GO enrichment analysis of non-coding accelerated elements using rGREAT

Paula Beati

April 7, 2025

Go term analysis

In this document you will follow the pipeline used to compute GO term enrichment for non-coding accelerated regions.

Software

rGREAT https://github.com/jokergoo/rGREAT

Input data

Mammals (ncMARs)

Coordinates: hg38

Conserved elements: `./data/input/10_mammals_conserved_noncoding_elements.bed`

Accelerated elements: `./data/input/13_non_coding_acc_regions_mammals_ranked.bed`

Aves (ncAvARs)

Coordinates: galGal6

Conserved elements: `./data/input/12_aves_conserved_noncoding_elements.bed`

Accelerated elements: `./data/input/14_non_coding_acc_regions_aves_ranked.bed`

Common accelerated regions (CARs)

Coordinates: hg38

Conserved elements: `./data/input/10_mammals_conserved_noncoding_elements.bed`

Accelerated elements: `./data/input/supp14_ncCAR_hg38.bed`

Pipeline

Includes

```
base_path <<- getwd()
script_base_path <<- file.path(base_path, 'source')
data_base_path <<- file.path(base_path, 'data')
go_terms_data_base_path <<- file.path(data_base_path, 'output', 'go_terms')

source(file.path(script_base_path,  'rGreat_base.R'))
source(file.path(script_base_path, 'rGreat_simulation.R'))
source(file.path(script_base_path, 'rGreat_analysis.R'))
source(file.path(script_base_path, 'rGREAT_regions_in_genes.R'))
source(file.path(script_base_path, 'rGreat_results.R'))
source(file.path(script_base_path, 'dotplot.R'))
```

We want to evaluate go term enrichment for the set of regions associated with accelerated elements. We have an extra issue trying to know if the significance is associated with acceleration or if it is just a consequence of enrichment in the conserved regions set. (Remember that accelerated regions are a subset of conserved regions.)

The strategy to answer this question is to compare results obtained for the set of accelerated regions with those obtained from equivalent sets of conserved regions. We sampled conserved regions, building 5,000 sets with the same size and chromosome distribution as the accelerated regions set, and evaluated rGREAT on each of these sets.

Steps 1 to 3 are performed in function `main_mammals` in `./source/rGreat_analysis.R`

Step 1

Load or generate `rGREAT` evaluation for random sets of non-coding conserved elements, GO:BP ontology.

```
    go_ontology <- 'bp'

    nc_cons_mammals_file_name <- '10_mammals_conserved_noncoding_elements.bed'
    nc_cons_mammals_file_path <- file.path(data_base_path, 'input',
                                           nc_cons_mammals_file_name)
    file.exists(nc_cons_mammals_file_path)
    nc_cons_mammals <- read.delim(nc_cons_mammals_file_path, sep = ' ',
                                  header = TRUE)

    nc_acc_mammals_file_name <- '13_non_coding_acc_regions_mammals_ranked.bed'
    nc_acc_mammals_file_path <- file.path(data_base_path, 'input',
                                          nc_acc_mammals_file_name)
    file.exists(nc_acc_mammals_file_path)
    nc_acc_mammals <- read.delim(nc_acc_mammals_file_path, sep = ' ',
                                 header = TRUE)

    hits_stats_sim <- simulation_stats_clade_go_terms(nc_cons_mammals,
                                       nc_acc_mammals, go_ontology, 'mammals')
```

Step 2

Evaluate `rGREAT` on non-coding accelerated elements.

```
    # 2. calculate rgreat for acc elements
    go_terms_nc_acc_mammals_all <- enrichment_clade_go_terms(nc_acc_mammals,
                                            go_ontology, 'mammals', 'nc_acc_')
```

Step 3

The empirical p-value for each GO term in the accelerated region set is computed as the proportion of 'observed_regions_hits' values in its distribution within conserved elements that are greater than the obtained 'observed_regions_hits' value for the accelerated region set. These p-values are corrected for multiple comparisons, the Benjamini-Hochberg method was applied to compute approximate false discovery rates (FDRs) for each term.

We considered a term significant when its adjusted p-value returned from rGREAT was less than 0.05 and the adjusted empirical p-value was less than 0.05.

In this step we filter significant GO terms from `rGREAT` results on non-coding accelerated elements.

```
    # 3. empirical p values for acc elements
    significant_clade_go_terms(go_terms_nc_acc_mammals_all, hits_stats_sim,
                                  go_ontology, 'mammals', 'nc_acc_')
```

Step 4

To calculate the proportion of genes associated with accelerated regions and annotated genes in each term, we combined the genes data with the results from `rGREAT`.

The regions are associated with genes using `rGREAT:::getRegionGeneAssociations`

The following code is part of the function `main_mammals` in `./source/rGREAT_regions_in_genes.R`

```
    clade <- 'mammals'
    ontology <- 'bp'

    go_annotation_type <- 'biomart'
    result_file_path <- bind_genes_data(clade, ontology, go_annotation_type)
    result_data_biomart <- read.delim(result_file_path, sep = '\t',
                                      header = TRUE)

    go_annotation_type <- 'default'
    result_file_path <- bind_genes_data(clade, ontology, go_annotation_type)
    result_data_default <- read.delim(result_file_path, sep = '\t',
                                      header = TRUE)
```

Step 5

Format result tables.

The following code is part of the function `main` in `./source/rGreat_results.R`

```
    clade <- 'mammals'
    ontology <- 'bp'

    data_file_path_1 <- combine_results_regions(clade, ontology)
    data_1 <- read.delim(data_file_path_1, sep = '\t', header = TRUE)

    data_file_path_2 <- combine_results_annotations(clade, ontology)
    data_2 <- read.delim(data_file_path_2, sep = '\t', header = TRUE)

    print(nrow(data_1))
```

```
## [1] 345
```

```
    print(nrow(data_2))
```

```
## [1] 345
```

```
    print(nrow(data_1) == nrow(data_2))
```

```
## [1] TRUE
```

Step 6

Represent top 30 GO enrichemnt result in a dot plot.

The following code is part of the function `main` in `./source/dotplot.R`

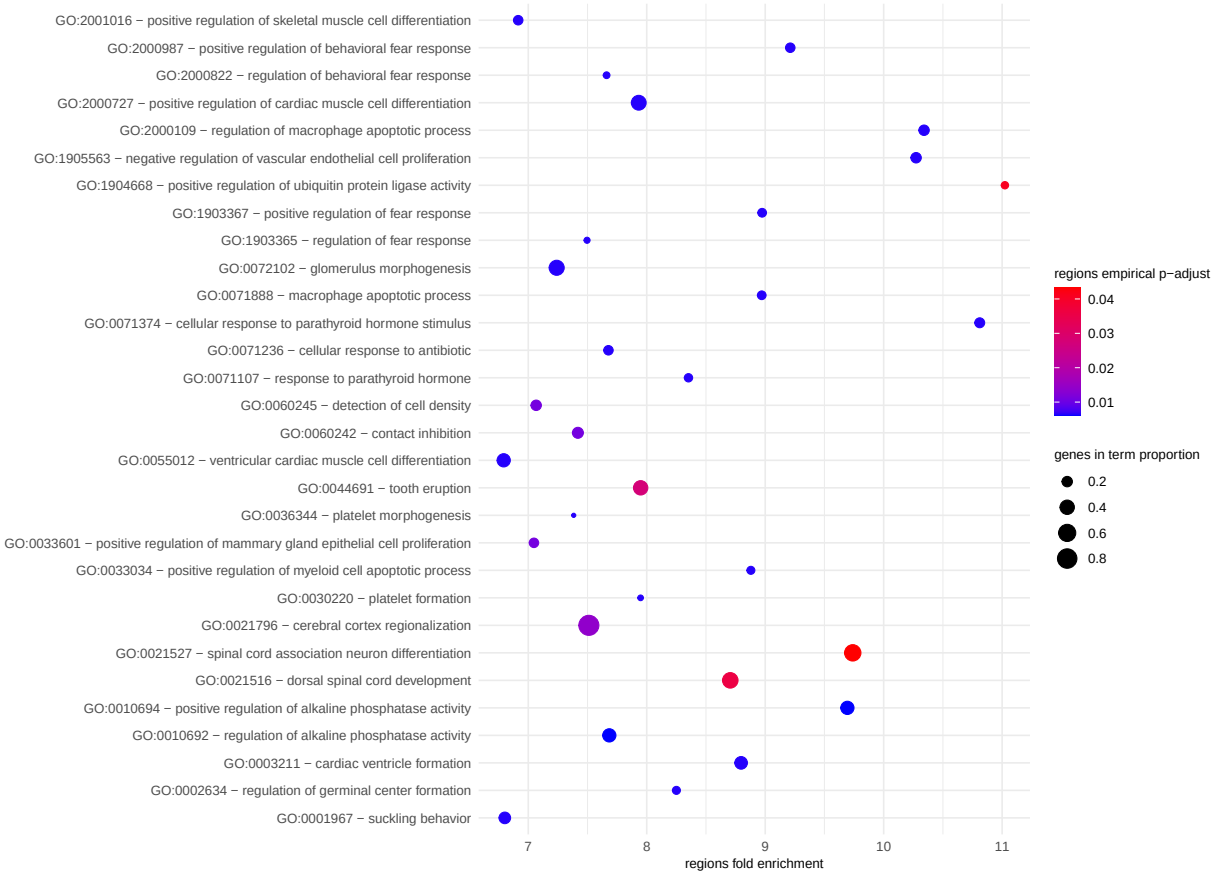
```
    clade <- 'mammals'
    go_ontology <- 'bp'

    plot_to_save <- dot_plot_clade_ontology(go_terms_data_base_path, go_ontology, clade)
    save_plot(plot_to_save, go_ontology, clade)
```

Result

```
    plot_base_path <- file.path(go_terms_data_base_path, clade, 'dot_plot')
    plot_file_name <- paste0('acc_nc_', clade, '_go', go_ontology,
                             '_top30fe.svg')
    plot_file_path <- file.path(plot_base_path, plot_file_name)

    knitr::include_graphics(plot_file_path)
```



Summary

The complete pipeline for mammals GO:BP, GO:MF, GO:CC is as follows:

1. `./source/rGreat_analysis.R`, `main_mammals()`
2. `./source/rGREAT_regions_in_genes.R`, `main_mammals()`
3. `./source/rGreat_results.R`, `main_mammals()`
4. `./source/dotplot.R`, `main_mammals()`

The complete pipeline for aves GO:BP, GO:MF, GO:CC is as follows:

1. `./source/rGreat_analysis.R`, `main_aves()`
2. `./source/rGREAT_regions_in_genes.R`, `main_aves()`
3. `./source/rGreat_results.R`, `main_aves()`
4. `./source/dotplot.R`, `main_aves()`