Evaluating several methods on the Schraivogel discovery pairs

Tim

2023-04-10

I apply several methods to a subset of the Schraivogel discovery pairs: Schraivogel method, Seurat DE, and NB regression. NB regression is applied *without* batch as a covariate. I also apply four variants of SCEPTRE to the data: exact vs. approximate, and with vs. without batch included as a covariate. I notate these four variants as follows: SCEPTRE-exact-with-batch, SCEPTRE-exact-no-batch, SCEPTRE-approximate-with-batch, and SCEPTRE-approximate-no-batch.

I load the SCEPTRE results that Gene generated on the Schraivogel discovery pairs.

```
schraivogel_chr8_result_dir <- paste0(LOCAL_SCEPTRE2_DATA_DIR,
                                      "results/schraivogel_analysis/")
gk_sceptre_schraivogel_res <- readRDS(paste0(schraivogel_chr8_result_dir,
                                             "sceptre_schraivogel_chr_8_results.rds"))
gk_orig_schraivogel_res <- readRDS(paste0(schraivogel_chr8_result_dir,
                                          "schraivogel_schraivogel_chr_8_results.rds"))
```

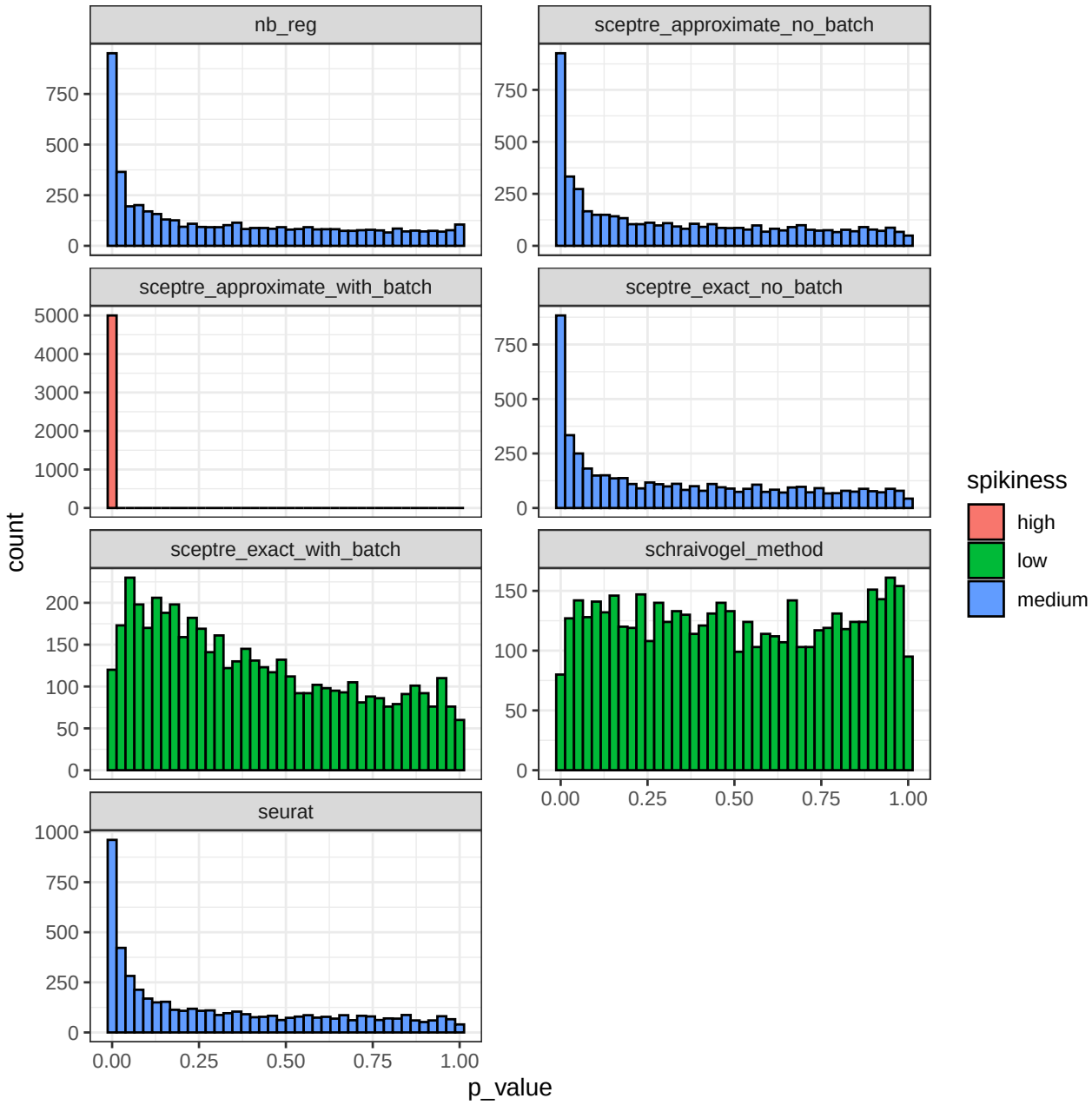
Many of the p-values are small. In fact, about 50% of the p-values are less than 0.001, which seems a bit strange. Let us randomly sample 5,000 of the pairs for which SCEPTRE produces a p-value below 0.001.

```
pairs_to_analyze <- gk_sceptre_schraivogel_res |>
   dplyr::filter(p_value < 0.001) |>
   dplyr::sample_n(5000) |>
   dplyr::select(response_id, grna_group)
```

I subject these pairs to NB regression, Seurat DE, SCEPTRE-exact-with-batch, SCEPTRE-exact-no-batch, SCEPTRE-approximate-with-batch, and SCEPTRE-approximate-no-batch. These methods (especially Seurat DE) take some time to run. I use the original Schraivogel results for the Schraivogel method.

I plot a histogram of the p-values outputted by each method. I color the histogram by "degree of spikiness" (as determined by me): high, medium, or low. SCEPTRE-approximate-with-batch is the spikiest near zero. Next, NB regression, SCEPTRE-approximate-no-batch, SCEPTRE-exact-no-batch, Schraivogel Method, and Seurat exhibit intermediate spikiness. Finally, SCEPTRE-exact-with-batch shows a low degree of spikiness. Still, the p-value distributions of SCEPTRE-exact-with-batch and Schraivogel method are rather dissimilar.

```
ggplot(combined_res, mapping = aes(x = p_value, fill = spikiness)) +
  facet_wrap(.~method, scales = "free_y", nrow = 4) +
  geom_histogram(bins = 40, col = "black") + theme_bw()
```



SCEPTRE-approximate-with-batch likely is pathological. NB regression, SCEPTRE-approximate-no-batch, SCEPTRE-exact-no-batch, and Seurat produce roughly similar p-value distributions. Meanwhile, SCEPTRE-exact-with-batch and Schraivogel-method are more uniform. SCEPTRE-exact-with-batch is the only method (to my knowledge) that attempts to adjust for batch effects (aside from SCEPTRE-approximate-with-batch, which is pathological). This likely explains why SCEPTRE-exact-with-batch is more uniform than SCEPTRE-exact-no-batch, Seurat, and NB regression.

The Schraivogel data likely contain batch effects. We had not picked up on these batch effects in the undercover analysis because nearly all NT gRNAs are in the same batch! However, batch effects seem to be rearing their head in the discovery analysis, leading to inflation. For example, consider the gene "MRPL13." SCEPTRE-exact-no-batch indicates that MRPL13 is associated with every single gRNA group tested except for one (at FWER level 0.1). As Gene has noted, it is biologically implausible that that a gene would be regulated by such a large number of gRNAs, so these results likely are an artifact of inflation (rather than reflective of real biology).

```
sceptre_no_batch_mrpl13_p <- sceptre_exact_no_batch_res |>
  dplyr::filter(response_id == "MRPL13") |>
  dplyr::pull(p_value)
sum(sceptre_no_batch_mrpl13_p < .1/nrow(sceptre_exact_no_batch_res))
```

```
## [1] 67
```

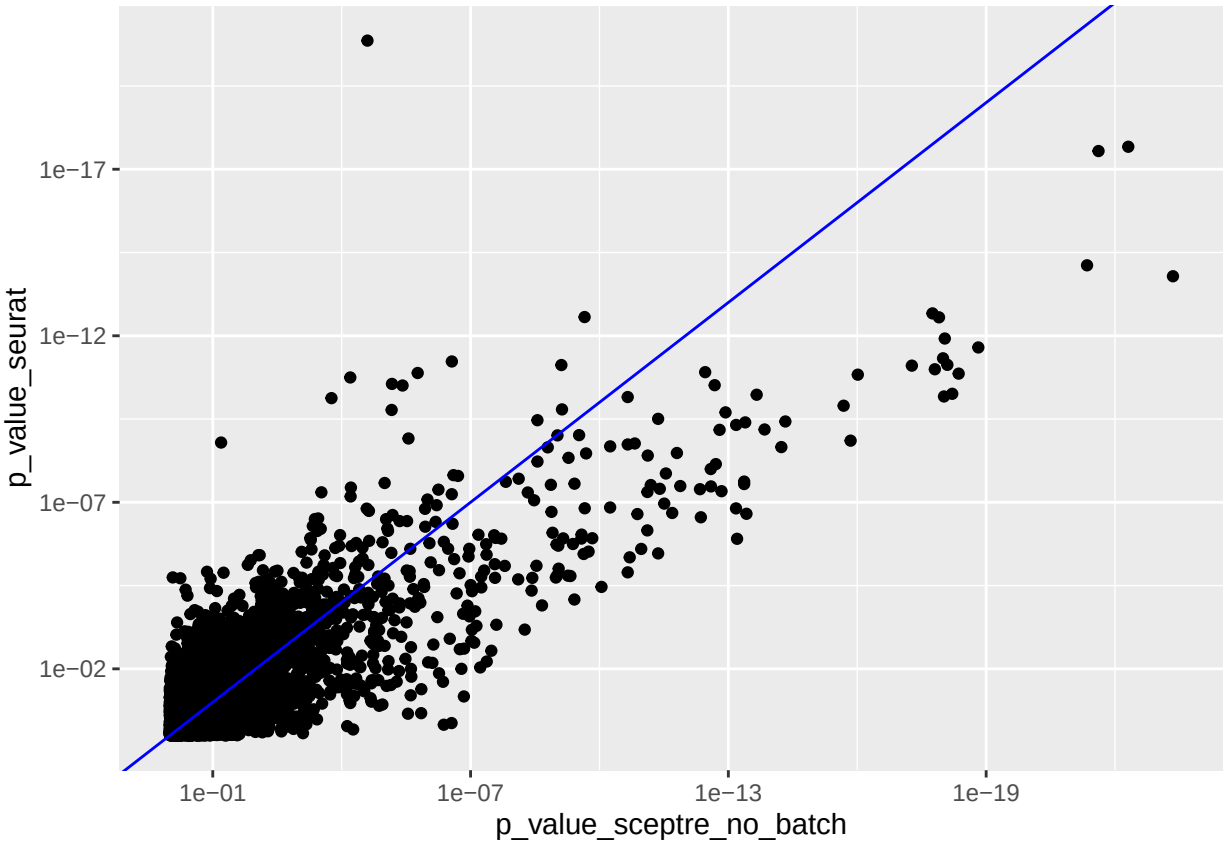
Interestingly, Seurat also indicates that MRPL13 is associated with every gRNA group except for one. Thus, Seurat likely also is miscalibrated on the discovery pairs.

```
seurat_mrpl13_p <- sceptre_exact_no_batch_res |>
  dplyr::filter(response_id == "MRPL13") |>
  dplyr::pull(p_value)
sum(seurat_mrpl13_p < .1/nrow(sceptre_exact_no_batch_res))
```

```
## [1] 67
```

In fact, the correlation between the SCEPTRE-exact-no-batch p-values and Seurat p-values is decent.

```
to_plot <- dplyr::left_join(sceptre_exact_no_batch_res,
                            seurat_res, by = c("response_id", "grna_group"),
                            suffix = c("_sceptre_no_batch", "_seurat"))
ggplot(data = to_plot |> dplyr::filter(p_value_sceptre_no_batch > 1e-30),
       mapping = aes(x =p_value_sceptre_no_batch , y = p_value_seurat)) +
  geom_point() + scale_x_continuous(trans = sceptre:::revlog_trans(10)) +
  scale_y_continuous(trans = sceptre:::revlog_trans(10)) +
  geom_abline(slope = 1, intercept = 0, col = "blue")
```



SCEPTRE-exact-with-batch, by contrast, indicates that MRPL13 is associated with *no* gRNA group tested

(after a Bonferroni correction at level 0.1). This result seems much more biologically plausible.

```
sceptre_no_batch_mrpl13_p <- sceptre_exact_with_batch_res |>
  dplyr::filter(response_id == "MRPL13") |> dplyr::pull(p_value)
sum(sceptre_no_batch_mrpl13_p < .1/nrow(sceptre_exact_no_batch_res))
```

```
## [1] 0
```

The Schraivogel method agrees with SCEPTRE in this regard.

```
schraivogel_mrpl13_p <- schraivogel_res |>
   dplyr::filter(response_id == "MRPL13") |>
  dplyr::pull(p_value)
sum(schraivogel_mrpl13_p < .1/nrow(sceptre_exact_no_batch_res))
```

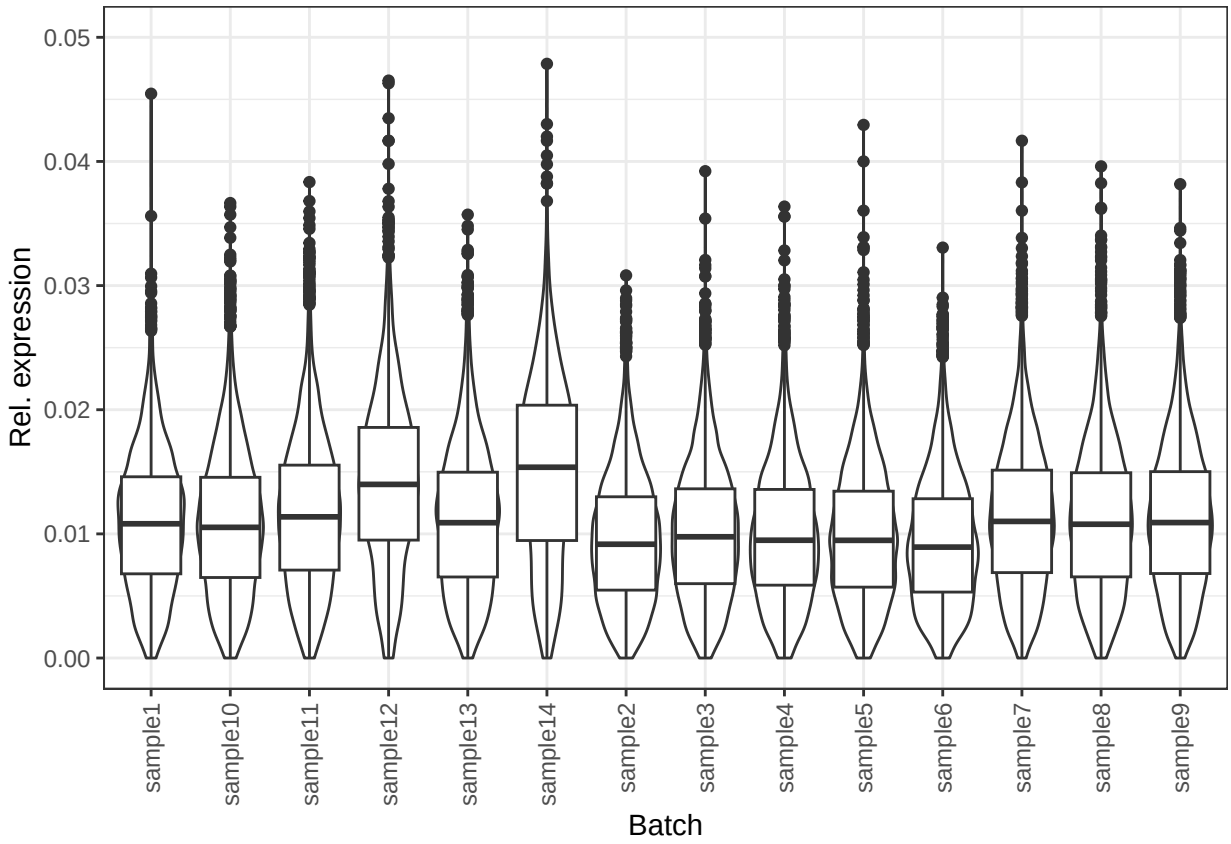
```
## [1] 0
```

These preliminary results indicate that we might be able to make a positive comparison between SCEPTRE and Seurat and the Schraivogel data (in favor of SCEPTRE).

I check to see whether the (relative) expression of MRPL13 varies across batch.

```
exp <- response_odm[["MRPL13",]] |> as.numeric()
cell_covariates <- response_odm |> ondisc::get_cell_covariates()

df <- data.frame( rel_exp =  exp/cell_covariates$n_umis,
                  batch = cell_covariates$batch)
ggplot(data = df, mapping = aes(x = batch, y = rel_exp)) +
  geom_violin() +  geom_boxplot() + theme_bw() +
  ylim(c(0.0, 0.05)) +
  theme(axis.text.x = element_text(angle = 90, vjust = 0.5, hjust=1)) +
  xlab("Batch") + ylab("Rel. expression")
```



We see clear variation in the expression of MRPL13 across batch. For example, the expression of MRPL13 appears to be especially high in batches 12 and 14.

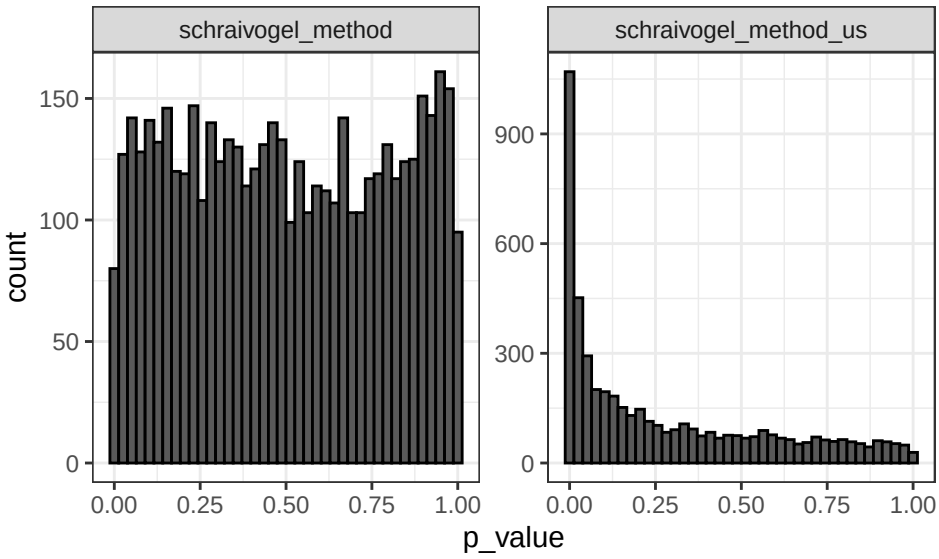
Discrepancy between our implementation of Schraivogel Method and the original Schraivogel results

Our implementation of Schraivogel Method does not seem to recapitulate the original results. In fact, the divergence is pretty substantial and likely will affect our communication of the results.

I plot histograms of the original Schraivogel p-values and our Schraivogel p-values. Ours are spikier than the originals.

```
schraivogel_res_us <- schraivogel_res_us |>
  dplyr::mutate(method = "schraivogel_method_us", spikiness = "medium")
to_plot <- rbind(schraivogel_res_us, schraivogel_res)

ggplot(to_plot, mapping = aes(x = p_value)) +
  facet_wrap(. ~ method, scales = "free_y", nrow = 1) +
  geom_histogram(bins = 40, col = "black") + theme_bw()
```


schraivogel_method
schraivogel_method_us
count
p_value
0
50
100
150
300
600
900
0.00
0.25
0.50
0.75
1.00