Figure 6 Initiator sequence differs between TATA and stably paused promoters

Wanqing Shao(was@stowers.org)

Contents

Description

Both the TATA box and downstream pausing sequences work synergetically with initiator (Inr), however, the TATA box and downstream pausing sequences have opposite effect on Pol II pausing, raising the possibility that some Inr sequences may work better with the TATA box, while others work better with downstream pausing sequences. To identify potential differences, we analyzed the naturally occurring Inr sequences from the TATA-containing promoters versus those of the stably paused promoters. # Enviroment setup

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

setwd("/data/analysis_code")
options(knitr.figure_dir =
          "Figure6_initiator_sequence_differs_between_tata_and_stably_paused_promoters"
        )

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

Analysis

Inr sequence differs at TATA and stably paused promoters

In our 2017 NG paper, we measured the paused Pol II half-life across *Drosophila* Kc167 cell genome. Here, we used the half-life data, separated genes into two groups and tested the promoter sequences at those two groups.

1) genes with TATA box and short Pol II pausing and
2) genes without TATA and show long Pol II pausing.

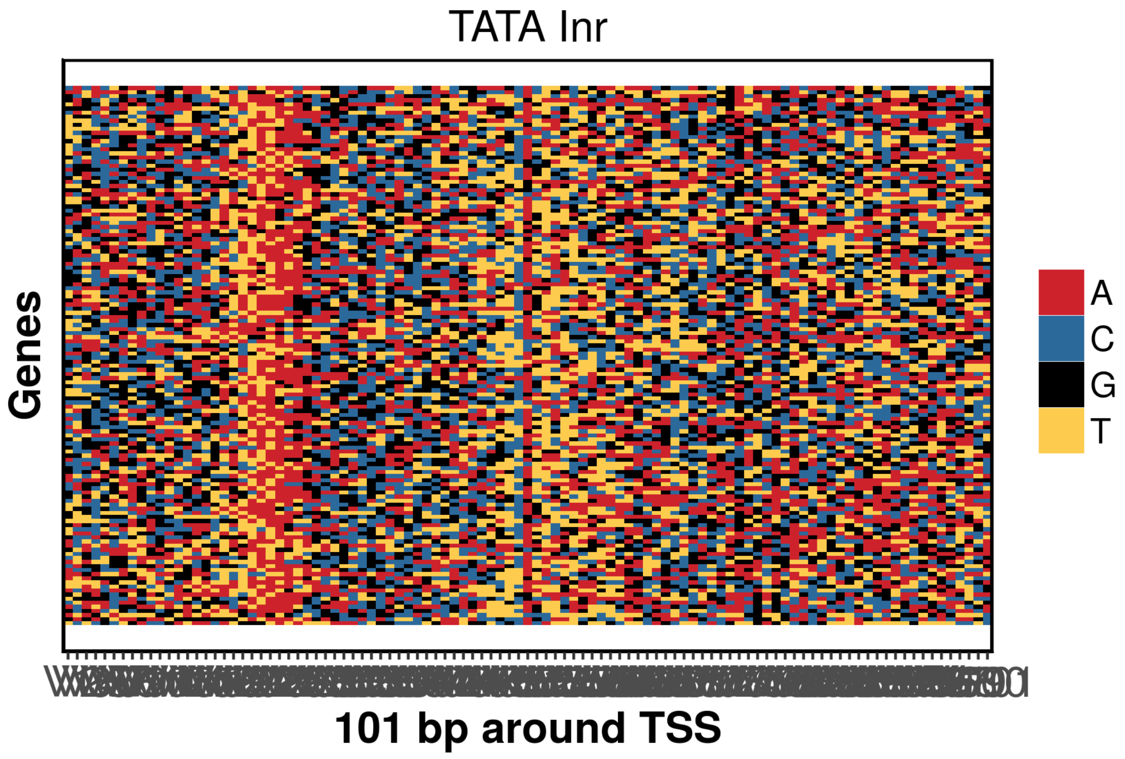
Data from 2017 NG paper are downloaded from https://github.com/zeitlingerlab/Shao_NG_2017/tree/master/rdata and stored at /data/rdata.

half_life_df.RData File containing half-life information.

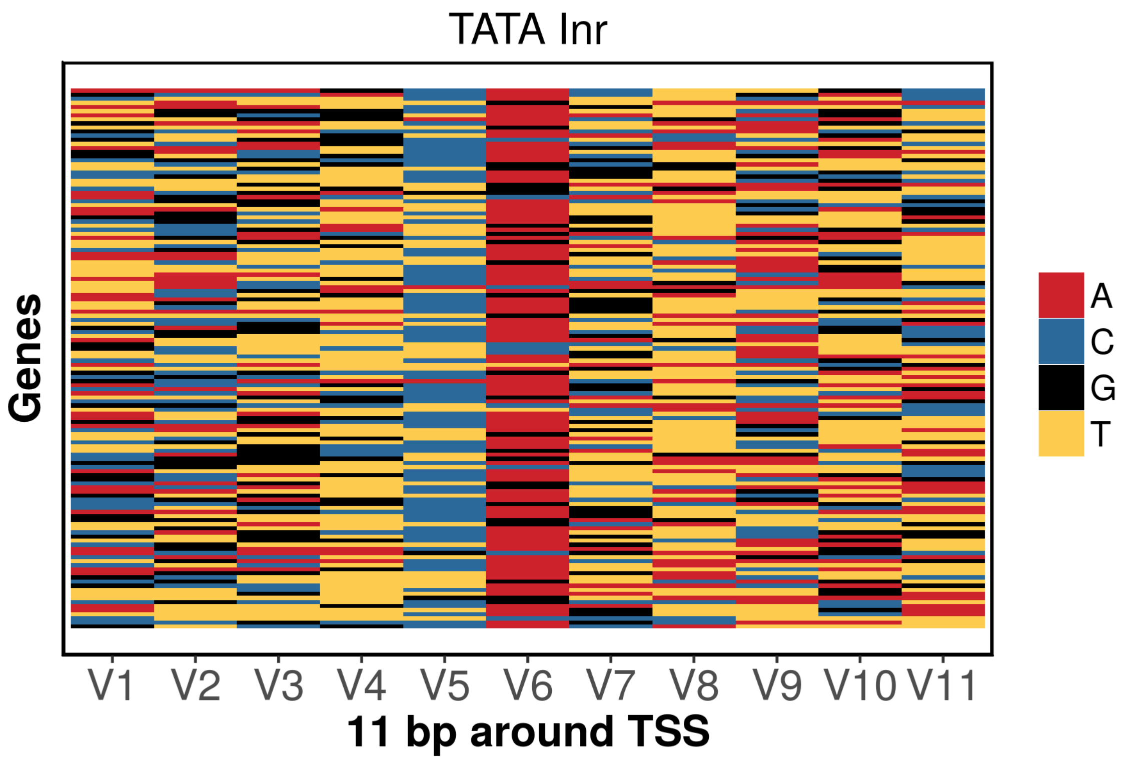
dm3_mrna_unique_tss.RData File containing transcription start site information for dm3 genome.

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

half_life_tss <- tss[tss$fb_t_id %in% half_life_df$fb_t_id]

find_motif <- function(motif_name, motif_seq, window_start,
                       window_end, gene_tss, mismatch=0) {

    motif <- DNAString(motif_seq)

    if(window_start >= 0 & window_end >=0){
      tss_r <- resize(gene_tss, window_end, "start") %>%
               resize(., window_end - window_start, "end")
    }
    if(window_start < 0 & window_end >=0){
      tss_r <- resize(gene_tss, window_end, "start") %>%
               resize(., abs(window_start)+window_end, "end")
    }
    if(window_start < 0 & window_end <0){
      tss_r <- resize(gene_tss, abs(window_start), "end") %>%
               resize(., abs(window_start)-abs(window_end), "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 = tss_r$fb_t_id, count =.)

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

tata_info_df <- find_motif("TATA", "STATAWAWR", -40, -20, half_life_tss, 1)
half_life_df <- merge(half_life_df, tata_info_df)

tata_tss <- tss[tss$fb_t_id %in% subset(half_life_df, TATA == "T" &
                                          half_life <= 30 &
                                          half_life > 0 )$fb_t_id]

pausing_tss <- tss[tss$fb_t_id %in% subset(half_life_df, TATA == "F" &
                                          (half_life >= 60 |
                                           half_life < 0) )$fb_t_id]
```

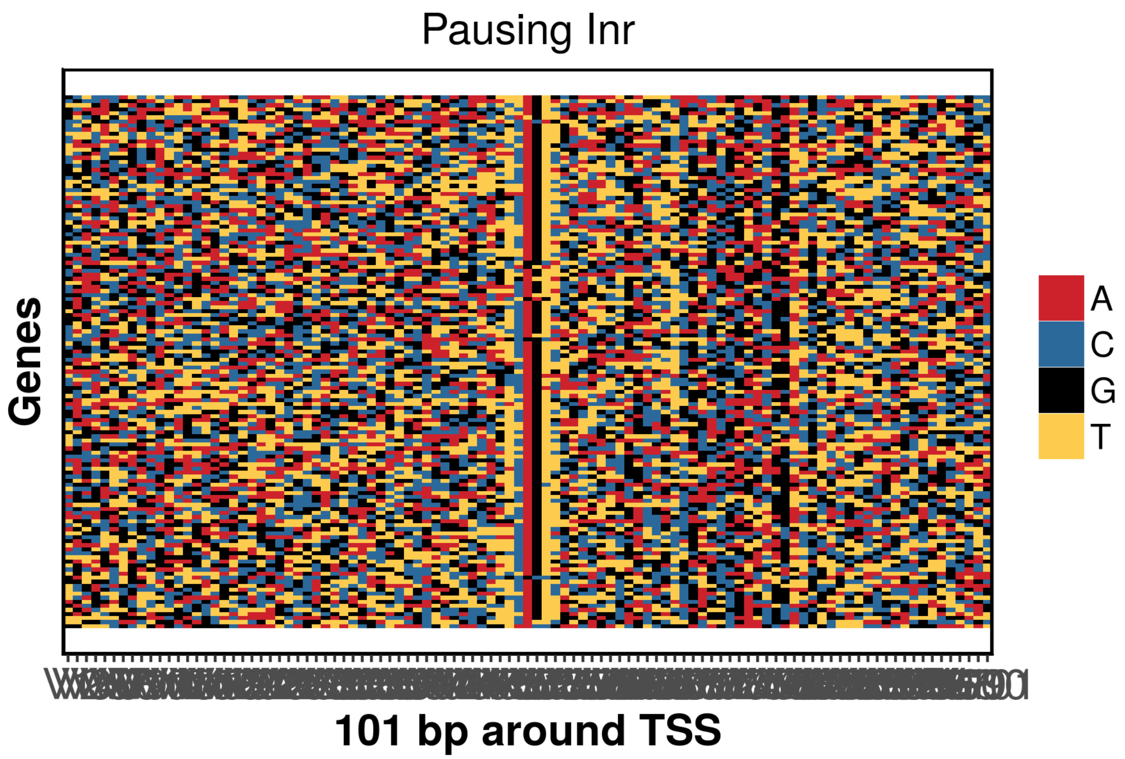
```
get_heatmap <- function(tss, window, name){
  seq <- getSeq(Dmelanogaster, resize(tss, window, "center") )
  seq_df <- as.character(seq) %>% lapply(., function(x)strsplit(x, "")) %>%
    unlist(., recursive = F) %>% do.call(rbind,.)%>% as.data.frame()

  seq_df$id <- 1:nrow(seq_df)
  seq_df_m <- melt(seq_df, id.vars  = "id")

  ATGC_plot <- ggplot(seq_df_m, aes(x= variable, y = id, fill = value)) +
    geom_tile() +
    scale_fill_manual(values = c("#CD222C", "#2C699B", "black", "#FDCC4E")) +
    xlab(paste(window, "bp around TSS")) + ylab("Genes")+
    ggtitle(name)+
    theme(axis.text.y = element_blank(),axis.ticks.y = element_blank(),
          panel.border = element_rect( colour = "black", fill = NA, size=1))
  print(ATGC_plot)
}

nothing <- get_heatmap(tata_tss, 101, "TATA Inr")
```



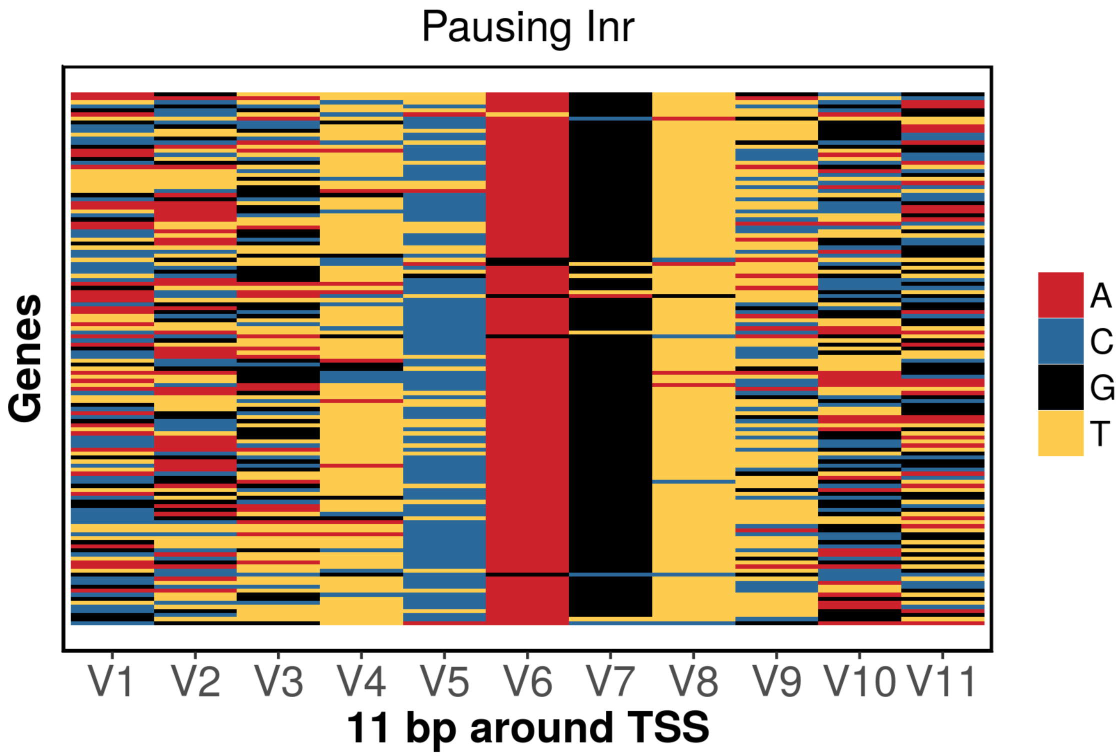
```
nothing <- get_heatmap(tata_tss, 11, "TATA Inr")
```



```r
nothing <- get_heatmap(pausing_tss[1:length(tata_tss)],
                       101, "Pausing Inr")
```

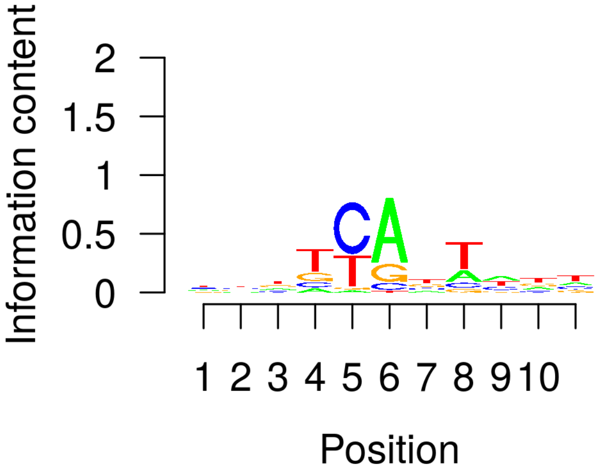


```r
nothing <- get_heatmap(pausing_tss[1:length(tata_tss)],
                       11, "Pausing Inr")
```

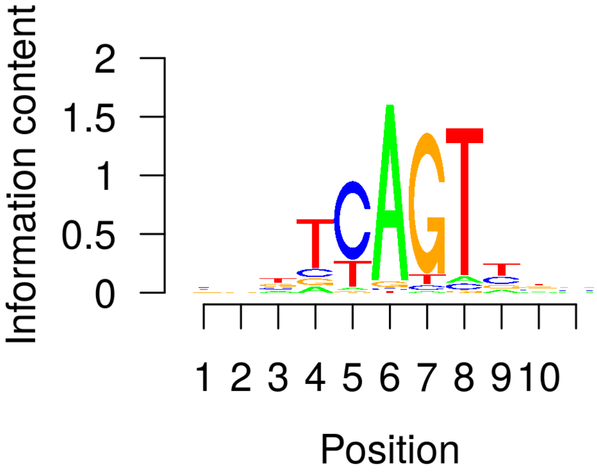


Sequence logo at TATA and stably paused promoters

```
proportion <- function(x){
   rs <- sum(x);
   return(x / rs);
}

get_logo <- function(tss, seq = NULL){
  if(is.null(seq)){
      seq <- getSeq(Dmelanogaster, resize(tss, 11, "center") )
  }
  seq_m <- as.character(seq) %>% lapply(., function(x)strsplit(x, "")) %>%
    unlist(., recursive = F) %>% do.call(rbind,.)
  freq_table <- apply(seq_m, 2, function(x)paste(x, collapse = "")) %>% DNAStringSet() %>% alphabetFreq
  mef2 <- apply(freq_table[, 1:4], 1, proportion)
  pwm <- makePWM(mef2)
  seqLogo(mef2)
  seq_m
}

tata_m <- get_logo(tata_tss)
```



```r
pausing_m <- get_logo(pausing_tss)
```



Statistical test for the occurance of the "G" at the Inr +2 position

```r
table(tata_m[, 7])
```

```
##
##  A  C  G  T
## 20 27 34 51
```

```r
table(pausing_m[, 7])
```

```
##
##   A   C   G   T
##   5  14 441  30
```

```
tata_g_percent <- table(tata_m[,7])["G"]/nrow(tata_m) * 100
pausing_g_percent <- table(pausing_m[,7])["G"]/nrow(pausing_m) * 100

message("G% at TATA promoter ", round(tata_g_percent, digits = 2), " %")
message("G% at pausing promoter ", round(pausing_g_percent, digits = 2), " %")

testing_m <- matrix(c(table(tata_m[,7])["G"], table(pausing_m[,7])["G"],
                      nrow(tata_m) - table(tata_m[,7])["G"],
                      nrow(pausing_m) - table(pausing_m[,7])["G"]),
                      nrow = 2,
                    dimnames = list(c("TATA", "pausing"), c("G", "None G")))

test_result <- fisher.test(testing_m, alternative = "two.sided")
test_result
```

```
##
##  Fisher's Exact Test for Count Data
##
## data:  testing_m
## p-value < 2.2e-16
## alternative hypothesis: true odds ratio is not equal to 1
## 95 percent confidence interval:
##  0.02290752 0.06461657
## sample estimates:
## odds ratio
## 0.03891662
```

```
test_result$p.value
```

```
## [1] 1.948845e-47
```

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] reshape2_1.4.3                      ggplot2_2.2.1
##  [3] pander_0.6.1                        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                      Rmisc_1.5
## [11] plyr_1.8.4                          lattice_0.20-35
## [13] magrittr_1.5                        GenomicRanges_1.30.3
## [15] GenomeInfoDb_1.14.0                 IRanges_2.12.0
## [17] S4Vectors_0.16.0                    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             bitops_1.0-6
##  [5] tools_3.4.4                zlibbioc_1.24.0
##  [7] digest_0.6.15              tibble_1.4.2
##  [9] gtable_0.2.0               evaluate_0.10.1
## [11] rlang_0.2.1                Matrix_1.2-14
## [13] DelayedArray_0.4.1         yaml_2.1.19
## [15] GenomeInfoDbData_1.0.0     stringr_1.3.1
## [17] knitr_1.20                 rprojroot_1.3-2
## [19] Biobase_2.38.0             XML_3.98-1.11
## [21] BiocParallel_1.12.0        rmarkdown_1.10
## [23] scales_0.5.0               backports_1.1.2
## [25] Rsamtools_1.30.0           htmltools_0.3.6
## [27] matrixStats_0.53.1         GenomicAlignments_1.14.2
## [29] SummarizedExperiment_1.8.1 colorspace_1.3-2
## [31] labeling_0.3               stringi_1.2.3
## [33] lazyeval_0.2.1             munsell_0.5.0
## [35] RCurl_1.95-4.10
```