fp)
} else {
  frangieh_out <- readRDS(fp)
}
```

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

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



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

```
## [1] 1.074138
```

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

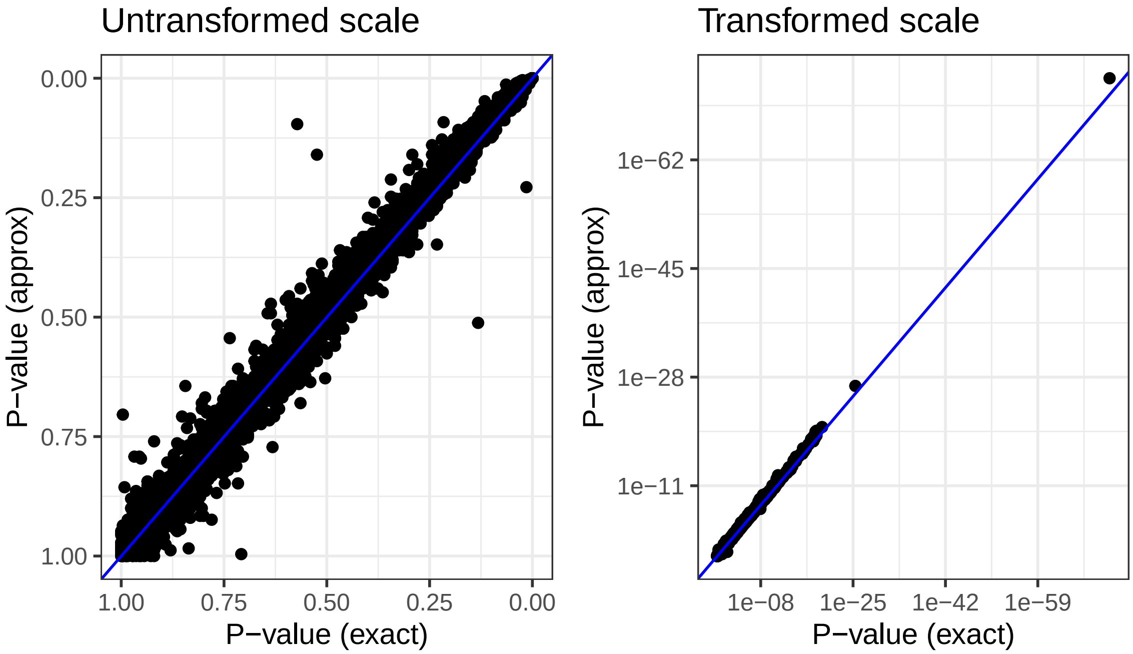
Schraivogel: batch excluded as covariate

Next, I carry out this analysis on the Schraivogel chromosome 8 data. First, I apply SCEPTRE-approximate and SCEPTRE-exact to analyze the data with batch *excluded* as a covariate. I sample twice as many pairs so that we can look further out into the tail.

```
# schraivogel
fp <- "schraivogel_out_no_batch.rds"
if (!file.exists(fp)) {
  schraivogel_out_no_batch <- run_analysis("data/schraivogel/enhancer_screen_chr8/",
                                           n_to_sample = 10000)
  saveRDS(object = schraivogel_out_no_batch, file = fp)
} else {
 schraivogel_out_no_batch <- readRDS(fp)
}
```

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(schraivogel_out_no_batch$res)
```



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

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

SCEPTRE-exact is about 8 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.

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

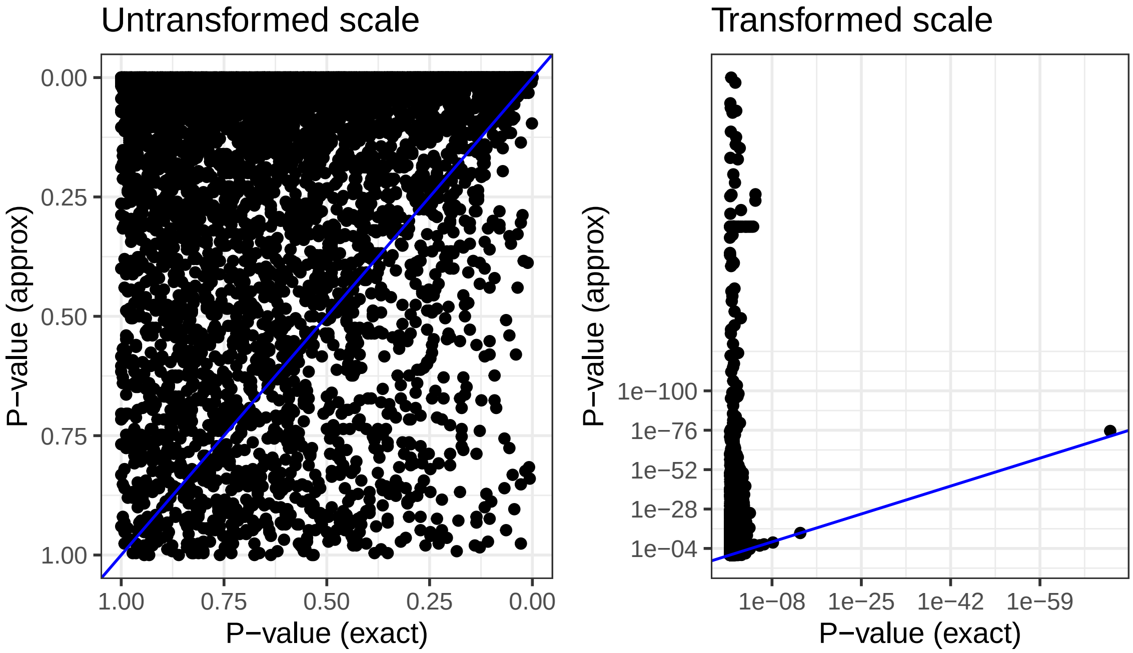
```
## [1] 7.973644
```

Schraivogel: batch included as covariate

Finally, I apply SCEPTRE-exact and SCEPTRE-approximate to the Schraivogel chromosome 8 data with batch *included* as a covariate.

```
# schraivogel
fp <- "schraivogel_out_with_batch.rds"
if (!file.exists(fp)) {
  schraivogel_out_with_batch <- run_analysis("data/schraivogel/enhancer_screen_chr8/",
                                             n_to_sample = 10000,
                                             formula_obj = formula(~log(response_n_umis) + log(response_n
  saveRDS(object = schraivogel_out_with_batch, file = fp)
} else {
 schraivogel_out_with_batch <- readRDS(fp)
}
```

```
plot_analysis_output(schraivogel_out_with_batch$res)
```



Whoah! The results diverge substantially. In the bulk the two sets of p-values are weakly correlated.

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

```
## [1] 0.2385387
```

And in the tail the approximate p-values are pretty much uniformly smaller than the bulk p-values. SCEPTRE-exact is about three times slower than SCEPTRE-approximate in this setting.

```
schraivogel_out_with_batch$exact_time/schraivogel_out_with_batch$approx_time
```

```
##     user   system  elapsed 
## 2.786417 7.311983 2.958268
```

Conclusion

SCEPTRE-exact and SCEPTRE-approximate yield similar p-values on the Papalexi data, Frangieh data, and Schraivogel data (when batch is excluded as a covariate). However, on the Schraivogel data with batch included as a covariate, the two methods diverge. As Gene pointed out, including batch as a covariate on the Schraivogel data causes the regression on the entire set of cells to diverge from the regression on the negative control cells, which presumably explains the discrepancy. We will need to determine how to update the `sceptre` method/package in light of these results.