Figure S1

Sergio Garcia-Moreno Alcantara (sga@stowers.org)

August 24, 2021

Aim

The aim of this analysis is to define promoter types in Drosophila ovaries cells based on the presence of specific core promoter elements

Enviroment setup

Set working directory and load required libraries and lab functions

```r
setwd("/n/projects/sga/analysis/SAGA/saga_publication/")
options(knitr.figure_dir = "plots/figure_s1")

# Standard packages
library(GenomicAlignments)
library(GenomicRanges)
library(Biostrings)
library(BSgenome.Dmelanogaster.UCSC.dm6)
library(TxDb.Dmelanogaster.UCSC.dm6.ensGene)
library(dplyr)
library(reshape2)
library(plyranges)
library(CAGEr)
library(magrittr)
library(ggplot2)
library(cowplot)
library(ggseqlogo)
library(gridExtra)
library(ggpubr)

# Lab sources
source("./shared_code/granges_common.r")
source("./shared_code/metapeak_common.r")
source("./shared_code/knitr_common.r")
```

Analysis

1. Define promoter types

```
## Load TSS
tss <- get(load("./rdata/dm6_mrna_ovaries_tss.RData"))

## Define the function to find promoter element (motifs) in each active tss
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
}

## Provide promoter element (motif) search information (motif sequence
## composition and search window relative to the TSS)
promoter_table <- read.table("./promoter_elements_sga.txt", header = T)
motifs <- promoter_table$name

## Find motifs across TSSs allowing 0 and 1 mismatch
motif_list_1mm <- mclapply(as.character(motifs), function(x) {
    motif <- find_motif(motif_name = x, tss$fb_t_id, 1)
    motif
}, mc.cores = 3)
```

```
motif_list_0mm <- mclapply(as.character(motifs), function(x) {
    motif <- find_motif(motif_name = x, tss$fb_t_id, 0)
    motif
}, mc.cores = 3)

motif_df_1mm <- reshape::merge_recurse(motif_list_1mm)
motif_df_0mm <- reshape::merge_recurse(motif_list_0mm)

save(motif_df_1mm, file = "./rdata/motif_df_ovaries_1mm.RData")
save(motif_df_0mm, file = "./rdata/motif_df_ovaries_0mm.RData")

tss_info <- as.data.frame(tss)[c(1:16)]

motif_info_df_0 <- merge(tss_info, motif_df_0mm)
motif_info_df_1 <- merge(tss_info, motif_df_1mm)

## Define promoter groups
tata_tss <- tss[tss$fb_t_id %in% subset(motif_df_1mm, TATA)$fb_t_id]
dpe_tss <- tss[tss$fb_t_id %in% subset(motif_df_1mm, !(TATA) & DPE_O | PB)$fb_t_id]
tct_tss <- tss[tss$fb_t_id %in% subset(motif_df_0mm, TCT)$fb_t_id]
hk_tss <- tss[tss$fb_t_id %in% subset(motif_df_0mm, !(TATA | TCT | MTE | DPE | DPE_K |
    DPE_O | PB | Inr) & (DRE | Motif1 | Motif6 | Motif7))$fb_t_id]

motif_list_ovaries <- list(tata = tata_tss, dpe = dpe_tss, tct = tct_tss, hk = hk_tss)
save(motif_list_ovaries, file = "./rdata/motif_list_ovaries.RData")
```

2. Plot a DNA-sequence heatmap of the different promoters types

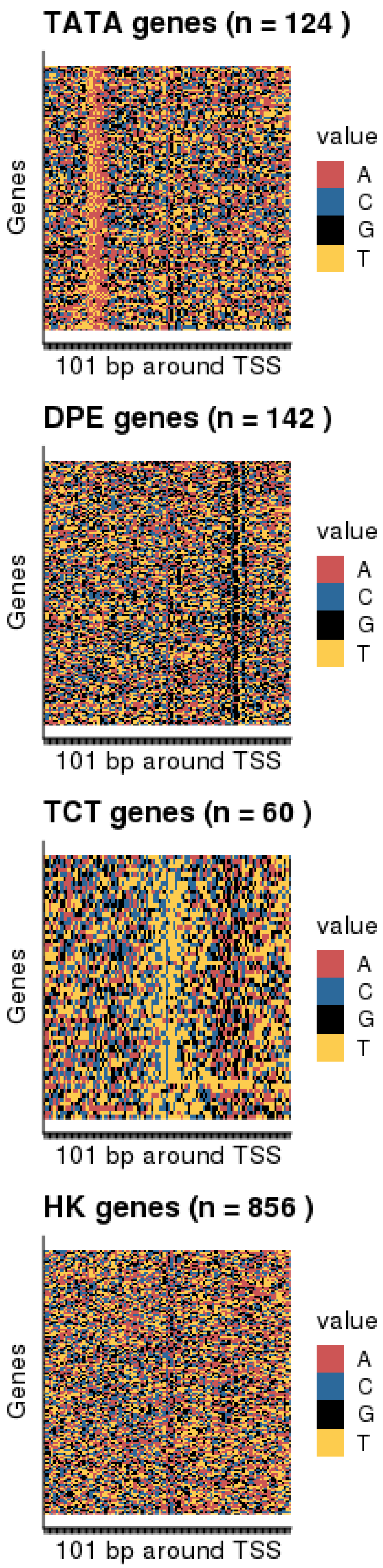
```
## Define function
get_heatmap <- function(tss, window, direction, name) {
    seq <- getSeq(Dmelanogaster, resize(tss, window, direction))
    seq_df <- as.character(seq) %>%
        lapply(., function(x) strsplit(x, "")) %>%
        unlist(., recursive = F) %>%
        do.call(rbind, .) %>%
  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

    seq_df$id <- 1:nrow(seq_df)
    seq_df_m <- reshape2::melt(seq_df, id.vars = "id")

    ATGC_plot <- ggplot(seq_df_m, aes(x = variable, y = id, fill = value)) + geom_raster() +
        scale_fill_manual(values = c("indianred3", "#2C699B", "black", "#FDCC4E")) +
        xlab(paste(window, "bp around TSS")) + ylab("Genes") + ggtitle(name) + theme_cowplot() +
        theme(axis.ticks.y = element_blank(), axis.text.y = element_blank(), axis.text.x = element_blan
}

## Generate heatmaps
tata_hm <- get_heatmap(tata_tss, 101, "center", paste("TATA genes", "(n =", length(tata_tss),
    ")"))
dpe_hm <- get_heatmap(dpe_tss, 101, "center", paste("DPE genes", "(n =", length(dpe_tss),
    ")"))
tct_hm <- get_heatmap(tct_tss, 101, "center", paste("TCT genes", "(n =", length(tct_tss),
    ")"))
hk_hm <- get_heatmap(hk_tss, 101, "center", paste("HK genes", "(n =", length(hk_tss),
    ")"))

plot_grid(tata_hm, dpe_hm, tct_hm, hk_hm, ncol = 1)
```



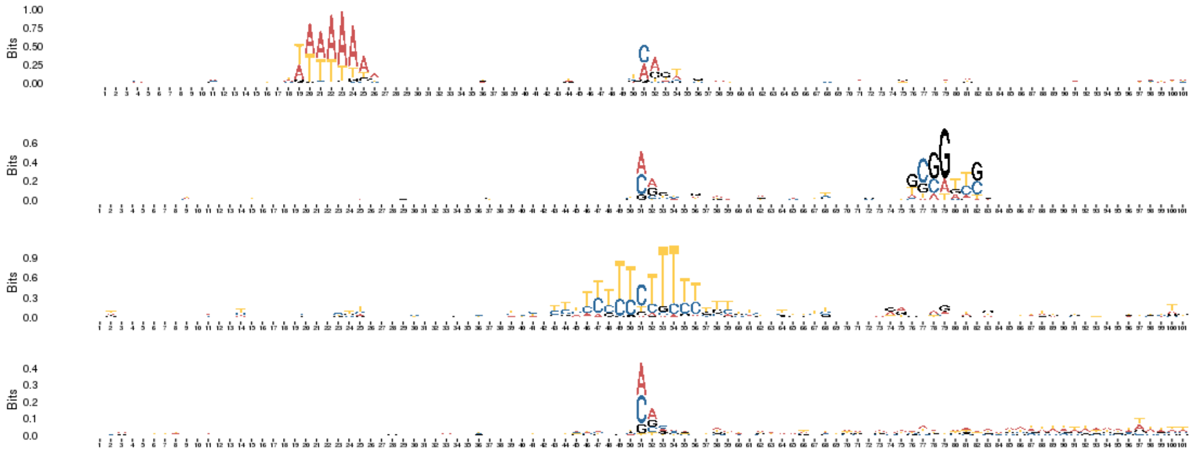
heatmap-1.png

3. Plot a position weigth matrix (PWM) across the promoter types TSSs

```
## Define function
get_logo <- function(tss) {
    cs2 = make_col_scheme(chars = c("A", "T", "C", "G"), cols = c("indianred3", "#FDCC4E",
        "#2C699B", "black"))
    seq <- as.vector(getSeq(Dmelanogaster, resize(tss, 101, "center")))
    ggseqlogo(seq, col_scheme = cs2) + theme(axis.text.x = element_text(size = 6),
        axis.ticks.x = element_line())
}

## Plot logos
tata_logo <- get_logo(tata_tss)
dpe_logo <- get_logo(dpe_tss)
tct_logo <- get_logo(tct_tss)
hk_logo <- get_logo(hk_tss)

plot_grid(tata_logo, dpe_logo, tct_logo, hk_logo, ncol = 1)
```



Session Info

```
sessionInfo()
```

```
## R version 4.1.0 (2021-05-18)
## Platform: x86_64-pc-linux-gnu (64-bit)
## Running under: CentOS Linux 7 (Core)
##
## Matrix products: default
## BLAS:   /n/apps/CentOS7/install/r-4.1.0/lib64/R/lib/libRblas.so
## LAPACK: /n/apps/CentOS7/install/r-4.1.0/lib64/R/lib/libRlapack.so
##
## 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] stats4    parallel  stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] digest_0.6.27
##  [2] pander_0.6.3
##  [3] data.table_1.14.0
##  [4] lattice_0.20-44
##  [5] ggpubr_0.4.0
##  [6] gridExtra_2.3
##  [7] ggseqlogo_0.1
##  [8] cowplot_1.1.1
##  [9] ggplot2_3.3.3
## [10] magrittr_2.0.1
## [11] CAGEr_1.34.0
## [12] MultiAssayExperiment_1.18.0
## [13] plyranges_1.12.0
## [14] reshape2_1.4.4
## [15] dplyr_1.0.6
## [16] TxDb.Dmelanogaster.UCSC.dm6.ensGene_3.12.0
## [17] GenomicFeatures_1.44.0
## [18] AnnotationDbi_1.54.0
## [19] BSgenome.Dmelanogaster.UCSC.dm6_1.4.1
## [20] BSgenome_1.60.0
## [21] rtracklayer_1.52.0
## [22] GenomicAlignments_1.28.0
## [23] Rsamtools_2.8.0
## [24] Biostrings_2.60.1
## [25] XVector_0.32.0
## [26] SummarizedExperiment_1.22.0
## [27] Biobase_2.52.0
## [28] MatrixGenerics_1.4.0
## [29] matrixStats_0.59.0
## [30] GenomicRanges_1.44.0
## [31] GenomeInfoDb_1.28.0
## [32] IRanges_2.26.0
## [33] S4Vectors_0.30.0
## [34] BiocGenerics_0.38.0
##
## loaded via a namespace (and not attached):
##   [1] VGAM_1.1-5             colorspace_2.0-1       ggsignif_0.6.1
##   [4] rjson_0.2.20           rio_0.5.26             ellipsis_0.3.2
##   [7] som_0.3-5.1            farver_2.1.0           bit64_4.0.5
##  [10] fansi_0.5.0            splines_4.1.0          cachem_1.0.5
##  [13] knitr_1.33             broom_0.7.6            cluster_2.1.2
##  [16] dbplyr_2.1.1           png_0.1-7              compiler_4.1.0
##  [19] httr_1.4.2             backports_1.2.1        assertthat_0.2.1
##  [22] Matrix_1.3-4           fastmap_1.1.0          formatR_1.11
##  [25] htmltools_0.5.1.1      prettyunits_1.1.1      tools_4.1.0
##  [28] gtable_0.3.0           glue_1.4.2             GenomeInfoDbData_1.2.6
##  [31] rappdirs_0.3.3         Rcpp_1.0.6             carData_3.0-4
##  [34] cellranger_1.1.0       vctrs_0.3.8            nlme_3.1-152
##  [37] xfun_0.23              stringr_1.4.0          openxlsx_4.2.3
##  [40] lifecycle_1.0.0        restfulr_0.0.13        formula.tools_1.7.1
##  [43] gtools_3.9.2           rstatix_0.7.0          XML_3.99-0.6
##  [46] beanplot_1.2           stringdist_0.9.6.3     zlibbioc_1.38.0
##  [49] MASS_7.3-54            scales_1.1.1           hms_1.1.0
##  [52] yaml_2.2.1             curl_4.3.1             memoise_2.0.0
##  [55] biomaRt_2.48.0         reshape_0.8.8          stringi_1.6.2
##  [58] RSQLite_2.2.7          highr_0.9              BiocIO_1.2.0
##  [61] permute_0.9-5          filelock_1.0.2         zip_2.2.0
##  [64] BiocParallel_1.26.0    operator.tools_1.6.3   rlang_0.4.11
##  [67] pkgconfig_2.0.3        bitops_1.0-7           evaluate_0.14
##  [70] purrr_0.3.4            labeling_0.4.2         bit_4.0.4
##  [73] tidyselect_1.1.1       plyr_1.8.6             R6_2.5.0
##  [76] generics_0.1.0         DelayedArray_0.18.0    DBI_1.1.1
##  [79] haven_2.4.1            foreign_0.8-81         pillar_1.6.1
##  [82] withr_2.4.2            mgcv_1.8-36            abind_1.4-5
##  [85] KEGGREST_1.32.0        RCurl_1.98-1.3         tibble_3.1.2
##  [88] crayon_1.4.1           car_3.0-10             KernSmooth_2.23-20
##  [91] utf8_1.2.1             BiocFileCache_2.0.0    rmarkdown_2.8
##  [94] progress_1.2.2         readxl_1.3.1           grid_4.1.0
##  [97] blob_1.2.1             vegan_2.5-7            forcats_0.5.1
## [100] tidyr_1.1.3            munsell_0.5.0
```