Package ‘EPOM’

Ruochen Jiang, Wei Vivian Li, Jessica Jingyi Li

2018-01-28

Introduction

EPOM is an R package developed corresponding to the paper :

Li, W.V., Razaee, Z.S. and Li, J.J., 2016, December. Epigenome overlap measure (EPOM) for comparing tissue/cell types based on chromatin states. In BMC genomics (Vol. 17, No. 1, p. S10). BioMed Central.

Set up

The package depends on the Bioconductor R package `rtracklayer` in order to read the bigwig file into R. It also depends on the `parallel` package in order to speed up the computation.

```
source("https://bioconductor.org/biocLite.R")
biocLite("rtracklayer")
library(parallel)
library(readr)
```

Functions

1.**epom()**

epom() is the main function of the package, and it takes 15 parameters. **bw_list** is a character vector containing all the full paths of bigwig files. **bed_list** is a character vector containing all the full paths of bed files. **bd_header** and **bed_sep** are used to read in the bed files with `read.table`. **cell_type** is a character vector containing the cell types of all samples. **histmark** is a character string specifying the histone modification mark used in the analysis, and **state_name** is a character string denoting the type of chromatin state used.

Below is an example for the **bw_list** (each element should be the full path):

```
bw_list <- c("C:/E001-H3K4me1.imputed.pval.signal.bigwig",
"C:/E002-H3K4me1.imputed.pval.signal.bigwig",
"C:/E003-H3K4me1.imputed.pval.signal.bigwig",
"C:/E004-H3K4me1.imputed.pval.signal.bigwig",
"C:/E005-H3K4me1.imputed.pval.signal.bigwig",
"C:/E006-H3K4me1.imputed.pval.signal.bigwig",
```

```
"C:/E007-H3K4me1.imputed.pval.signal.bigwig",
"C:/E008-H3K4me1.imputed.pval.signal.bigwig")
```

An example for the BigWig file.

```
> bw1
GRanges object with 121452164 ranges and 1 metadata column:
              seqnames                ranges strand |     score
                 <Rle>             <IRanges>  <Rle> | <numeric>
          [1]     chr1             [  1,  25]      * |         0
          [2]     chr1             [ 26,  50]      * |         0
          [3]     chr1             [ 51,  75]      * |         0
```

Below is an example for the **bed_list** (each element should be the full path):

```
bed_list <- c("C:/E001_25_imputed12marks_stateno.bed",
"C:/E002_25_imputed12marks_stateno.bed",
"C:/E003_25_imputed12marks_stateno.bed",
"C:/E004_25_imputed12marks_stateno.bed",
"C:/E005_25_imputed12marks_stateno.bed",
"C:/E006_25_imputed12marks_stateno.bed",
"C:/E007_25_imputed12marks_stateno.bed",
"C:/E008_25_imputed12marks_stateno.bed")
```

An example for the bed file.

```
> E001_25_imputed12marks_stateno
# A tibble: 933,207 x 4
      X1     X2     X3    X4
   <chr>  <int>  <int> <int>
 1  chr1      0  10000    25
 2  chr1  10000  10600    21
 3  chr1  10600 237600    25
 4  chr1 237600 237800    19
```

An example for the **cell_type**, each element should be a cell type.

```
cell_type <- c("ESC", "ESC", "ES-deriv", "ES-deriv", "ES-deriv", "ES-deriv", "ESC")
```

It is worth noticing that **bw_list**, **bed_list** and **cell_type** all have the same length and they should match each other in order.

An example for **hismark**:

```
histmark = "H3K4me1"
```

An example for **state**. Here Chromatin states are identified in Table 1 in the paper Li et al (2016). According to the paper states 10 to 18 correspond to the enhancer state.

```
state = 10:18
```

An example for **state_name**.

```
state_name = "enhancer"
```

in_dir indicates the directory that stores the intermediate files generated in previous running. The default value is `NULL`. If it is the first time you run `epom` on the given data, you can ignore this argument.

```
in_dir = "C:/Users/username/Desktop/read_in"
```

out_dir indicates the output directory. In the output directory, you can find .RData files saved for each step. The final result will also be saved in this directory. The default value for out_dir is the working directory of R. If users do not specify the **out_dir** argument, the output files will be saved in the working directory. An example for **out_dir**:

```
out_dir = "C:/Users/username/Desktop/output"
```

Even though the arguments **alpha_1**, **alpha_2**, **cores** and **m** are given default values, a brief description is given here in case the users want to modify them. **alpha_1** is the threshold used in ANOVA. **alpha_2** and **m** are the thresholds used in the t test procedure. **cores** specifies the number of cores used in parallel computation (try to aviod using all the **cores** on the machine).

```
epom(bw_list, bed_list, bd_header = FALSE, bed_sep = "\t",
     cell_type, histmark, state, state_name,
     in_dir = in_dir, save = TRUE, out_dir = out_dir,
     alpha_1 = 1e-10, cores = detectCores()-1, alpha_2 = 0.01, m = 13)
```

The `epom` function returns a symmetric matrix containing the epom scores. The rows and columns correspond to the samples, and the matrix entries are the pairwise similarity scores of the samples.

2.**Usage of Individual Functions**

If users want to modify the EPOM method to perform their own analysis and just employ a portion of the procedures in EPOM, the following examples can serve as a guide.

```
# read in a single bigwig file
bw1 <- import.bw("C:/E001-H3K4me1.imputed.pval.signal.bigwig")
bw2 <- import.bw("C:/E002-H3K4me1.imputed.pval.signal.bigwig")
bw3 <- import.bw("C:/E003-H3K4me1.imputed.pval.signal.bigwig")

#An example for the BigWig file.
# > bw1
# GRanges object with 121452164 ranges and 1 metadata column:
#              seqnames                ranges strand |     score
#                 <Rle>             <IRanges>  <Rle> | <numeric>
#          [1]     chr1             [  1,  25]      * |         0
#          [2]     chr1             [ 26,  50]      * |         0
```

```
#          [3]     chr1             [ 51,  75]      * |         0

# read in bed file
bed1 <- as.data.frame(read.table('E001_25_imputed12marks_stateno.bed', header = FALSE,
sep='\t', stringsAsFactors=TRUE, quote=''))
bed2 <- as.data.frame(read.table('E002_25_imputed12marks_stateno.bed', header = FALSE,
sep='\t', stringsAsFactors=TRUE, quote=''))
bed3 <- as.data.frame(read.table('E003_25_imputed12marks_stateno.bed', header = FALSE,
sep='\t', stringsAsFactors=TRUE, quote=''))

#An example for the bed file.
# > bed1
# A tibble: 933,207 x 4
#      X1     X2     X3    X4
#   <chr>  <int>  <int> <int>
# 1  chr1      0  10000    25
# 2  chr1  10000  10600    21
# 3  chr1  10600 237600    25
# 4  chr1 237600 237800    19

# align bw file and bed file
# transform bw1 to 200 bp
matrix_gen_1 <- matrix_bp_trans(bw1, 200)
matrix_gen_2 <- matrix_bp_trans(bw2, 200)
matrix_gen_3 <- matrix_bp_trans(bw3, 200)

# get the index of the 200 bp matrix according to the bed file and state information.
index <- list()
index[[1]] <- index_num4state(matrix_gen_1, bed1, state)
index[[2]] <- index_num4state(matrix_gen_2, bed2, state)
index[[3]] <- index_num4state(matrix_gen_3, bed3, state)
```

After the previous steps, we will obtain a matrix called **matrix_gen_1** containing signal of E001 sample and the bandwidth is 200bp. We will also obtain a vector of indices indicating the selected regions according to enhancers in the sample. Then we will take the union of all the selected index regions among all the samples. The union will be the candidate associated regions.

More specifically, if we have 124 epigenetic samples, we will generate a matrix called `matrix_gen` containing 3+124 = 127 columns, with the first three columns recording the information of “chromatin state”, “begin” and “width” as in the bigwig file. For the rest 124 columns, each column corresponds to the signal in one sample.

We will also have a vector called **row_select** that is the union of the indicies (**index**) selected according to the enhancer region of each sample.

```
row_select <- Reduce(union, index)
```

The matrix **matrix_gen** is generated while the BigWig files are transformed to 200 bp matrices. We initialized **matrix_gen** as the matrix generated by the first BigWig file and then column stack the signal for the latter transformed matrices as follows.

```
matrix_gen = matrix_gen_1
matrix_gen <- cbind(matrix_gen, matrix_gen_2[,4])
matrix_gen <- cbind(matrix_gen, matrix_gen_3[,4])
```

The next step is to perform ANOVA for each region.

```
l = anova_select(cell_type, row_select, matrix_gen, state = "enhancer", alpha_1 = 1e-10,
                 in_dir = in_dir, save = TRUE, out_dir = out_dir, cores = detectCores()-1)
```

Then perform the t test for each region.

```
l_t = t_select(l$select_reg, l$mat_anova, cell_type, state = "enhancer", alpha_2 = 0.01,
      m = 13, in_dir = in_dir, save = TRUE, out_dir = out_dir, cores = detectCores()-1)
```

Finnaly, calculate the epom scores.

```
epom_matrix = epom_score(l_t, cell_type, save = TRUE, out_dir = out_dir)
```