target: An R Package to Predict Combined Function of Transcription Factors

Mahmoud Ahmed[1] **and Deok Ryong Kim**[1]

[1]**Department of Biochemistry and Convergence Medical Sciences and Institute of Health Sciences, Gyeongsang National University School of Medicine. Jinju, Korea**

Abstract Abstracts should be up to 300 words and provide a succinct summary of the article. Although the abstract should explain why the article might be interesting, care should be taken not to inappropriately over-emphasise the importance of the work described in the article. Citations should not be used in the abstract, and the use of abbreviations should be minimized.

Keywords

transcription-factors; DNA-binding; gene-expression; r-package; bioconductor; workflow

R version: R version 3.6.1 (2019-07-05)

Bioconductor version: 3.9

Introduction

The introduction provides context as to why the software tool was developed and what need it addresses. It is good scholarly practice to mention previously developed tools that address similar needs, and why the current tool is needed.

Methods

Implementation

For software tool papers, this section should address how the tool works and any relevant technical details required for implementation of the tool by other developers.

Operation

This part of the methods should include the minimal system requirements needed to run the software and an overview of the workflow for the tool for users of the tool.

Use Cases

Table 1. Expression and binding data of YY1 and YY2 in HeLa cells.

GEO ID	Data Type	Design	Ref.
GSE14964	Microarrays	YY#-knockdown	Chen et al. [1]
GSE31417	ChIP-Seq	YY1 vs input	Michaud et al. [2]
GSE96878	ChIP-Seq	YY2 vs input	Wu et al. [3]

Table 1

```
# load required libraries
library(tidyverse)
library(reshape2)
library(broom)
library(cowplot)
library(ggupset)
library(seqLogo)
library(ggplotify)
library(rtracklayer)
library(GenomicRanges)
library(Biostrings)
library(TxDb.Hsapiens.UCSC.hg19.knownGene)
library(BSgenome.Hsapiens.UCSC.hg19)
library(org.Hs.eg.db)
library(BCRANK)
library(target)
```

Preparing the binding data

```
# locate the peaks bed files
peak_files <- c(YY1 = 'data/0th.Utr.05.YY1.AllCell.bed',
                YY2 = 'data/0th.Utr.05.YY2.AllCell.bed')

# load the peaks bed files as GRanges
peaks <- map(peak_files, ~GRanges(import.bed(.x)))
```

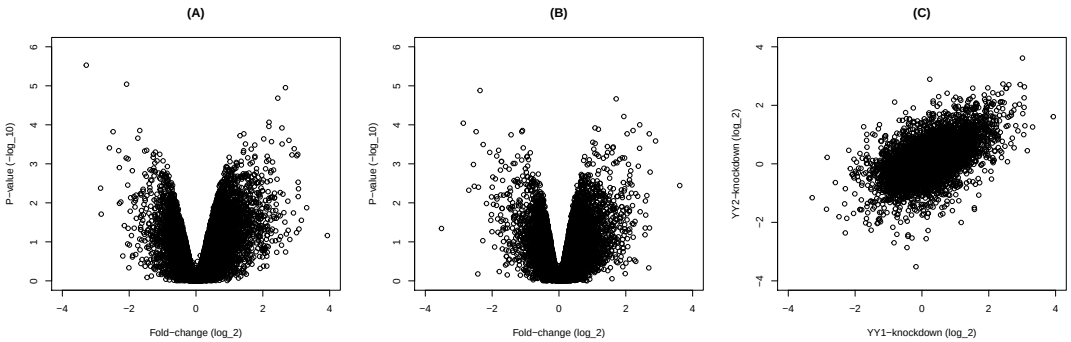

Figure 1. Differential expression between factor knockdown and wild-type in HeLa cells.

Preparing the expression data

```
# locate the expression text files
expression_files <- c(YY1 = 'data/DataSet_01_18.tsv',
                      YY2 = 'data/DataSet_01_19.tsv')

# load the expression text files
express <- map(expression_files,
               ~read_tsv(.x, col_names = FALSE) %>%
                 dplyr::select(2,3, 7, 9) %>% #9
                 setNames(c('tf', 'gene', 'fc', 'pvalue')) %>%
                 filter(tf %in% c('YY1', 'YY2')) %>%
                 na.omit())
```

Figure 1

```
par(mfrow = c(1, 3))

# volcano plot of YY1 knockdown
plot(express$YY1$fc,
     -log10(express$YY1$pvalue),
     xlab = 'Fold-change (log_2)',
     ylab = 'P-value (-log_10)',
     xlim = c(-4, 4), ylim = c(0, 6))
title('(A)')

# volcano plot of YY2 knockdown
plot(express$YY2$fc,
     -log10(express$YY2$pvalue),
     xlab = 'Fold-change (log_2)',
     ylab = 'P-value (-log_10)',
     xlim = c(-4, 4), ylim = c(0, 6))
title('(B)')

# plot fold-change of YY1 and YY2
plot(express$YY1$fc[order(express$YY1$gene)],
     express$YY2$fc[order(express$YY2$gene)],
     xlab = 'YY1-knockdown (log_2)',
     ylab = 'YY2-knockdown (log_2)',
     xlim = c(-4, 4), ylim = c(-4, 4))
title('(C)')
```

Preparing genome annotation

```
# load genome data
symbol_entrez <- select(org.Hs.eg.db,
```

```
                        unique(c(express$YY1$gene)),
                        'ENTREZID',
                        'SYMBOL') %>%
  setNames(c('gene', 'gene_id'))

# format genome to join with express
genome <- transcripts(TxDb.Hsapiens.UCSC.hg19.knownGene,
                      filter = list(gene_id = symbol_entrez$gene_id),
                      columns = c('tx_id', 'tx_name', 'gene_id')) %>%
  promoters(upstream = 100000) %>%
  as_tibble() %>%
  filter(length(gene_id) > 1) %>%
  mutate(gene_id = as.character(gene_id))
```

```
# make regions by merging the genome and express data
regions <- map(express,
               ~inner_join(genome, symbol_entrez) %>%
                 inner_join(.x) %>%
                 makeGRangesFromDataFrame(keep.extra.columns = TRUE))
```

Predicting gene targets of indiviual factors

```
# get associated peaks
ap <- map2(peaks, regions,
           ~associated_peaks(peaks=.x,
                             regions = .y,
                             regions_col = 'tx_id'))

# get direct targets
dt <- map2(peaks, regions,
           ~direct_targets(peaks=.x,
                           regions = .y,
                           regions_col = 'tx_id',
                           stats_col = 'fc'))
```

Figure 2

```
par(mfrow = c(2, 2))

# plot distance by score of associate peaks
map2(ap, c('(A)', '(B)'),
     ~{
       plot(.x$distance,
            .x$peak_score,
            xlab = 'Distance',
            ylab = 'Peak Score')
       title(.y)
     })

# make labels, colors and groups
labs <- c('down', 'none', 'up')
cols <- c('green', 'gray', 'red')

# make three groups by quantiles
groups <- map(dt,
              ~cut(.x$stat,
                   breaks = quantile(.x$stat, c(0, .1, .9, 1)),
                   labels = labs))

# plot the group functions
pmap(list(dt, groups, c('(C)', '(D)')),
    function(x, y, z) {
```

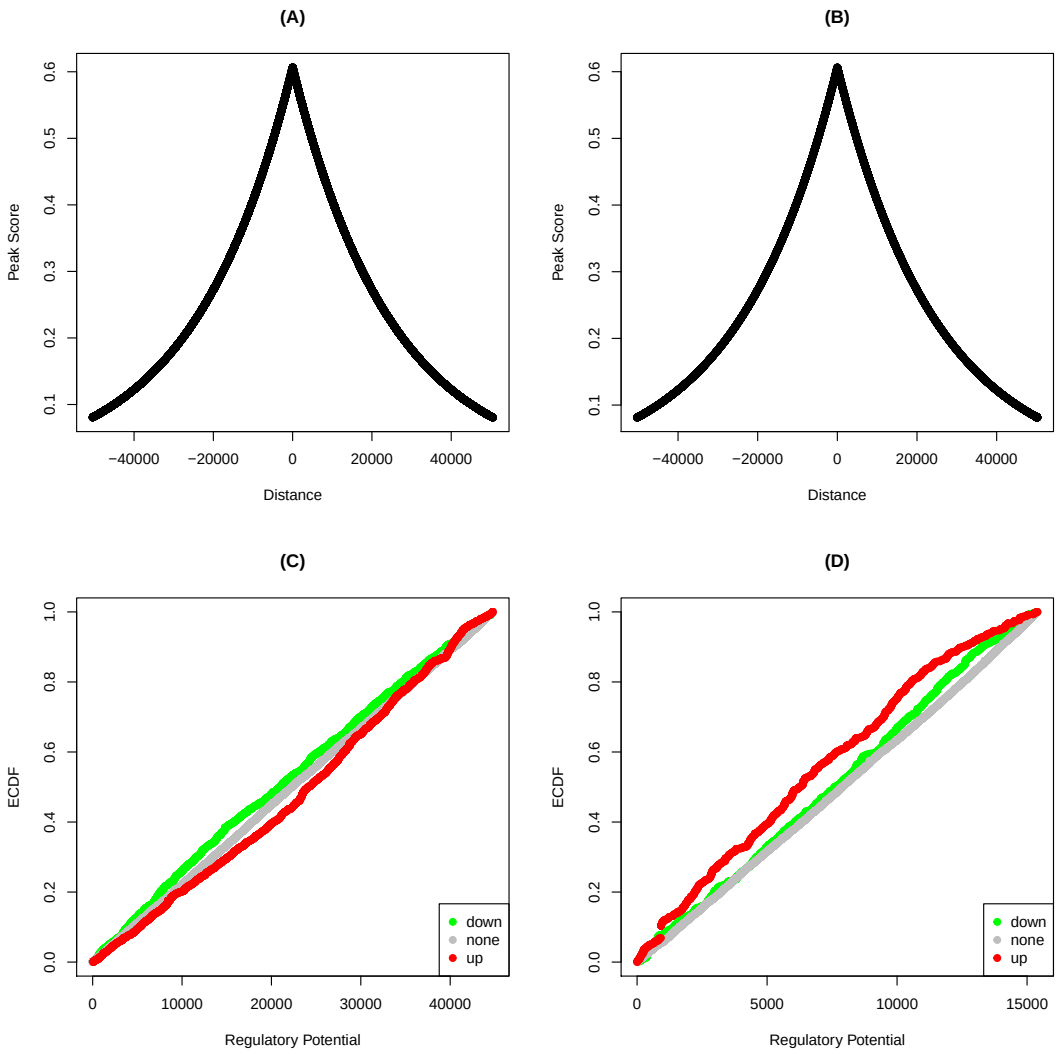

Figure 2. Predicted functions of YY1 and YY2 on their specific targets.

```
      plot_predictions(x$score_rank,
                       group = y,
                       colors = cols,
                       labels = labs,
                       xlab = 'Regulatory Potential',
                       ylab = 'ECDF')

      title(z)
    })
```

```
# test individual factor functions
map2(dt, groups,
     ~test_predictions(.x$rank,
                       group = .y,
                       compare = c('down', 'up')))
```

Table 2. Testing for statistical significance of the regulated gene groups.

Factor	Statistic	P.value	Method	Alternative
YY1	0.224	2.2e-16	Two-sample KS test	two-sided
YY2	0.149	2.5e-15	Two-sample KS test	two-sided

Table 2

Predicting the shared targets of two factors

```
# merge and name peaks
common_peaks <- reduce(subsetByOverlaps(peaks$YY1, peaks$YY2))
common_peaks$name <- paste0('common_peak_', 1:length(common_peaks))
```

```
# bind express tables into one
both_express <- bind_rows(express) %>%
  nest(fc, pvalue, .key = 'values_col') %>%
  spread(tf, values_col) %>%
  unnest(YY1, YY2, .sep = '_')

# make regions using genome and expression data of both factors
both_regions <- inner_join(genome, symbol_entrez) %>%
  inner_join(both_express) %>%
  makeGRangesFromDataFrame(keep.extra.columns = TRUE)
```

```
# get associated peaks with both factors
common_ap <- associated_peaks(peaks = common_peaks,
                              regions = both_regions,
                              regions_col = 'tx_id')

# get direct targets of both factors
common_dt <- direct_targets(peaks = common_peaks,
                            regions = both_regions,
                            regions_col = 'tx_id',
                            stats_col = c('YY1_fc', 'YY2_fc'))
```

Figure 3

```
par(mfrow = c(1, 2))

# plot distiace by score for associated peaks
plot(common_ap$distance,
     common_ap$peak_score,
     xlab = 'Distance',
     ylab = 'Peak Score')
title('(A)')

# make labels, colors and gorups
labs <- c('competitive', 'none', 'cooperative')
cols <- c('green', 'gray', 'red')

# make three groups by quantiles
common_groups <- cut(common_dt$stat,
                     breaks = quantile(common_dt$stat, c(0, .1, .9, 1)),
                     labels = labs)

# plot predicted function
plot_predictions(common_dt$score_rank,
                 group = common_groups,
                 colors = cols,
                 labels = labs,
                 xlab = 'Regulatory Potential',
                 ylab = 'ECDF')
title('(B)')
```

```
# test factors are cooperative
test_predictions(common_dt$score_rank,
                 group = common_groups,
```

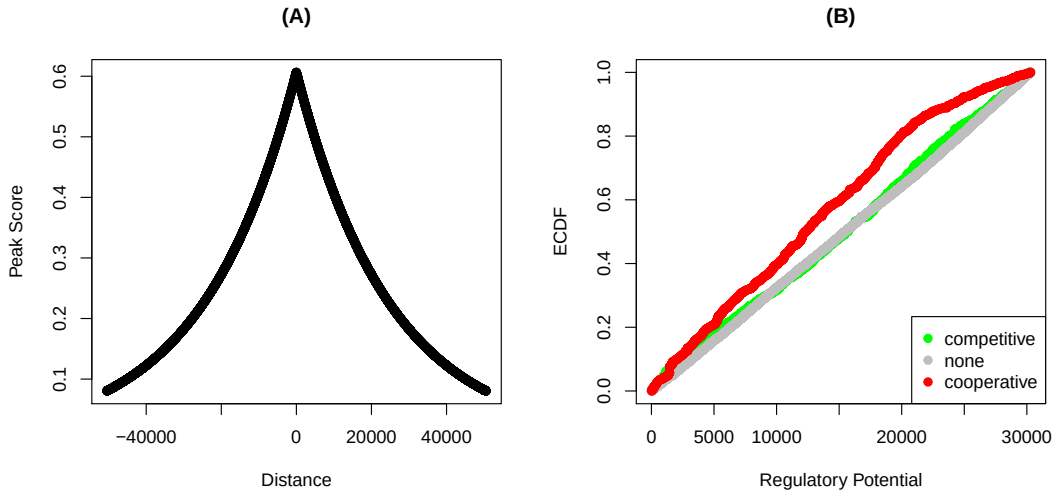

Figure 3. Predicted function of YY1 and YY2 on their shared targets.

```
                compare = c('cooperative', 'none'),
                alternative = 'greater')

# test factors are more cooperative than competitive
test_predictions(common_dt$score_rank,
                group = common_groups,
                compare = c('cooperative', 'competitive'),
                alternative = 'greater')
```

Table 3. Testing for statistical significance of combined functions of the two factors.

Compare	Satistic	Pvalue	Method	Alternative
Coop vs None	0.168	1.5e-30	KS test	The CDF of x lies above that of y
Coop vs Comp	0.151	2.2e-16	KS test	The CDF of x lies above that of y

Table 3

Binding motif analysis

```
# group peaks by their assigned targets
peak_groups <- split(common_dt$tx_id, common_groups)

# reorder peaks and get top n peaks
peak_groups <- lapply(peak_groups, function(x) {
    # get peaks in x targets group
    p <- common_ap[common_ap$assigned_region %in% unique(x)]

    # order peaks by score
    p <- p[order(p$peak_score, decreasing = TRUE)]

    # get n top peaks
    #n <- length(p)
    n <- 100
    p[1:n]
})
```

```
bcout <- map(peak_groups[c('competitive', 'cooperative')],
             ~{
               # make a temporary file
```

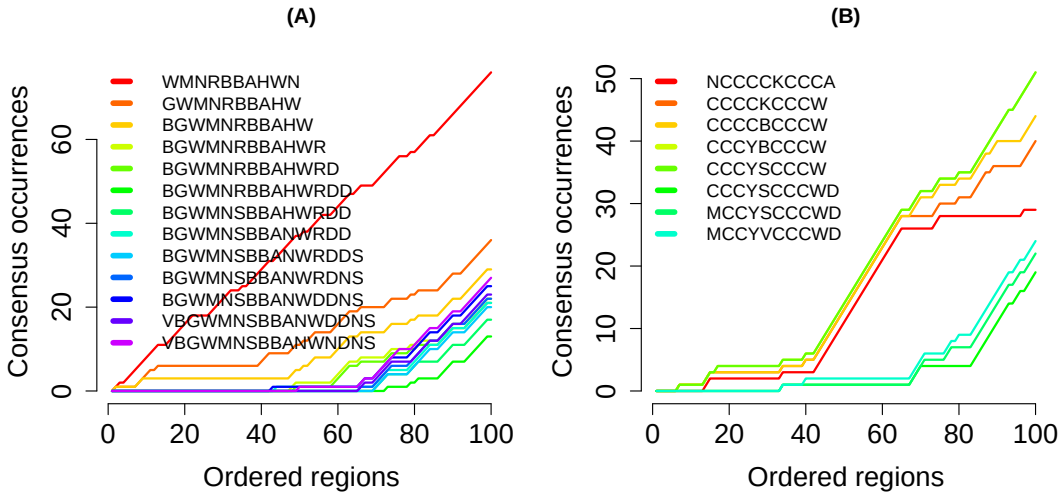

Figure 4. Occurences of consensus sequences in the ranked regions.

```
    tmp_fasta <- tempfile()

    # extract sequences of top peaks from the hg19 genome
    pseq <- getSeq(BSgenome.Hsapiens.UCSC.hg19,
                   names = .x)

    # write sequences to fasta file
    writeXStringSet(pseq, tmp_fasta)

    # set random see
    set.seed(123)

    # call bcrank with the fasta file
    bcrank(tmp_fasta, silent = TRUE)
  })
```

Figure 4

```
par(mfrow = c(1, 2))

# plot the occurences of consensus sequesnce in the regions
map2(bcout, c('(A)', '(B)'),
     ~{
       plot(toptable(.x, 1))
       title(.y)
     })
```

```
## $competitive
## NULL
##
## $cooperative
## NULL
```

Figure 5

```
map(bcout, c('(A)', '(B)'),
    ~{
      seqLogo(pwm(toptable(.x, 1)))
      title(.y)
    })
```

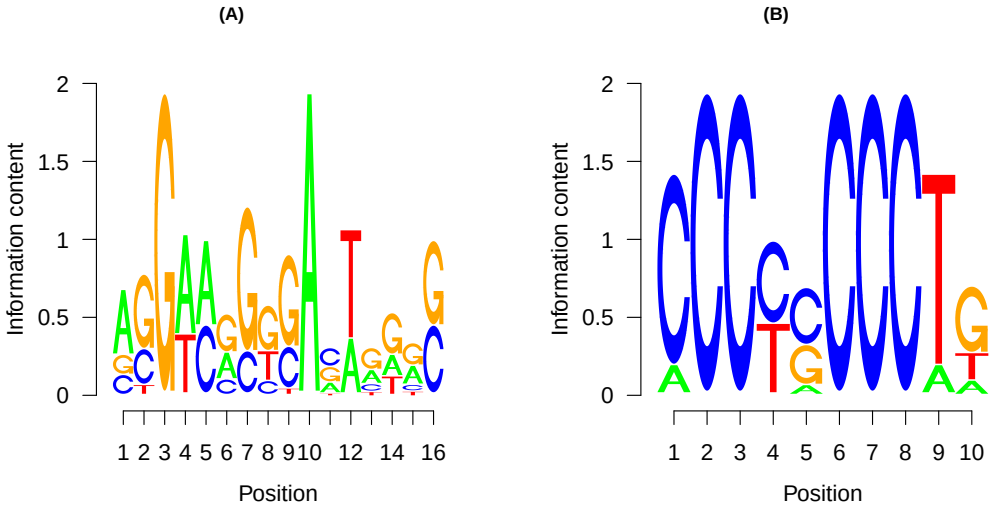

Figure 5. Predicted motifs of the cooperative and competitive binding sites.

Summary

This section is required if the paper does not include novel data or analyses. It allows authors to briefly summarize the key points from the article.

Software availability

This section will be generated by the Editorial Office before publication. Authors are asked to provide some initial information to assist the Editorial Office, as detailed below.

1. URL link to where the software can be downloaded from or used by a non-coder (AUTHOR TO PROVIDE; optional)
2. URL link to the author's version control system repository containing the source code (AUTHOR TO PROVIDE; required)
3. Link to source code as at time of publication (*F1000Research* TO GENERATE)
4. Link to archived source code as at time of publication (*F1000Research* TO GENERATE)
5. Software license (AUTHOR TO PROVIDE; required)

Author information

In order to give appropriate credit to each author of an article, the individual contributions of each author to the manuscript should be detailed in this section. We recommend using author initials and then stating briefly how they contributed.

Competing interests

All financial, personal, or professional competing interests for any of the authors that could be construed to unduly influence the content of the article must be disclosed and will be displayed alongside the article. If there are no relevant competing interests to declare, please add the following: 'No competing interests were disclosed'.

Grant information

Please state who funded the work discussed in this article, whether it is your employer, a grant funder etc. Please do not list funding that you have that is not relevant to this specific piece of research. For each funder, please state the funder's name, the grant number where applicable, and the individual to whom the grant was assigned. If your work was not funded by any grants, please include the line: 'The author(s) declared that no grants were involved in supporting this work.'

Acknowledgments

This section should acknowledge anyone who contributed to the research or the article but who does not qualify as an author based on the criteria provided earlier (e.g. someone or an organization that provided writing assistance). Please state how they contributed; authors should obtain permission to acknowledge from all those mentioned in the Acknowledgments section.

Please do not list grant funding in this section.

References

[1] Li Chen, Toshi Shioda, Kathryn R. Coser, Mary C. Lynch, Chuanwei Yang, and Emmett V. Schmidt. Genome-wide analysis of YY2 versus YY1 target genes. *Nucleic Acids Research*, 38(12):4011–4026, 2010. ISSN 03051048. doi: 10.1093/nar/gkq112.

[2] Joëlle Michaud, Viviane Praz, Nicole James Faresse, Courtney K Jnbaptiste, Shweta Tyagi, Frédéric Schütz, and Winship Herr. HCFC1 is a common component of active human CpG-island promoters and coincides with ZNF143, THAP11, YY1, and GABP transcription factor occupancy. *Genome research*, 23(6):907–16, jun 2013. ISSN 1549-5469. doi: 10.1101/gr.150078.112. URL `http://www.ncbi.nlm.nih.gov/pubmed/23539139`.

[3] Xiao-Nan Wu, Tao-Tao Shi, Yao-Hui He, Fei-Fei Wang, Rui Sang, Jian-Cheng Ding, Wen-Juan Zhang, Xing-Yi Shu, Hai-Feng Shen, Jia Yi, Xiang Gao, and Wen Liu. Methylation of transcription factor YY2 regulates its transcriptional activity and cell proliferation. *Cell discovery*, 3:17035, 2017. ISSN 2056-5968. doi: 10.1038/celldisc.2017.35. URL `http://www.ncbi.nlm.nih.gov/pubmed/29098080`.