DASE User's Guide

Xiang Liu xiang.liu@moffitt.org Mingxiang Teng mingxiang.teng@moffitt.org Department of Biostatistics and Bioinformatics Moffit Cancer Center, Tampa, FL, USA

2022 - 01 - 13

Contents

Introduction	1
Getting Started	1
Preparing Inputs	2
Basic Usage of $DASE$	3
Interpretation of $DASE$ Outputs	6
Additional Options	9
Citation	11

Introduction

Super enhancers (SEs) are broad enhancer domains usually containing multiple constituent enhancers that hold elevated activities in gene regulation. Disruption in one or more constituent enhancers causes aberrant SE activities that lead to gene dysregulation in diseases. To quantify SE aberrations, differential analysis is performed to compare SE activities between cell conditions. The state-of-art strategy in estimating differential SEs relies on overall activities and neglect the changes in length and structure of SEs. DASE uses a weighted spline model to identify differential SEs between two conditions by accounting for the combinatorial effects of constituent enhancers weighted with their activities and locations (internal dynamics). In addition to overall changes, our method finds four novel types (shortened, shifted, hollowed and other complex scenarios) of differential SEs pointing to the structural differences within SEs.

Getting Started

Load the package in R

Preparing Inputs

DASE requires the following input files:

- 1. Enhancer BED file: a merged enhancer BED file including the enhancers (e.g. H3K27Ac sequencing peaks) from all compared samples.
- 2. SE BED file: a merged SE BED file including the SE regions from all compared samples.
- 3. Coverage files: either the path of BAM/BigWig files for each condition and replicate, or a sequencing count table for all enhancers across conditions and replicates.

Below are the examples of the enhancer and SE BED files, following the UCSC definition (https://genome.ucsc.edu/FAQ/FAQformat.html).

Enhancer BED file

The enhancer BED file can be of any table formats as long as the first 6 columns contain the information of "chromosome", "start", "end", "name", "score" and "strand". Here is an example of BED files with only 6 columns.

```
# load enhancer BED file
enhancer_path <- system.file("extdata","enhancer.bed",package="DASE")</pre>
enhancer_region <- read.table(enhancer_path,sep="\t",header=F)</pre>
head(enhancer_region)
        V1
                 V2
                                            V5 V6
                           V3
#> 1 chr21 10119622 10119934 Peak_59320
                                            58
#> 2 chr21 10413373 10414538 Peak_2651 1000
#> 3 chr21 13973708 13974647 Peak_51112
#> 4 chr21 14027434 14027662 Peak_40070
                                            88
#> 5 chr21 14381282 14381485 Peak_44344
#> 6 chr21 14382640 14384785 Peak_2461 1000
```

SE BED file

The SE BED file can be of any formats as long as the first 6 columns contain the information of "chromosome", "start", "end", "name", "score", and "strand". Here is an example of BED files with only 6 columns.

Basic Usage of DASE

In this section, we use DASE to find differential SEs by comparing SE profiles on **chromosome 21** between two cancer cell lines (K562 and MCF7). Here, we focus on the basic usage of running DASE with different coverage input files (BAM, BigWig, and enhancer raw count table). By default, DASE will run permutations to determine a significant threshold to aid the defining of different SE categories. Additional options, such as blacklist region removal and running without permutation can be found later in the "Additional Options" section.

Run DASE with BAM or BigWig coverage files

This section demonstrates *DASE* with BAM or BigWig input files. The BAM or BigWig files are used to estimate the reads counts of enhancers in different samples. *DASE* uses *featureCounts* to count the enhancer abundance from BAM files or uses *rtracklayer* for BigWig files. The first step is to get the paths of all BAM or BigWig files for each sample. Then, with the specified enhancer and SE regions, we can run *DASE* as follow (Here we use BigWig files as an example).

```
# path of BigWig file for each condition
s1_r1_bw <- system.file("extdata", "K562_1_chr21.bw", package="DASE")
s1_r2_bw <- system.file("extdata","K562_2_chr21.bw",package="DASE")</pre>
s2_r1_bw <- system.file("extdata", "MCF7_1_chr21.bw", package="DASE")
s2_r2_bw <- system.file("extdata","MCF7_2_chr21.bw",package="DASE")
# running DASE with BigWig files
DASE_out <- DASE(se_in=se_region,e_in=enhancer_region,data_type = "bw",
                 s1_r1_bam=s1_r1_bw,s1_r2_bam=s1_r2_bw,
                 s2_r1_bam=s2_r1_bw,s2_r2_bam=s2_r2_bw)
#> [1] "Step 1: merge and filter SE"
#> [1] "Step 2: calculate log2FC of constituent enhancer using Deseg2"
#> [1] "Step 3: b-spline fit log2FC"
#> [1] "Processing total of 34 SEs"
#> [1] "Step 4: permutation to get log2FC cutoff"
#> [1] "Permutation: 1"
#> [1] "Permutation: 2"
#> [1] "Permutation: 3"
#> [1] "Permutation: 4"
#> [1] "Permutation: 5"
#> [1] "Permutation: 6"
#> [1] "Permutation: 7"
#> [1] "Permutation: 8"
#> [1] "Permutation: 9"
#> [1] "Permutation: 10"
#> [1] "Step 5: pattern segments process"
#> [1] "Step 6: final category estimate"
```

Run DASE with enhancer raw count table

This section demonstrates DASE with enhancer raw count table. Instead of count enhancer reads from BAM or BigWig files, the function adapts count table directly to save time and space. The format of count table is shown below. The first column must be the enhancer name with "chr_start_end" format. The count table is adapted with parameter <code>enhancer_count_table</code>.

```
# read enhancer count table
enhancer_count_path <- system.file("extdata","chr21_enhancer_count.txt",package="DASE")</pre>
enhancer_count <- read.table(enhancer_count_path,sep="\t",header=T)</pre>
head(enhancer count)
                  enhancer S1 r1 S1 r2 S2 r1 S2 r2
#> 1 chr21 5128185 5128529
                              12
                                     2
                                           21
#> 2 chr21_5240507_5241144
                               29
                                     20
                                           15
                                                  1
#> 3 chr21_5241953_5242568
                                     29
                                           11
                                                  2
                               33
#> 4 chr21 5242733 5243984
                                     68
                                           12
                                                  7
                             160
#> 5 chr21_5244027_5244554
                              52
                                     29
                                           13
                                                  4
#> 6 chr21_5244634_5245418
                               59
                                     19
                                           13
                                                  1
# run DASE
DASE_out_count <- DASE(se_in=se_region,e_in=enhancer_region,
                 enhancer_count_table=enhancer_count)
#> [1] "Step 1: merge and filter SE"
#> [1] "Step 2: calculate log2FC of constituent enhancer using Deseq2"
#> [1] "Step 3: b-spline fit log2FC"
#> [1] "Processing total of 34 SEs"
#> [1] "Step 4: permutation to get log2FC cutoff"
#> [1] "Permutation: 1"
#> [1] "Permutation: 2"
#> [1] "Permutation: 3"
#> [1] "Permutation: 4"
#> [1] "Permutation: 5"
#> [1] "Permutation: 6"
#> [1] "Permutation: 7"
#> [1] "Permutation: 8"
#> [1] "Permutation: 9"
#> [1] "Permutation: 10"
#> [1] "Step 5: pattern segments process"
#> [1] "Step 6: final category estimate"
```

Run DASE with different spline functions

DASE uses spline functions to fit log2 fold change values of constituent enhancers between conditions. We chose to implement 3 widely-adapted spline functions (b-spline, natural spline, and smooth.spline) in DASE to provide flexibility, although we only reported results from b-splines in our manuscript due to its superiority in practice. DASE uses b-spline as default. The three functions provide similar results overall. However, they could have very different performance in some cases. The examples below show a brief comparison between spline functions.

Smooth spline performs badly on SEs with a small number of constituent enhancers (fitting points) (Figure 1a). When there are more number of data points (usually larger than 6), all spline functions show similar performance (Figure 1c). However, in practice, natural splines might overfit when regression weights are considered based on our algorithms (Figure 1b). Note that we implement a strategy to control overfitting with b-spline (see our manuscript).

Users can chose different spline functions with $spline_fun$ parameter in "DASE". Following are some examples of run DASE with different spline functions.

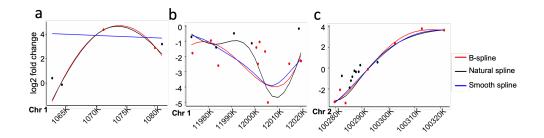
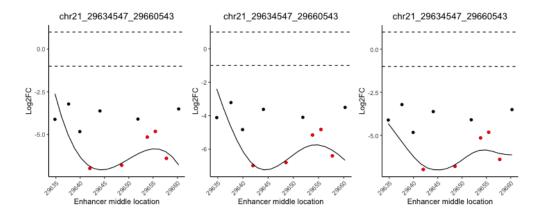


Figure 1: Spline fitting on different SEs.

```
#> [1] "Step 1: merge and filter SE"
#> [1] "Step 2: calculate log2FC of constituent enhancer using Deseq2"
#> [1] "Step 3: b-spline fit log2FC"
#> [1] "Processing total of 34 SEs"
#> [1] "Step 4: permutation to get log2FC cutoff"
#> [1] "Step 5: pattern segments process"
#> [1] "Step 6: final category estimate"
# run DASE with natural spline function
DASE_out_ns <- DASE(se_in=se_region,e_in=enhancer_region,
                 enhancer_count_table=enhancer_count,permut=F,
                 spline_fun = "ns")
#> [1] "Step 1: merge and filter SE"
#> [1] "Step 2: calculate log2FC of constituent enhancer using Deseq2"
#> [1] "Step 3: natural spline fit log2FC"
#> [1] "Processing total of 34 SEs"
#> [1] "Step 4: permutation to get log2FC cutoff"
#> [1] "Step 5: pattern segments process"
#> [1] "Step 6: final category estimate"
# run DASE with smooth spline function
DASE_out_smooth <- DASE(se_in=se_region,e_in=enhancer_region,</pre>
                 enhancer count table=enhancer count,permut=F,
                 spline_fun = "smooth")
#> [1] "Step 1: merge and filter SE"
#> [1] "Step 2: calculate log2FC of constituent enhancer using Deseq2"
#> [1] "Step 3: smooth spline fit log2FC"
#> [1] "Processing total of 34 SEs"
#> [1] "Step 4: permutation to get log2FC cutoff"
#> [1] "Step 5: pattern segments process"
#> [1] "Step 6: final category estimate"
# spline fitted plots of different spline functions
library(gridExtra)
grid.arrange(DASE_out_bs$pattern_list[[7]],
             DASE_out_ns$pattern_list[[7]],
             DASE_out_smooth$pattern_list[[7]],nrow=1)
```



Left is b-spline; middle is natural spline; right is smooth spline.

Interpretation of *DASE* Outputs

The output of DASE is a list with multiple data types including:

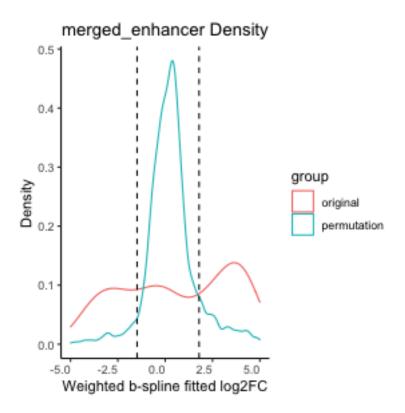
- 1. lfc_shrink: a shrinking lfc object from DESeq2 for all enhancers. It can be used to creat MA plot.
- 2. cutoff: significant threshold for fitted log2 fold changes.
- 3. density_plot: a density plot of permutation and original fitted $\log 2$ fold changes, if permut=T.
- 4. boxplot: a boxplot of final SE categories.
- 5. se_category: a data frame containing final SE categories.
- 6. pattern list: a list containing figures for each SE pattern.
- 7. ce_fit: a data frame containing DESeq2 output and spline-fitted log2 fold change of all constitute enhancers.

$Significant\ threshold$

We use permutation of spline fitted $\log 2$ fold changes to decide the significant threshold. Under default settings, DASE will run permutation 10 times with SEpermut function. If permutation is disabled, the default significant threshold is -1 and 1. Users can also choose their own thresholds with $cutoff_v$ parameter. Please refer to the $Additional\ options$ section for this.

```
# Significant threshold
DASE_out_count$cutoff
#> [1] -1.484361 1.790290
```

```
# Permutation density plot
DASE_out_count$density_plot
```



Black dash lines indicate the thresholds which are obtained based on the inflection points of the permutation distribution.

$Super-enhancer\ internal\ dynamic\ categories$

SE_category, pattern_list, ce_fit are the outputs related to SE internal dynamics.

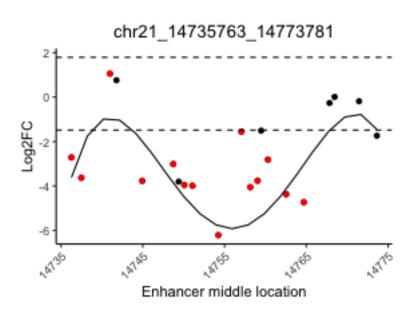
```
# se_categories
head(DASE_out_count$se_category)
#>
                 se\_merge\_name\ total\_width\ number\_enhancer\ category\ direction
#> 1 chr21_14735763_14773781
                                  2415.972
                                                         21
                                                                Other
                                                                           none
                                                         18
                                                                Other
#> 58 chr21_45402658_45478681
                                  1756.618
                                                                           none
#> 31 chr21_38784287_38854382
                                  2417.733
                                                         21
                                                                Other
                                                                           none
#> 51 chr21_43767083_43811262
                                   816.333
                                                         13
                                                                Other
                                                                           none
#> 41 chr21_40366721_40390218
                                  6696.416
                                                          5 Overall
#> 12 chr21_29634547_29660543
                                  4321.917
                                                         11 Overall
                          mean\_FC \ rank
#>
      non_mid_percent
#> 1
                0.962 -2.5822940
                                     1
#> 58
                0.712 0.9187119
                                     2
#> 31
                                     3
                0.510 1.1477844
#> 51
                0.347 0.2621700
                                     4
                1.000 5.1182229
#> 41
                                     1
                                     2
#> 12
                1.000 -4.8675004
```

Each column represents the following:

1. se_merge_name: name of merged SE, "chr_start_end".

- 2. total width: width of merged SE (unit=k).
- 3. number_enhancer: number of CEs in each SE.
- 4. category: SE category identified by DASE.
- 5. direction: enrichment direction of SEs (none:Other or non-differential category; +: enriched in sample 2; -: enriched in sample 1; l: sample 1 shifted in 5' direction; r: sample 2 shifted in 5' direction).
- 6. non_mid_percent: percentage of non middle segments.
- 7. mean FC: mean of log2 SE coverage fold change.
- 8. rank: SE category ranking based on non_mid_percent first and mean_FC for each SE category. (rank=1 means the most like to the SE category.)

```
# an example of one sample
DASE_out_count$pattern_list[[1]]
```



This figure shows the constituent enhancer patterns within a SE example which is identified as *shortened*. Black line is the fitted log2 fold change curve; dots indicate constituent enhancers. Red dots indicate the constituent enhancers with heavy weights.

```
# example of ce_fit
head(DASE_out_count$ce_fit)
                                 chr
                                                    end width S1_r1 S1_r2 S2_r1
#>
                  e_merge_name
                                         start
#> 1: chr21_14751738_14756723 chr21 14751738 14756723
                                                                        471
#> 2: chr21_14760734_14764325 chr21 14760734 14764325
                                                          3592
                                                                        303
                                                                               91
                                                                 254
#> 3: chr21_14750604_14751557 chr21 14750604 14751557
                                                           954
                                                                  61
                                                                        49
                                                                               14
#> 4: chr21_14749603_14750540 chr21 14749603 14750540
                                                           938
                                                                  52
                                                                        46
                                                                               14
#> 5: chr21_14757516_14758773 chr21 14757516 14758773
                                                          1258
                                                                  63
                                                                        33
                                                                               11
#> 6: chr21_14737131_14737833 chr21 14737131 14737833
                                                           703
                                                                  35
                                                                        31
                                                                               13
      S2\_r2 S1\_r1\_norm S1\_r2\_norm S2\_r1\_norm S2\_r2\_norm
                                                           baseMean log2FoldChange
#> 1:
         13 600.60927 1445.38402 11.401371
                                               14.391655 517.94658
                                                                           -6.333719
#> 2:
             329.49191
                         929.83303
                                    21.615100
                                                33.211512 328.53789
         30
                                                                           -4.540646
              79.12995
#> 3:
          9
                         150.36904
                                      3.325400
                                                 9.963453
                                                            60.69696
                                                                           -4.262678
          8
              67.45504
                         141.16277
                                     3.325400
                                                 8.856403
                                                                           -4.238023
#> 4:
                                                           55.19990
          7
              81.72437
                         101.26894
                                      2.612814
                                                 7.749353
#> 5:
                                                           48.33887
                                                                           -4.337614
#> 6:
                                                 6.642302 37.56601
                                                                           -3.965940
              45.40243
                          95.13143
                                      3.087871
                                                  padj baseMean shrink
          lfcSE
                       stat
                                  pvalue
```

```
#> 1: 0.6199495 -10.216508 1.672580e-24 2.263000e-22
                                                            517.94658
#> 2: 0.6328657
                -7.174738 7.244567e-13 1.849415e-11
                                                            328.53789
#> 3: 0.7638403 -5.580588 2.397068e-08 2.206281e-07
                                                             60.69696
#> 4: 0.7751509 -5.467353 4.568069e-08 4.039606e-07
                                                             55.19990
#> 5: 0.7788941
                -5.568940 2.562936e-08 2.327284e-07
                                                             48.33887
#> 6: 0.8182816 -4.846669 1.255515e-06 8.368035e-06
                                                             37.56601
#>
      log2FoldChange_shrink lfcSE_shrink pvalue_shrink padj_shrink
#> 1:
                  -6.211757
                               0.6280041 1.672580e-24 2.263000e-22
#> 2:
                               0.6463209
                                          7.244567e-13 1.849415e-11
                  -4.370066
#> 3:
                  -3.986558
                               0.8052662
                                          2.397068e-08 2.206281e-07
                                          4.568069e-08 4.039606e-07
#> 4:
                  -3.952824
                               0.8168441
#> 5:
                  -4.053649
                               0.8223654 2.562936e-08 2.327284e-07
#> 6:
                               0.8652202 1.255515e-06 8.368035e-06
                  -3.630087
#>
                se_merge_name
#> 1: chr21_14735763_14773781
#> 2: chr21_14735763_14773781
#> 3: chr21_14735763_14773781
#> 4: chr21_14735763_14773781
#> 5: chr21_14735763_14773781
#> 6: chr21_14735763_14773781
```

Columns from "e_merge_name" to "width" indicate the property of each CEs. Columns from "S1_r1" to "S2_r2_norm" indicate the coverage of each CE in each sample. Columns from "baseMean" to "pvalue" indicate the results of DESeq2. Columns from "baseMean_shrink" to "padj_shrink" indicate DESeq2 shrinkage value. Please refer to DASE help for more information.

Additional Options

In addition of default parameters, *DASE* can exclude the ENCODE blacklist regions from consideration. *DASE* can also adapt an user-define blacklist region with *custom_range* parameter. We have included the human blacklist region file from ENCODE (accession ID: ENCFF356LFX) in our package.

```
# blacklist file
blacklist_path <- system.file("extdata", "region_blacklist.bed", package="DASE")</pre>
blacklist_region <- read.table(blacklist_path, sep="\t", header=F)</pre>
head(blacklist_region)
       V1
                 V2
                           VЗ
#> 1 chr1
            628903
                      635104
#> 2 chr1
           5850087 5850571
#> 3 chr1
           8909610
                     8910014
#> 4 chr1 9574580
                     9574997
#> 5 chr1 32043823 32044203
#> 6 chr1 33818964 33819344
```

There is also an option to opt-out permutation calculation. The default setting of DASE will run permutation 10 times. Users can turn it off with permut=F. When there is no permutation, the default thresholds are -1 and 1, or users can choose the thresholds they like with $cutoff_v$ parameter. However, defining customized thresholds are only available under permut=F.

DASE with enhancer blacklist region

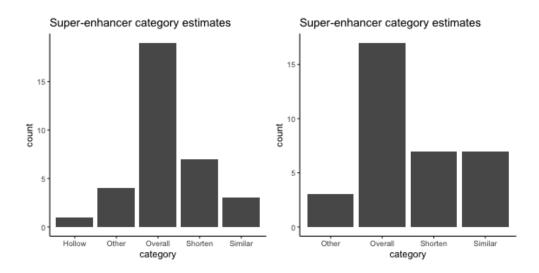
Example of using blacklist option.

```
# run DASE with blacklist file and customized region
DASE_out_bl <- DASE(se_in=se_region,e_in=enhancer_region,bl_file = blacklist_region,
                 custom range = c("chr21:14735763-29634779","chr21:33539902-43710703"),
                 enhancer count table=enhancer count)
#> [1] "Step 1: merge and filter SE"
#> [1] "Step 2: calculate log2FC of constituent enhancer using Deseq2"
#> [1] "Step 3: b-spline fit log2FC"
#> [1] "Processing total of 8 SEs"
#> [1] "Step 4: permutation to get log2FC cutoff"
#> [1] "Permutation: 1"
#> [1] "Permutation: 2"
#> [1] "Permutation: 3"
#> [1] "Permutation: 4"
#> [1] "Permutation: 5"
#> [1] "Permutation: 6"
#> [1] "Permutation: 7"
#> [1] "Permutation: 8"
#> [1] "Permutation: 9"
#> [1] "Permutation: 10"
#> [1] "Step 5: pattern segments process"
#> [1] "Step 6: final category estimate"
```

Users can find that the number of SEs (i.e. 8) in "DASE_out_bl" is less than that in "DASE_out_count" which is calculated without blacklist region (i.e. 34).

DASE with no permutation

Two examples of different permutation options.



By comparing results based on different threshold cutoffs, the right plot (with larger threshold cutoff) identified more SEs in the *similar* category than the left plot (with smaller threshold cutoff).

Citation

Please cite our paper when using DASE: [https://www.biorxiv.org/content/10.1101/2021.09.25.461810]