Figure S4 Reporter-ChIP-nexus recapitulates endogenous Pol II pausing

Wanqing Shao(was@stowers.org)

Contents

Description

To test whether Pol II profile on the plasmid recapitulates the pattern of endogenous promoters, we cloned promoter sequences from *Drosophila pseudoobscura* into our reporter. Pol II profile at *Act5C*, *Pino*, *pepck*, *pxb*, *comm2*, *CG12730*, *pk* and *dve* on the reporter or the endogeneous locus are plotted. Gene-specific 5' RNA sequencing was used to manually map the transcription start site.

Enviroment setup

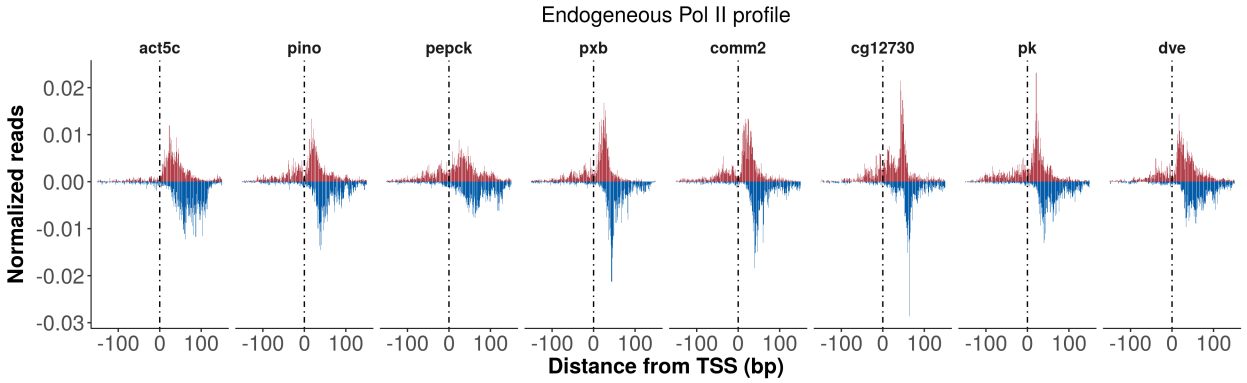
```
library(GenomicRanges, warn.conflicts=F)
library(magrittr)

setwd("/data/analysis_code")
options(knitr.figure_dir =
          "FigureS4_reporter_chipnexus_recapitulates_endogenous_polii_pausing"
        )

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

Pol II profile on the plasmid

```
plasmid_annotations <- import("./plasmid_annotation.bed")
genome_annotations <- import("./dps_genome_annotation.bed")

dps_sample_path <- load_bigwig("genome_dps_dmso_1h_rpb3_chipnexus")

get_exo_metapeak <- function(sample, upstream=100, downstream=101,
                             smooth=NA, endogeneous = F, dps_sample_path=NULL){

  gene <- as.character(subset(sample_list, sample_name == sample)$gene)
    chromosome <- as.character(subset(sample_list, sample_name == sample )$chromosome)

    if(endogeneous ==F){

        sample_path <- load_bigwig(sample)
        region <- plasmid_annotations[seqnames(plasmid_annotations) == chromosome &
                                      plasmid_annotations$name == gene] %>%
                  resize(., 1, "start")
        seqlevels(region) <- chromosome
      metapeak <- exo_metapeak(region, sample_path,
                               upstream=upstream, downstream=downstream,
                               sample_name=gene, smooth=smooth)
      metapeak$sample <- paste(metapeak$sample_name, metapeak$strand)
      metapeak

    }else{

      region <- genome_annotations[grep(gene, genome_annotations$name, ignore.case = T)]
      seqlevels(region) <- as.character(seqnames(region))
      metapeak <- exo_metapeak(region, dps_sample_path,
                               upstream=upstream, downstream=downstream,
                               sample_name=gene, smooth=smooth)
      metapeak$sample <- paste(metapeak$sample_name, metapeak$strand)
      metapeak

    }
}

genome_examples <- c("reporter_dps_act5c_rpb3_chipnexus",
                     "reporter_dps_pino_rpb3_chipnexus",
                     "reporter_dps_pepck_rpb3_chipnexus",
                     "reporter_dps_pxb_rpb3_chipnexus",
                     "reporter_dps_comm2_rpb3_chipnexus",
                     "reporter_dps_cg12730_rpb3_chipnexus",
                     "reporter_dps_pk_rpb3_chipnexus",
                     "reporter_dps_dve_rpb3_chipnexus")


genome_metapeaks <- mclapply(genome_examples, function(x){
  metapeak <- get_exo_metapeak(x, upstream=150, downstream = 151,
```

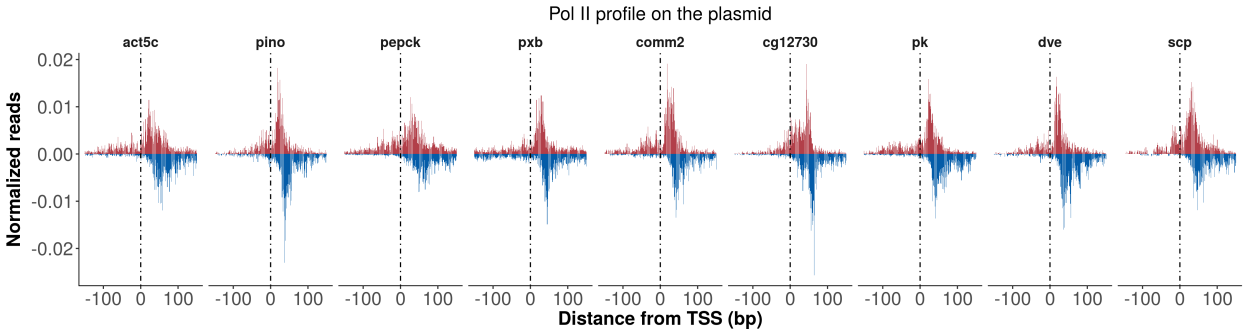
```
                                endogeneous = T, dps_sample_path= dps_sample_path)
  metapeak$reads <- metapeak$reads / sum(abs(metapeak$reads))
  metapeak
}, mc.cores =3) %>% do.call(rbind, .)

plot_exo_single_gene <- function(metapeak, name, ncol = 1, scale = "free", tss = 4){

  metapeak$sample_name <- factor(metapeak$sample_name,
                                 levels = unique(metapeak$sample_name))

  metapeak.p <- subset(metapeak, strand == "+")
  metapeak.n <- subset(metapeak, strand == "-")

  x <- ggplot(metapeak.p, aes(x=tss_distance, y=reads)) +
       geom_bar(fill="#B23F49", stat="identity") +
       geom_bar(data=metapeak.n, aes(x=tss_distance, y=reads),
                fill="#045CA8", stat="identity") +
       ggtitle(name)+
       xlab("Distance from TSS (bp)") +
       ylab("Normalized reads") +
       geom_vline(xintercept=0, linetype=tss)+
       facet_wrap(facets = "sample_name", ncol =ncol, scale = scale)

    print(x)
}

plot_exo_single_gene(genome_metapeaks, "Endogeneous Pol II profile",
                     ncol = length(genome_examples), scale = "fixed")
```



Endogeneous Pol II profile

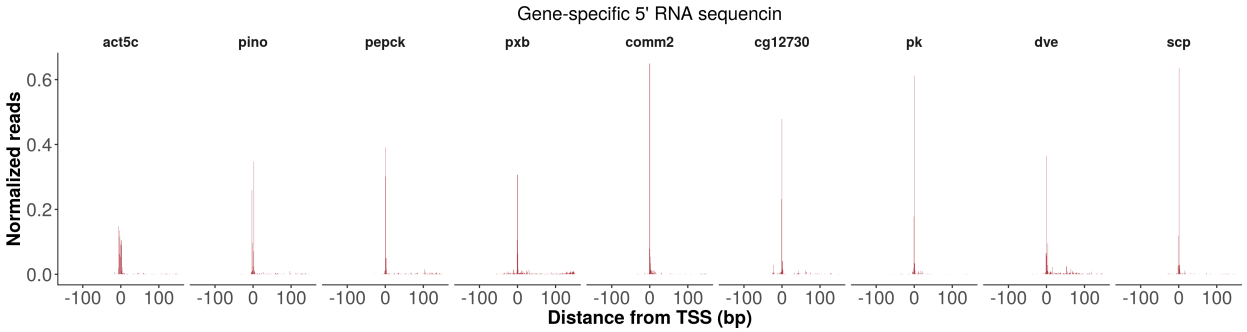
```
plasmid_examples <- c(genome_examples, "reporter_scp_rpb3_chipnexus")

plasmid_metapeaks <- mclapply(plasmid_examples, function(x){
  metapeak <- get_exo_metapeak(x, upstream=150, downstream = 151)
  metapeak$reads <- metapeak$reads / sum(abs(metapeak$reads))
  metapeak
}, mc.cores =3) %>% do.call(rbind, .)
```

```
plot_exo_single_gene(plasmid_metapeaks, "Pol II profile on the plasmid",
                     ncol = length(plasmid_examples), scale = "fixed")
```



Gene-specific 5' RNA sequencing profile

```
rna_5_samples <- c("reporter_dps_act5c_rna_5_sequencing",
                   "reporter_dps_pino_rna_5_sequencing",
                   "reporter_dps_pepck_rna_5_sequencing",
                   "reporter_dps_pxb_rna_5_sequencing",
                   "reporter_dps_comm2_rna_5_sequencing",
                   "reporter_dps_cg12730_rna_5_sequencing",
                   "reporter_dps_pk_rna_5_sequencing",
                   "reporter_dps_dve_rna_5_sequencing",
                   "reporter_scp_rna_5_sequencing")

rna_5_metapeaks <- mclapply(rna_5_samples, function(x){
  metapeak <- get_exo_metapeak(x, upstream=150, downstream = 151)
  metapeak$reads <- metapeak$reads / sum(abs(metapeak$reads))
  metapeak
}, mc.cores =3) %>% do.call(rbind, .)

plot_exo_single_gene(rna_5_metapeaks, "Gene-specific 5' RNA sequencin",
                     ncol = length(plasmid_examples), scale = "fixed",
                     tss = 0)
```



Session Info

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] parallel  stats4    stats     graphics  grDevices utils     datasets
## [8] methods   base
##
## other attached packages:
##  [1] lattice_0.20-35      reshape2_1.4.3       rtracklayer_1.38.3
##  [4] ggplot2_2.2.1        pander_0.6.1         magrittr_1.5
##  [7] GenomicRanges_1.30.3 GenomeInfoDb_1.14.0  IRanges_2.12.0
## [10] S4Vectors_0.16.0     BiocGenerics_0.24.0
##
## loaded via a namespace (and not attached):
##  [1] Rcpp_0.12.17               compiler_3.4.4
##  [3] pillar_1.2.3               plyr_1.8.4
##  [5] XVector_0.18.0             bitops_1.0-6
##  [7] tools_3.4.4                zlibbioc_1.24.0
##  [9] digest_0.6.15              evaluate_0.10.1
## [11] tibble_1.4.2               gtable_0.2.0
## [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                 Biostrings_2.46.0
## [21] rprojroot_1.3-2            grid_3.4.4
## [23] Biobase_2.38.0             XML_3.98-1.11
## [25] BiocParallel_1.12.0        rmarkdown_1.10
## [27] matrixStats_0.53.1         backports_1.1.2
## [29] scales_0.5.0               Rsamtools_1.30.0
## [31] htmltools_0.3.6            GenomicAlignments_1.14.2
## [33] SummarizedExperiment_1.8.1 colorspace_1.3-2
## [35] labeling_0.3               stringi_1.2.3
## [37] RCurl_1.95-4.10            lazyeval_0.2.1
## [39] munsell_0.5.0
```