SCEPTRE analysis of sample Gasperini data

Gene Katsevich

November 16, 2023

In this document, we will apply `sceptre` to analyze the sample Gasperini data.

1 Load libraries and data

```r
# load libraries
library(MuData)
library(dplyr)
library(ggplot2)
library(conflicted)
library(kableExtra)
library(sceptre)
conflicted::conflicts_prefer(dplyr::rename)
conflicts_prefer(dplyr::filter)
conflicted::conflicts_prefer(base::unname)
```

```r
# read in the MuData object
gasperini_data_dir <- "~/code/research/Pipeline_Gasperini_2019/"
gasperini_data_fp <- paste0(
  gasperini_data_dir,
  "/mudata/Gasperini_2019_sample_pilot.h5mu"
)
gasperini_data <- readH5MU(gasperini_data_fp)
```

2 Import data into SCEPTRE

The first step is to get the data into a form that `sceptre` expects; see Import data from a collection of R objects in the manual.

```r
scRNA_data <- gasperini_data@ExperimentList$scRNA
guides_data <- gasperini_data@ExperimentList$guides
response_matrix <- scRNA_data@assays@data@listData[[1]]
grna_matrix <- guides_data@assays@data@listData[[1]]
extra_covariates <- scRNA_data@colData |>
  as.data.frame() |>
  select(percent_mito) # batch_number omitted because it is constant in these data
moi <- "high"
response_names <- scRNA_data@rowRanges@elementMetadata$feature_name |>
  as.character()
grna_target_data_frame <- guides_data@rowRanges@elementMetadata |>
  as.data.frame() |>
  rename(grna_id = feature_name, grna_target = target_elements, chr = guide_chr,
         start = guide_start, end = guide_end) |>
  select(grna_id, grna_target, chr, start, end)
```

Next, we import this data into a `sceptre_object`.

```
sceptre_object <- import_data(
  response_matrix = response_matrix,
  grna_matrix = grna_matrix,
  grna_target_data_frame = grna_target_data_frame,
  moi = moi,
  extra_covariates = extra_covariates,
  response_names = response_names
)
```

We can print this `sceptre_object` to get basic information about the data:

```
sceptre_object
```

```
## An object of class sceptre_object.
##
## Attributes of the data:
##  • 7314 cells
##  • 2127 responses
##  • High multiplicity-of-infection
##  • 98 targeting gRNAs (distributed across 49 targets)
##  • 0 non-targeting gRNAs
##  • 6 covariates (grna_n_nonzero, grna_n_umis, percent_mito, response_n_nonzero, response_n_umis, resp
```

3 Set analysis parameters

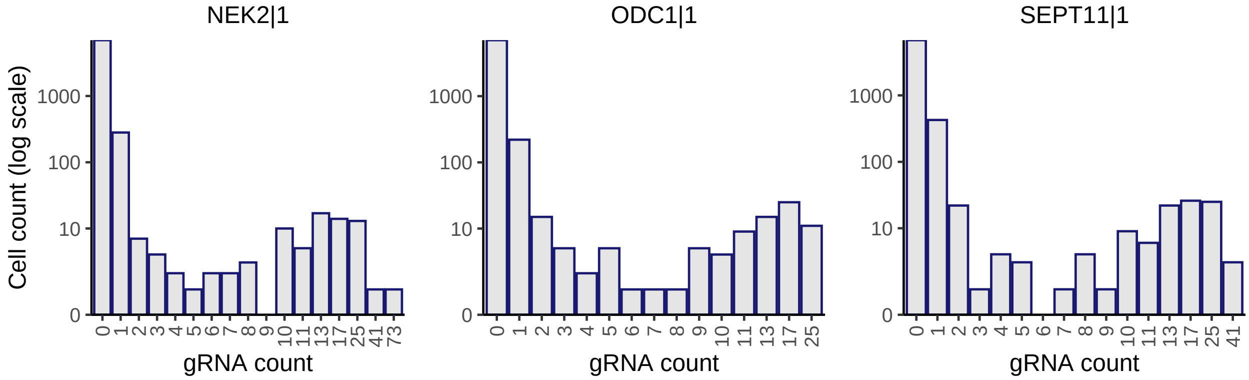
The next step is to set the analysis parameters.

```
# construct positive control pairs
response_ids <- rownames(response_matrix)
on_target_response_idx <- which(response_names %in% grna_target_data_frame$grna_target)
positive_control_pairs <- data.frame(
  grna_target = response_names[on_target_response_idx],
  response_id = response_ids[on_target_response_idx]
)
# no discovery pairs, since no enhancer-targeting perturbations present
discovery_pairs <- data.frame(grna_target = character(), response_id = character())
# set analysis parameters
sceptre_object <- sceptre_object |>
  set_analysis_parameters(
    discovery_pairs = discovery_pairs,
    positive_control_pairs = positive_control_pairs,
    side = "left"
  )
```

4 Assign gRNAs to cells

Next, we assign gRNAs to cells. Prior to this, it is useful to visualize the gRNA count distributions. Here, we do so for three gRNAs:

```r
# plot gRNA count distributions
grnas_to_plot <- c("NEK2|1", "ODC1|1", "SEPT11|1")
sceptre_object |>
  plot_grna_count_distributions(grnas_to_plot = grnas_to_plot)
```



The gRNA distributions are nicely bimodal, with 5 being a threshold to divide gRNAs with background reads from those actually present in a cell (this was Gasperini's original choice). Within `sceptre`, we can either use the thresholding method or a more sophisticated mixture method to assign gRNAs to cells. The mixture method adaptively sets a different threshold for different gRNAs in different cells.

```r
# assign gRNAs to cells via thresholding
sceptre_object <- sceptre_object |>
  assign_grnas(method = "thresholding", threshold = 5)

# alternatively, assign gRNAs to cells via mixture
sceptre_object_mixture <- sceptre_object |>
  assign_grnas(method = "mixture")
```

We can visualize the results of these two gRNA assignments (Figures 1 and 2).

```r
plot(sceptre_object, grnas_to_plot = grnas_to_plot)
```

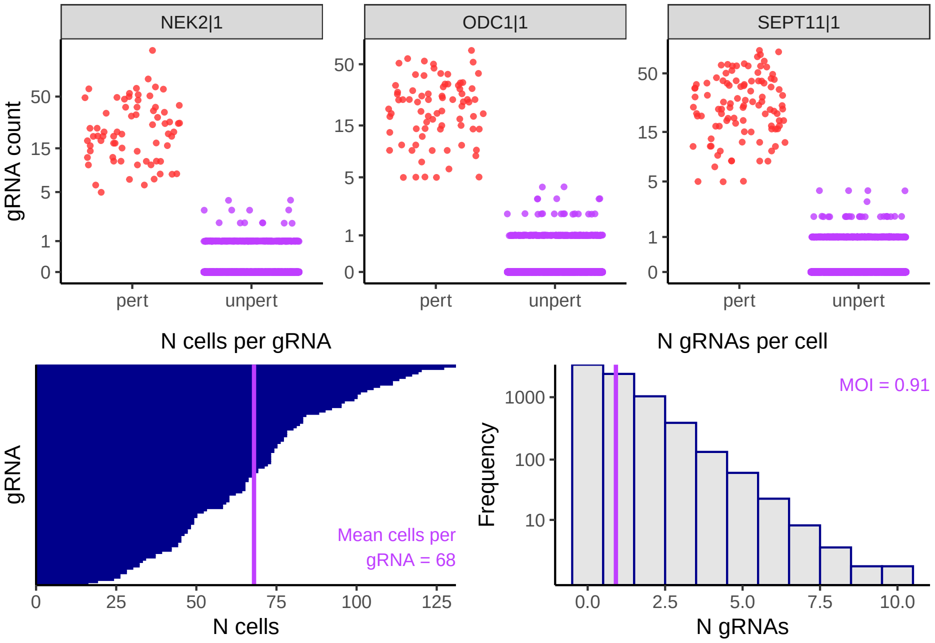


Figure 1: Thresholding-based gRNA assignment

```
plot(sceptre_object_mixture, grnas_to_plot = grnas_to_plot)
```

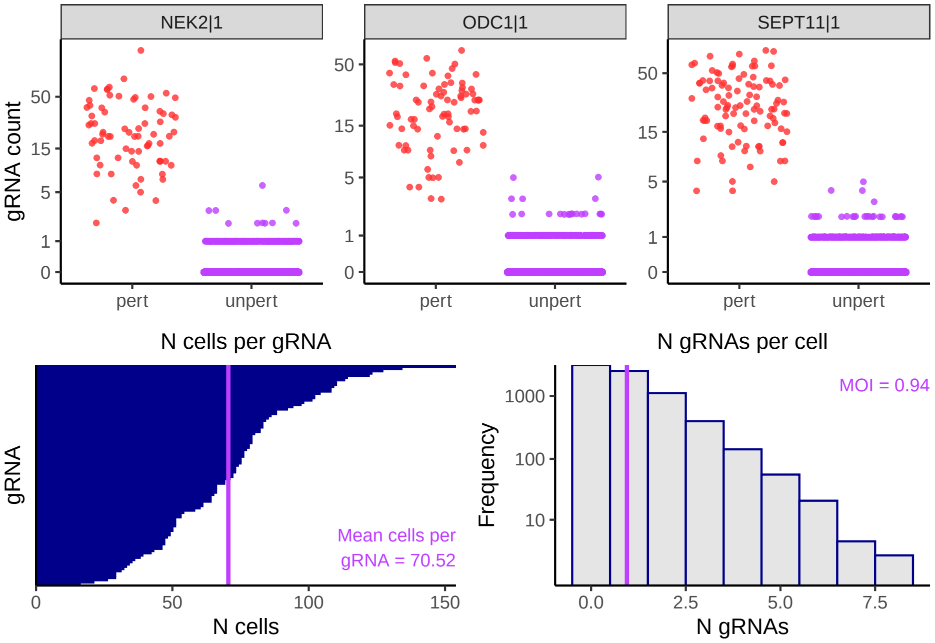


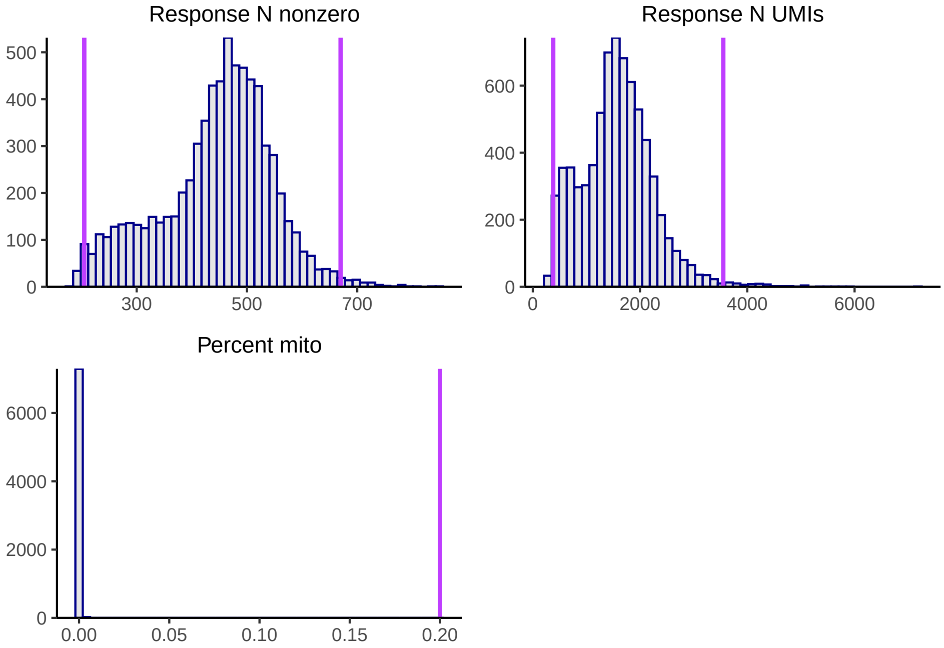
Figure 2: Mixture-based gRNA assignment

On these data, the two methods appear to give similar results.

5 Quality control

The next step is quality control. We can visualize covariates like the number of genes expressed in a cell ("Response N nonzero") and the total number of UMIs in a cell ("Response N UMIs").

```
sceptre_object |> plot_covariates()
```



We can remove cells with extreme values of these covariates via `run_qc`; we choose the default values of the QC thresholds, depicted in the figure above.

```
sceptre_object <- sceptre_object |> run_qc()
```

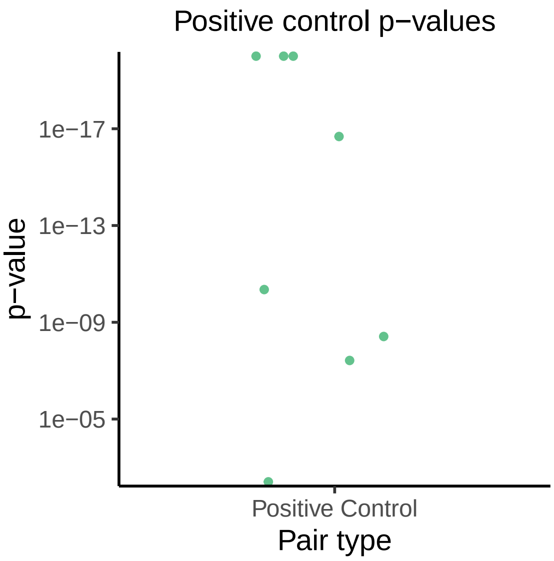
6 Positive control analysis

Finally, we analyze the positive control pairs via `run_power_check()`, of which there are only 8 in this sample dataset.

```
sceptre_object <- sceptre_object |> run_power_check()
```

The next figure shows the positive control p-values.

```
sceptre_object |>
  plot() +
  labs(title = "Positive control p-values")
```



Finally, we can print a table of the positive control p-values:

```
sceptre_object |>
  get_result(analysis = "run_power_check") |>
  select(grna_target, response_id, p_value, log_2_fold_change) |>
  mutate(p_value = format(p_value, scientific = TRUE, digits = 2)) |>
  kable(format = "latex",
        row.names = NA,
        booktabs = TRUE,
        digits = 2,
        caption = "Positive control p-values",
        linesep = '') |>
  kable_styling(latex_options = "hold_position")
```

Table 1: Positive control p-values

grna_target	response_id	p_value	log_2_fold_change
UBE2T	ENSG00000077152	7.3e-24	-1.92
LRIF1	ENSG00000121931	4.0e-22	-1.71
SDHC	ENSG00000143252	1.2e-21	-1.91
UAP1	ENSG00000117143	2.1e-17	-1.38
RBM17	ENSG00000134453	4.4e-11	-0.86
SRRM1	ENSG00000133226	3.9e-09	-0.51
NEK2	ENSG00000117650	3.8e-08	-1.51
BROX	ENSG00000162819	3.9e-03	-0.41

The positive control p-values are quite small, which is a good sign.