Figure 4A and S7 Paused Pol II half-lives and core promoter elements

Wanqing Shao(was@stowers.org)

Contents

Description

To analyze the relationship between the three TFIID-bound regions and Pol II pausing, we first performed a genome-wide analysis of their co-occurrence in the genome. We scored each core promoter element based on the presence of consensus sequence at the expected promoter position, allowing for 1 mismatch: TATA box (STATAWAWR), Inr (TCAKTY), and pausing elements (CSARCSSA, KCGGTTSK or KCGRWCG). We then analyzed how their presence (alone or in pairwise combination) is correlated with the half-lives of paused Pol II that we measured previously (Shao and Zeitlinger, 2017).

Data from 2017 NG paper are downloaded from https://github.com/zeitlingerlab/Shao_NG_2017/tree/master/rdata and stored at /data/rdata.

half_life_df.RData File containing half-life information.

dm3_mrna_unique_tss.RData File containing transcription start site information for dm3 genome.

Enviroment setup

```
library(GenomicRanges, warn.conflicts=F)
library(magrittr)
library(Biostrings)
library(BSgenome.Dmelanogaster.UCSC.dm3)
library(seqLogo)
library(reshape2)

setwd("/data/analysis_code")
options(knitr.figure_dir =
          "Figure4A_S7_core_promoter_elements_and_paused_polii_halflives"
        )

source("shared_code/knitr_common.r")
source("shared_code/ggplot_common.r")
source("shared_code/granges_common.r")
```

Analysis

Genome-wide relationship between core promoter elements and paused Pol II half-life

In our previous paper we measured the half-life of paused Pol II in *Drosophila melanogaster* Kc167 cells. In this analysis, we will re-analyze out half-life data and focusing showing the relationship between core promoter elements and paused Pol II half-life

Checking core promoter elements

```
half_life_df <- get(load("rdata/half_life_df.RData"))
tss <- get(load("rdata/dme_mrna_unique_tss.RData"))

find_motif <- function(motif_name, fb_t_id,  mismatch=0) {

    motif_info <- subset(promoter_table, name == motif_name)
    motif <- DNAString(motif_info$motif)
    up_dis <- motif_info$window_start
    down_dis <- motif_info$window_end

    gene_tss <- tss[tss$fb_t_id %in% fb_t_id]

    if(up_dis >= 0 & down_dis >=0){
      tss_r <- resize(gene_tss, down_dis, "start") %>%
               resize(., down_dis - up_dis, "end")
    }
    if(up_dis < 0 & down_dis >=0){
      tss_r <- resize(gene_tss, down_dis, "start") %>%
               resize(., abs(up_dis)+down_dis, "end")
    }
    if(up_dis < 0 & down_dis <0){
      tss_r <- resize(gene_tss, abs(up_dis), "end") %>%
               resize(., abs(up_dis)-abs(down_dis), "start")
    }

    promoter_seq <- getSeq(Dmelanogaster, tss_r)
    names(promoter_seq) <- tss_r$fb_t_id

    count_df <-
      vcountPattern(motif, promoter_seq, fixed = FALSE,
                    min.mismatch = 0, max.mismatch = mismatch) %>%
      data.frame(fb_t_id = fb_t_id, count =.)

    count_df$count <- ifelse(count_df$count >0, T, F)
    colnames(count_df)[2] <- motif_name
    count_df
}

promoter_table <- read.table("promoter_elements.txt", header=T)
motifs <- promoter_table$name
half_life_tss <- tss[tss$fb_t_id %in% half_life_df$fb_t_id]
```

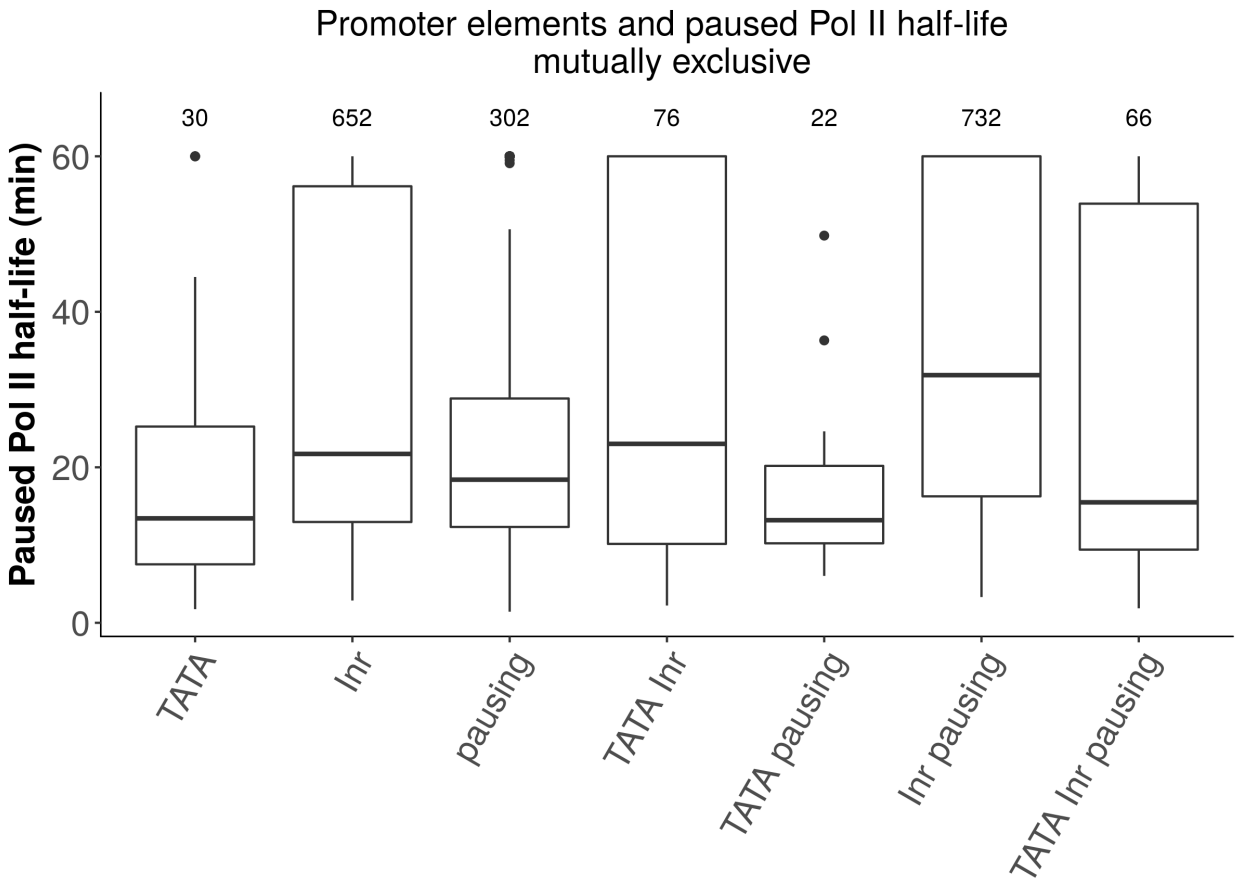
```
motif_list <- lapply(as.character(motifs), function(x){
  motif <- find_motif(motif_name=x, half_life_tss$fb_t_id, mismatch = 1)
  motif
  })

motif_df <- reshape::merge_recurse(motif_list)
all_info_df <- merge(half_life_df, motif_df)
all_info_df$half_life <- ifelse(all_info_df$half_life >=0 &
                                  all_info_df$half_life<= 60,
                                all_info_df$half_life, 60)
```

Mutually exclusive model

The following analysis specifies that promoters in groups are mutually exclusive, e.g. TATA-Inr group (n = 76) means 76 promoters have TATA and Inr but not pausing elements.

```
new_info_df <- with(all_info_df,
                    data.frame(fb_t_id = fb_t_id, gene = gene,
                               half_life = half_life, TATA = TATA,
                               Inr = Inr, pausing_elements = DPE | MTE | PB))

tata <- subset(new_info_df, TATA & !(Inr | pausing_elements)) %>%
  data.frame(type = "TATA")
inr <- subset(new_info_df, Inr& !(TATA | pausing_elements))%>%
  data.frame(type = "Inr")
pausing_element <- subset(new_info_df,  pausing_elements & !(TATA | Inr)) %>%
  data.frame(type = "pausing")

tata_inr <- subset(new_info_df, TATA & Inr & !(pausing_elements))%>%
  data.frame(type = "TATA Inr")
tata_pausing <- subset(new_info_df, TATA & pausing_elements & !(Inr) )%>%
  data.frame(type = "TATA pausing")
inr_pausing <-  subset(new_info_df, Inr & pausing_elements & !(TATA))%>%
  data.frame(type = "Inr pausing")
tata_inr_pausing <-  subset(new_info_df, Inr & pausing_elements & TATA)%>%
  data.frame(type = "TATA Inr pausing")

combined_df <- rbind(tata, inr, pausing_element,
                     tata_inr, tata_pausing, inr_pausing,
                     tata_inr_pausing)


half_life_boxplot <- function(combined_df, title){
  count_info <- table(combined_df$type) %>% as.data.frame()
  ggplot(combined_df, aes(x = type, y = half_life)) +
  geom_boxplot() +
  geom_text(data = count_info, aes(x = count_info$Var1, label = Freq, y = 65),
              position = position_dodge(width = .75),
              show.legend = FALSE )+
  theme(axis.text.x = element_text(angle = 60, hjust = 1)) +
  ylab("Paused Pol II half-life (min)") +
  xlab("") +
  ggtitle(title)
```

```
}

half_life_boxplot(combined_df,
        "Promoter elements and paused Pol II half-life \n mutually exclusive")
```




```
pairwise_type <- unique(combined_df$type) %>%
  as.character(.) %>%
  combn(., 2, simplify = F)

calc_pval <- function(type1, type2, df){
  type1_df <- subset(df, type == type1)
  type2_df <- subset(df, type == type2)
  pval = wilcox.test(type1_df$half_life, type2_df$half_life)$ p.value
  data.frame(compare = paste(type1, "vs.", type2),
             pval = pval)
}

pval_df <- lapply(pairwise_type, function(x)calc_pval(x[1], x[2], combined_df)) %>%
  do.call(rbind, .)
pander(pval_df)
```

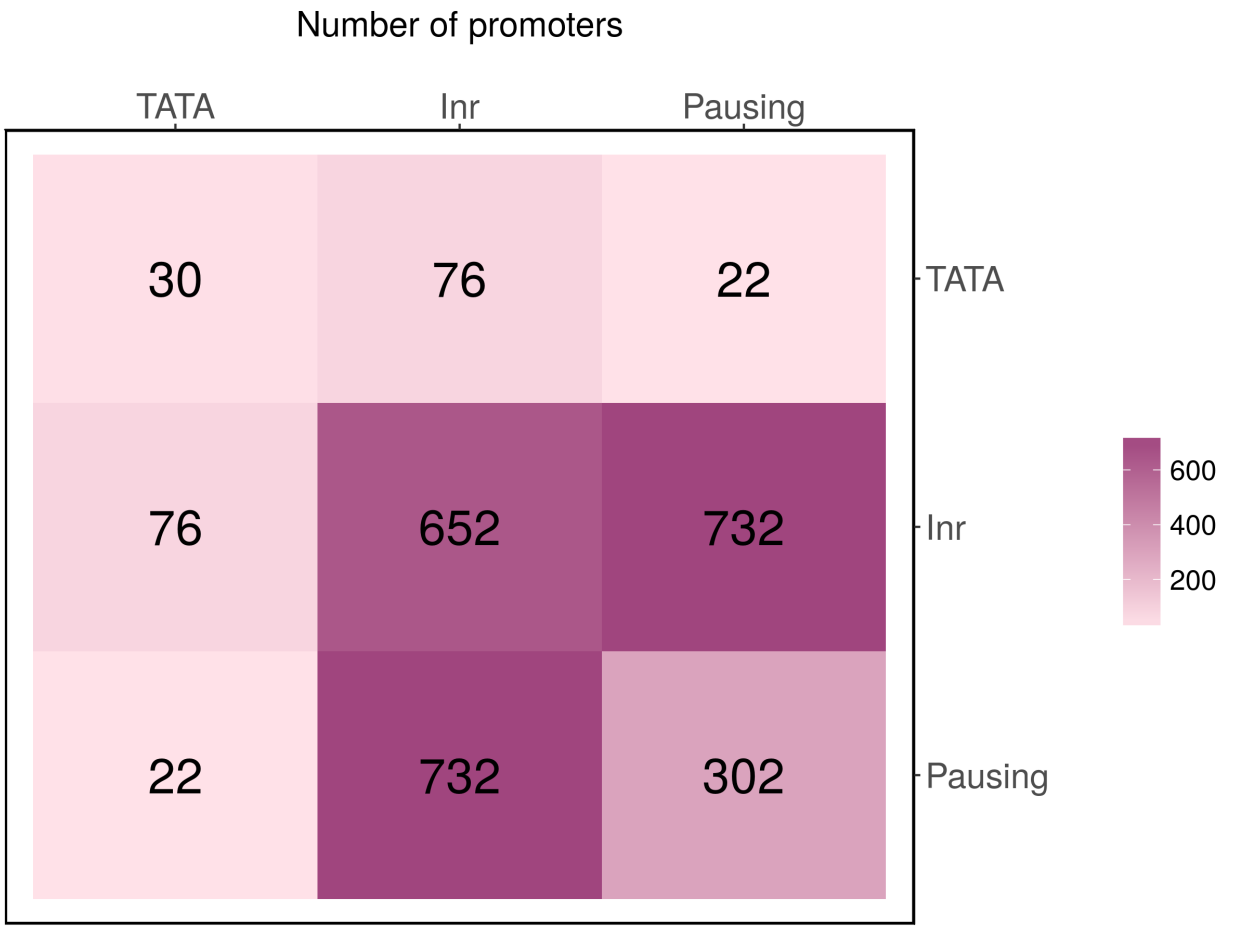
compare	pval
TATA vs. Inr	0.001213

compare	pval
TATA vs. pausing	0.0167
TATA vs. TATA Inr	0.03463
TATA vs. TATA pausing	0.9926
TATA vs. Inr pausing	8.541e-06
TATA vs. TATA Inr pausing	0.1467
Inr vs. pausing	0.001445
Inr vs. TATA Inr	0.5756
Inr vs. TATA pausing	0.000986
Inr vs. Inr pausing	3.298e-07
Inr vs. TATA Inr pausing	0.05875
pausing vs. TATA Inr	0.4081
pausing vs. TATA pausing	0.01107
pausing vs. Inr pausing	3.996e-14
pausing vs. TATA Inr pausing	0.611
TATA Inr vs. TATA pausing	0.03769
TATA Inr vs. Inr pausing	0.01467
TATA Inr vs. TATA Inr pausing	0.3571
TATA pausing vs. Inr pausing	7.429e-06
TATA pausing vs. TATA Inr pausing	0.1542
Inr pausing vs. TATA Inr pausing	0.000317

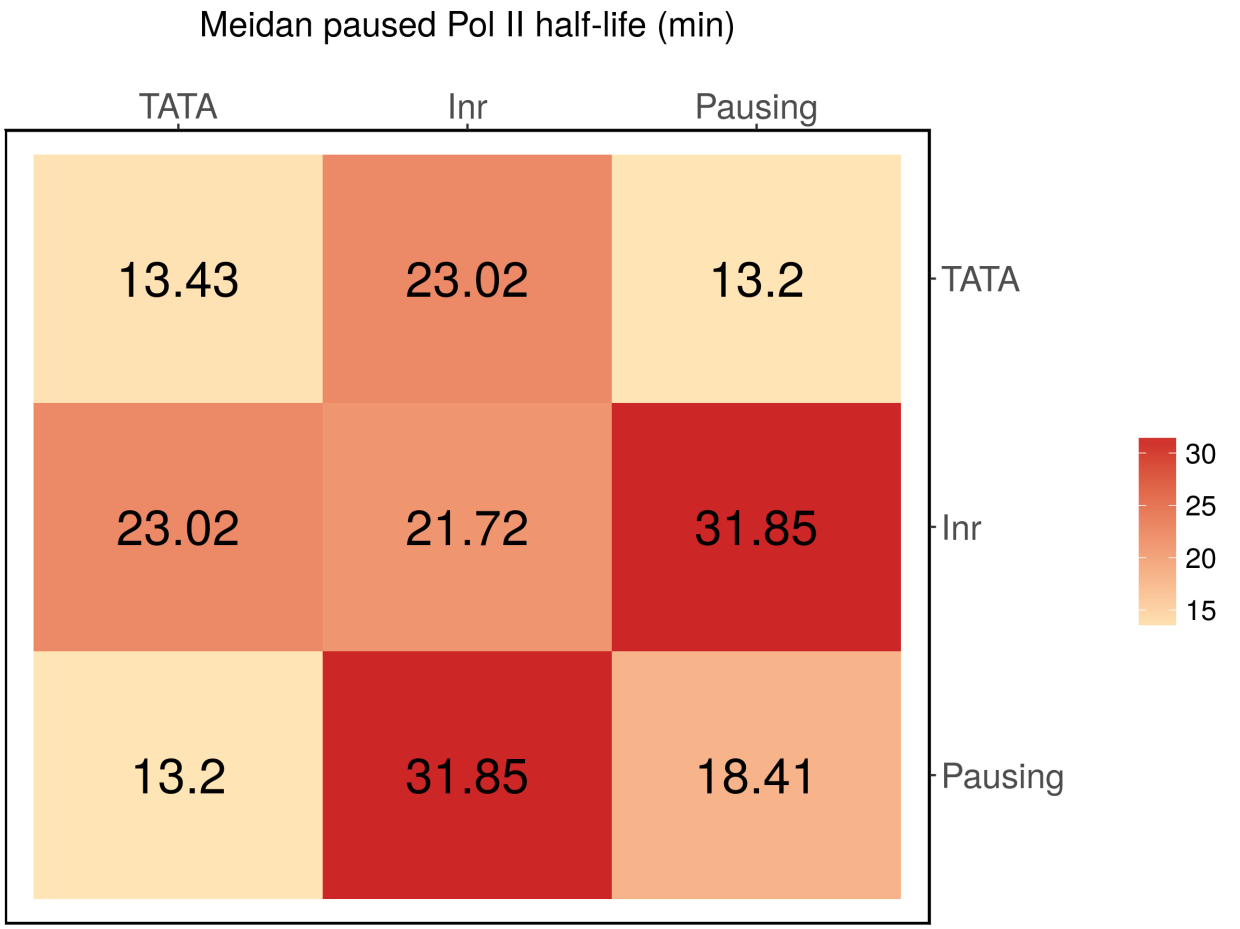
```
write.csv(pval_df, file =
  "promoter_element_combination_halflife_wilcox_test_mutually_exclusive.csv")
```

```
info_list <- list(TATA_TATA = tata,
                  Inr_Inr = inr,
                  Pausing_Pausing = pausing_element,
                  TATA_Inr = tata_inr,
                  TATA_Pausing = tata_pausing,
                  Inr_Pausing = inr_pausing)

generate_pairwise_df <- function(info_list){
  pairwise_df <- lapply(names(info_list), function(x){
  info_df <- data.frame(motif1 = gsub("_.*", "", x),
                        motif2 = gsub(".*_", "", x),
                        half_life = median(info_list[[x]]$half_life),
                        count = nrow(info_list[[x]]))
  info_df
  }) %>% do.call(rbind, .)
  pairwise_df_flip <- pairwise_df
  pairwise_df_flip$motif1 <- pairwise_df$motif2
  pairwise_df_flip$motif2 <- pairwise_df$motif1

  pairwise_df <- rbind(pairwise_df,pairwise_df_flip ) %>% unique()
  pairwise_df
}


promoter_number_heatmap <- function(pairwise_df){
  ggplot(pairwise_df, aes(x = motif1, y = motif2, fill = count)) +
  scale_fill_gradient(low="#ffe1e8", high= "#a0457e") +
  geom_tile() + xlab("") + ylab("") +
```

```
  geom_text(aes(label=count), color="black", size=8) +
  scale_x_discrete(position = "top") +
  scale_y_discrete(position = "right") +
  ggtitle("Number of promoters") +
  theme(panel.border = element_rect( colour = "black", fill = NA, size=1))
}

half_life_heatmap <- function(pairwise_df){
  ggplot(pairwise_df, aes(x = motif1, y = motif2, fill = half_life)) +
  scale_fill_gradient(low="moccasin", high= "firebrick3") +
  geom_tile() + xlab("") + ylab("") +
  geom_text(aes(label=round(half_life, digits = 2)), color="black", size=8) +
  scale_x_discrete(position = "top") +
  scale_y_discrete(position = "right") +
  ggtitle("Meidan paused Pol II half-life (min)") +
  theme(panel.border = element_rect( colour = "black", fill = NA, size=1))
}

pairwise_df <- generate_pairwise_df(info_list)
pairwise_df$motif1 <- factor(pairwise_df$motif1,
                             levels = c("TATA", "Inr", "Pausing"))
pairwise_df$motif2 <- factor(pairwise_df$motif2,
                             levels = rev(c("TATA", "Inr", "Pausing")))

promoter_number_heatmap(pairwise_df)
```




```
half_life_heatmap(pairwise_df)
```

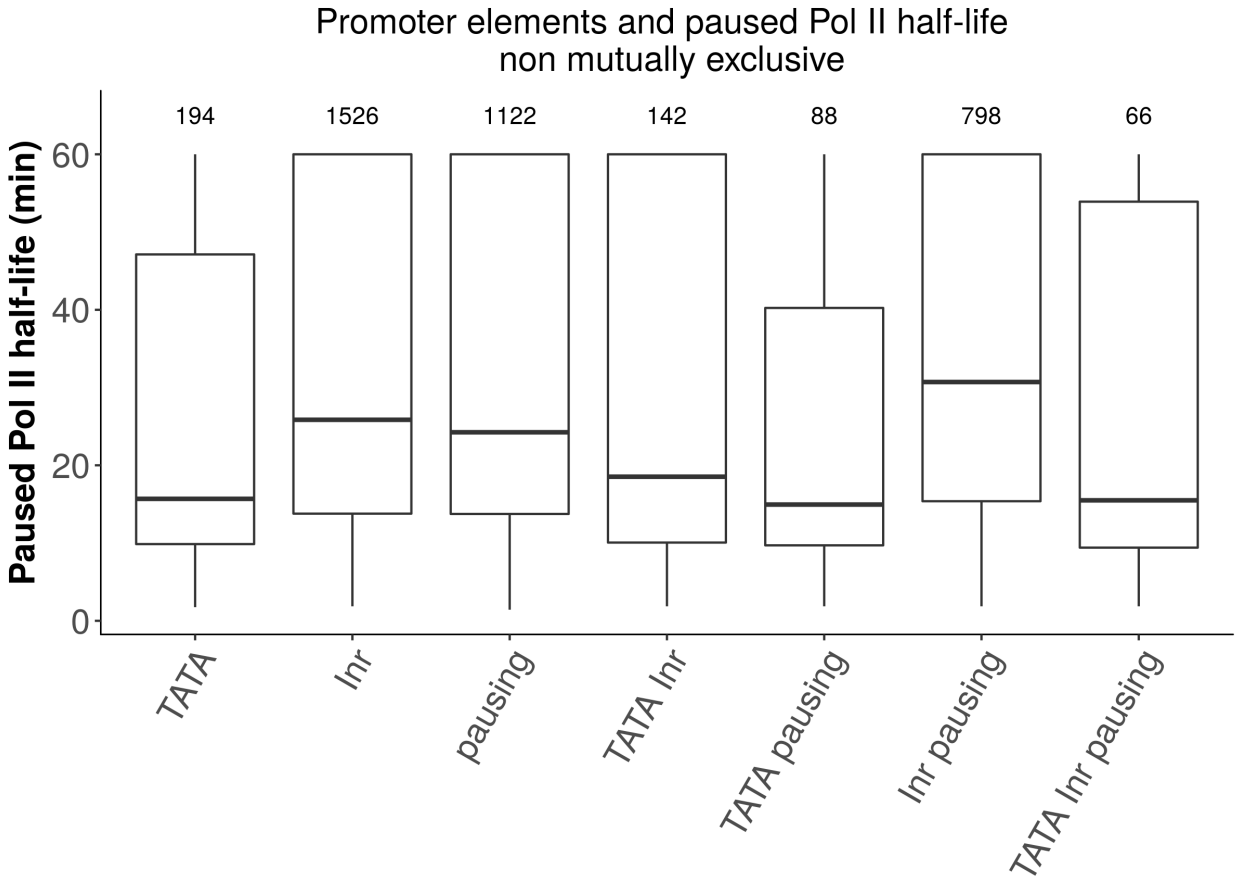


Non-mutually exclusive model

The following analysis allows other core promoter emelements besides the spefied one, , e.g. TATA-Inr group (n = 142) means 142 promoters have TATA and Inr and some of them can also have pausing elements.

```
tata <- subset(new_info_df, TATA )%>%
  data.frame(type = "TATA")
inr <- subset(new_info_df, Inr)%>%
  data.frame(type = "Inr")
pausing_element <- subset(new_info_df,  pausing_elements )%>%
  data.frame(type = "pausing")

tata_inr <- subset(new_info_df, TATA & Inr )%>%
  data.frame(type = "TATA Inr")
tata_pausing <- subset(new_info_df, TATA & pausing_elements  )%>%
  data.frame(type = "TATA pausing")
inr_pausing <-  subset(new_info_df, Inr & pausing_elements )%>%
  data.frame(type = "Inr pausing")


combined_df <- rbind(tata, inr, pausing_element,
                     tata_inr, tata_pausing, inr_pausing,
                     tata_inr_pausing)
```

```
half_life_boxplot(combined_df,
  "Promoter elements and paused Pol II half-life \n non mutually exclusive")
```

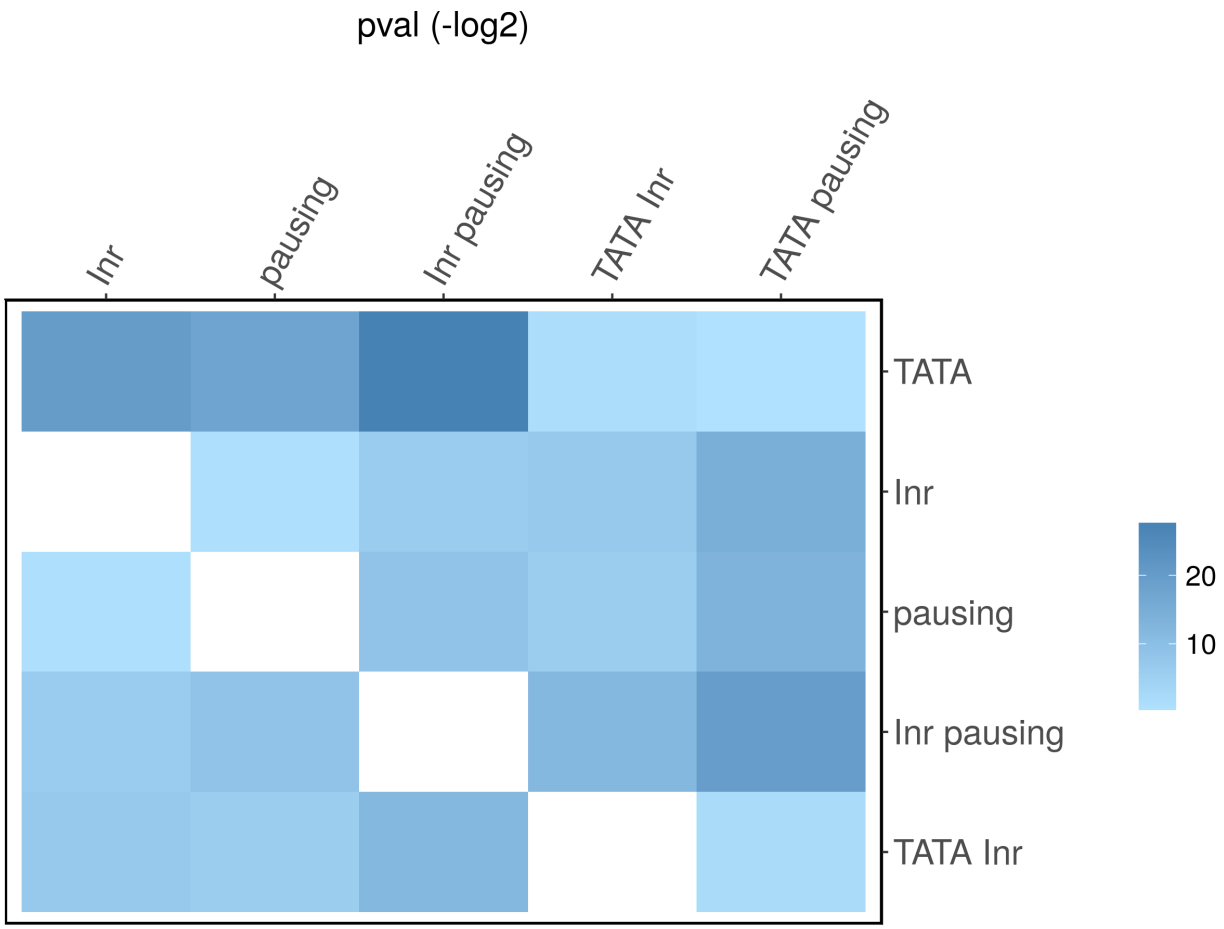



```
pairwise_type <- unique(combined_df$type) %>%
  as.character(.) %>%
  combn(., 2, simplify = F)

pval_df <- lapply(pairwise_type, function(x)calc_pval(x[1], x[2], combined_df)) %>%
  do.call(rbind, .)
pander(pval_df)
```

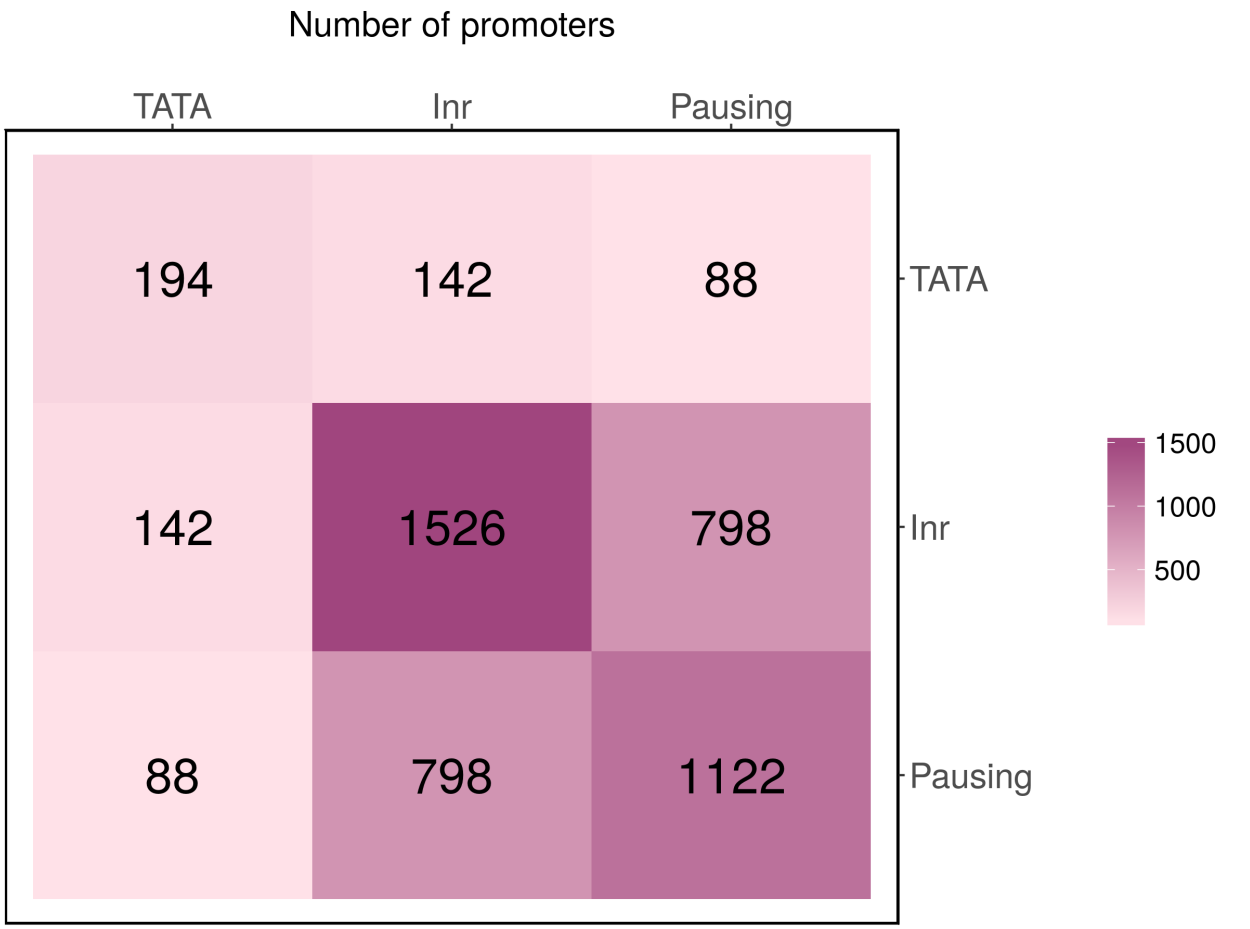
compare	pval
TATA vs. Inr	8.321e-07
TATA vs. pausing	4.347e-06
TATA vs. TATA Inr	0.3175
TATA vs. TATA pausing	0.656
TATA vs. Inr pausing	4.595e-09
TATA vs. TATA Inr pausing	0.8378
Inr vs. pausing	0.4518
Inr vs. TATA Inr	0.005839
Inr vs. TATA pausing	3.7e-05
Inr vs. Inr pausing	0.01037
Inr vs. TATA Inr pausing	0.008028

compare	pval
pausing vs. TATA Inr	0.01323
pausing vs. TATA pausing	9.074e-05
pausing vs. Inr pausing	0.001826
pausing vs. TATA Inr pausing	0.01404
TATA Inr vs. TATA pausing	0.2078
TATA Inr vs. Inr pausing	0.0002447
TATA Inr vs. TATA Inr pausing	0.5778
TATA pausing vs. Inr pausing	1.129e-06
TATA pausing vs. TATA Inr pausing	0.5888
Inr pausing vs. TATA Inr pausing	0.0009331

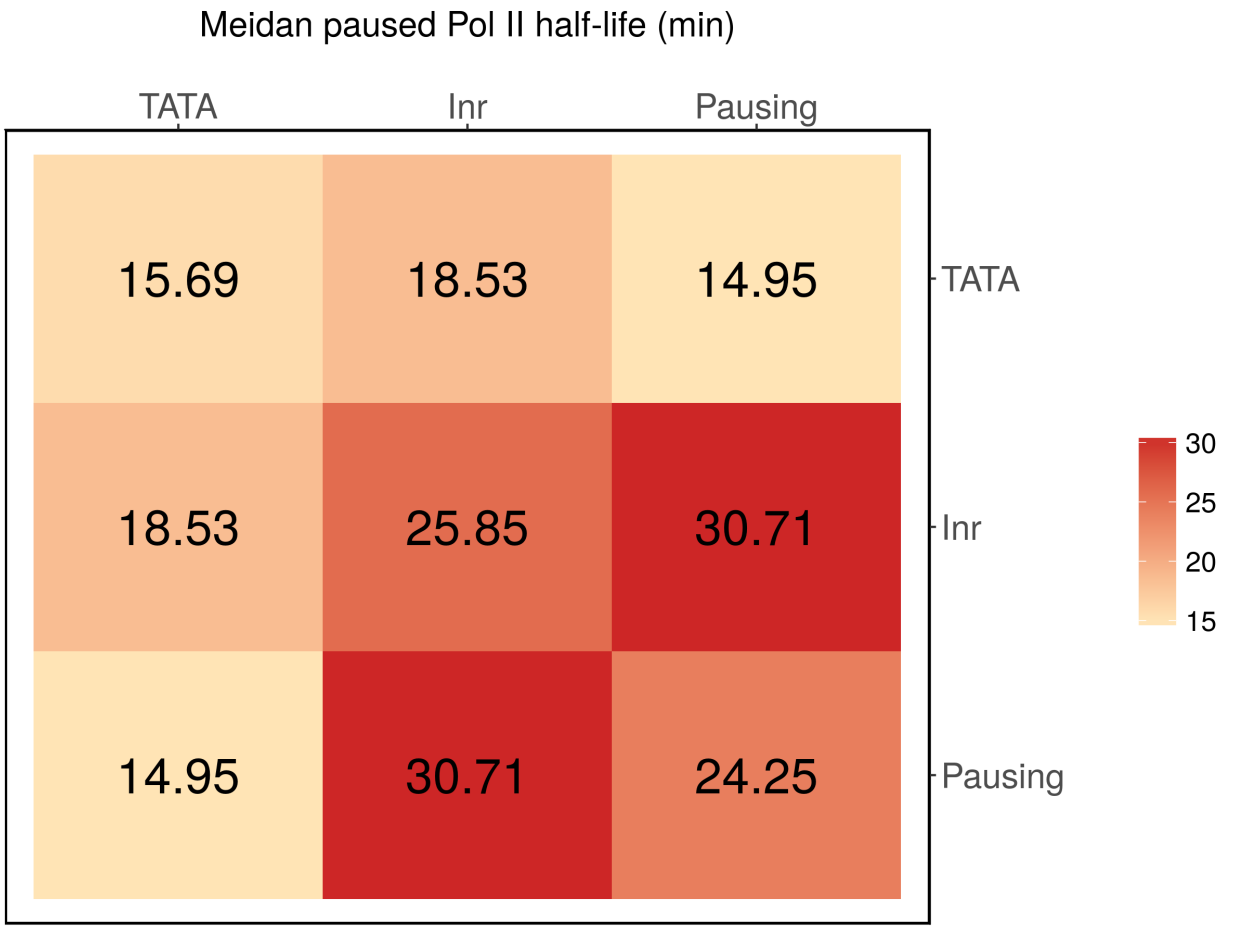
```
write.csv(pval_df, file =
  "promoter_element_combination_halflife_wilcox_test_non_mutually_exclusive.csv")

pval_df$group1 <- gsub(" vs.*", "",pval_df$compare)
pval_df$group2 <- gsub(".*vs. ", "",pval_df$compare)
pval_df_flip <- pval_df
pval_df_flip$group1 <- pval_df$group2
pval_df_flip$group2 <- pval_df$group1

pval_df <- rbind(pval_df,pval_df_flip ) %>% unique()

pval_df <- subset(pval_df, !group1 %in% c("TATA Inr pausing", "TATA")&
                    !group2 %in% c("TATA pausing", "TATA Inr pausing"))

pval_df$group1 <-
  factor(pval_df$group1, levels = c("Inr", "pausing", "Inr pausing",
                                    "TATA Inr", "TATA pausing"))
pval_df$group2 <-
  factor(pval_df$group2, levels = rev(c("TATA", "Inr", "pausing", "Inr pausing",
                                        "TATA Inr")))


ggplot(pval_df, aes(x = group1, y = group2, fill = -1 * log2(pval))) +
  scale_fill_gradient(low="lightskyblue1", high= "steelblue") +
  geom_tile() + xlab("") + ylab("") +
  #geom_text(aes(label=round(half_life, digits = 2)), color="black", size=8) +
  scale_x_discrete(position = "top") +
  scale_y_discrete(position = "right") +
  ggtitle("pval (-log2)") +
  theme(panel.border = element_rect( colour = "black", fill = NA, size=1),
        axis.text.x = element_text(angle = 60, hjust = 0))
```




```
info_list <- list(TATA_TATA = tata,
                  Inr_Inr = inr,
                  Pausing_Pausing = pausing_element,
                  TATA_Inr = tata_inr,
                  TATA_Pausing = tata_pausing,
                  Inr_Pausing = inr_pausing)

pairwise_df <- generate_pairwise_df(info_list)
pairwise_df$motif1 <-
  factor(pairwise_df$motif1, levels = c("TATA", "Inr", "Pausing"))
pairwise_df$motif2 <-
  factor(pairwise_df$motif2, levels = rev(c("TATA", "Inr", "Pausing")))

promoter_number_heatmap(pairwise_df)
```




```
half_life_heatmap(pairwise_df)
```



SessionInfo

This analysis was performed with the following R/Bioconductor session:

```
sessionInfo()
```

```
## R version 3.4.4 (2018-03-15)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: Ubuntu 16.04.4 LTS
##
## Matrix products: default
## BLAS: /usr/lib/libblas/libblas.so.3.6.0
## LAPACK: /usr/lib/lapack/liblapack.so.3.6.0
##
## locale:
##  [1] LC_CTYPE=en_US.UTF-8       LC_NUMERIC=C
##  [3] LC_TIME=en_US.UTF-8        LC_COLLATE=en_US.UTF-8
##  [5] LC_MONETARY=en_US.UTF-8    LC_MESSAGES=en_US.UTF-8
##  [7] LC_PAPER=en_US.UTF-8       LC_NAME=C
##  [9] LC_ADDRESS=C               LC_TELEPHONE=C
## [11] LC_MEASUREMENT=en_US.UTF-8 LC_IDENTIFICATION=C
##
## attached base packages:
```

```
##  [1] grid      parallel  stats4    stats     graphics  grDevices utils
##  [8] datasets  methods   base
##
## other attached packages:
##  [1] ggplot2_2.2.1                        pander_0.6.1
##  [3] reshape2_1.4.3                       seqLogo_1.44.0
##  [5] BSgenome.Dmelanogaster.UCSC.dm3_1.4.0 BSgenome_1.46.0
##  [7] rtracklayer_1.38.3                   Biostrings_2.46.0
##  [9] XVector_0.18.0                       magrittr_1.5
## [11] GenomicRanges_1.30.3                 GenomeInfoDb_1.14.0
## [13] IRanges_2.12.0                       S4Vectors_0.16.0
## [15] BiocGenerics_0.24.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.17               pillar_1.2.3
##  [3] compiler_3.4.4             plyr_1.8.4
##  [5] bitops_1.0-6               tools_3.4.4
##  [7] zlibbioc_1.24.0            digest_0.6.15
##  [9] tibble_1.4.2               gtable_0.2.0
## [11] evaluate_0.10.1            lattice_0.20-35
## [13] rlang_0.2.1                Matrix_1.2-14
## [15] DelayedArray_0.4.1         yaml_2.1.19
## [17] GenomeInfoDbData_1.0.0     stringr_1.3.1
## [19] knitr_1.20                 rprojroot_1.3-2
## [21] reshape_0.8.7              Biobase_2.38.0
## [23] XML_3.98-1.11              BiocParallel_1.12.0
## [25] rmarkdown_1.10             scales_0.5.0
## [27] backports_1.1.2            Rsamtools_1.30.0
## [29] htmltools_0.3.6            matrixStats_0.53.1
## [31] GenomicAlignments_1.14.2   SummarizedExperiment_1.8.1
## [33] colorspace_1.3-2           labeling_0.3
## [35] stringi_1.2.3              lazyeval_0.2.1
## [37] munsell_0.5.0              RCurl_1.95-4.10
```