3 Ping Pong Analysis

Bryan Teefy

08/09/2019

Ping-Pong Hits

PIWI targets are often degraded with a distinctive "ping-pong signature" consisting of a 10 bp overlap between the 5' ends of antisense-mapped piRNAs and sense-mapped piRNAs/degradome reads. Transcripts that had a ping-pong signature comprised of at least 10 of the contributing species (antisense piRNA, sense piRNA and degradome read) in the correct orientation were deemed "ping-pong hits".

To identify transcripts with "ping-pong" hits, we used the following approach:

piRNA and Degradome BAM files generated from the RSEM mapping strategy in RMD 1 were converted to BED files using the script: rsem_bam2bed.sh.

The script rsem_grouping.sh retains only those transcripts that have at least 10 piRNA/degradome reads beginning at a particular position such that the depth criteria for calling a ping-pong hit can be met. The script outputs transcript ID, start position of a piRNA/degradome read, and 10 bps from the start of the piRNA/degradome read. Adding 10 bps from the start position allows for the subsequent overlap_BT.perl script to find reads that overlap by the specified 10 bp length.
rsem_grouping.sh uses: "group_reads_sense.perl" and "group_reads_antisense.perl"

The script overlap_rsem.sh generates a matrix containing ping-pong hit coordinates.
overlap_rsem.sh makes uses: "overlap_BT.perl"

overlap_rsem.sh output files were converted to .txt files, merged in R, and used to generate TRUE/FALSE statements in reference to whether a transcript has a Whole Animal or a Epithelial Animal ping-pong hit.

The resultant table is: "rsem_ping_pong_hits.txt"

Load the Necessary Files

```
#Load normalized piRNA and degradome counts, expression data, DGE data from RMD2.

Read_Counts_Master_DF <- read.table("objects/Annotated_piRNA_Degradome_DGE_Count_Matrix.txt", sep = "\t

#Load binary matrix of ping-pong hits per transcript.

Ping_Pong_Matrix <- read.table("objects/rsem_ping_pong_hits.txt", sep = "\t")
```

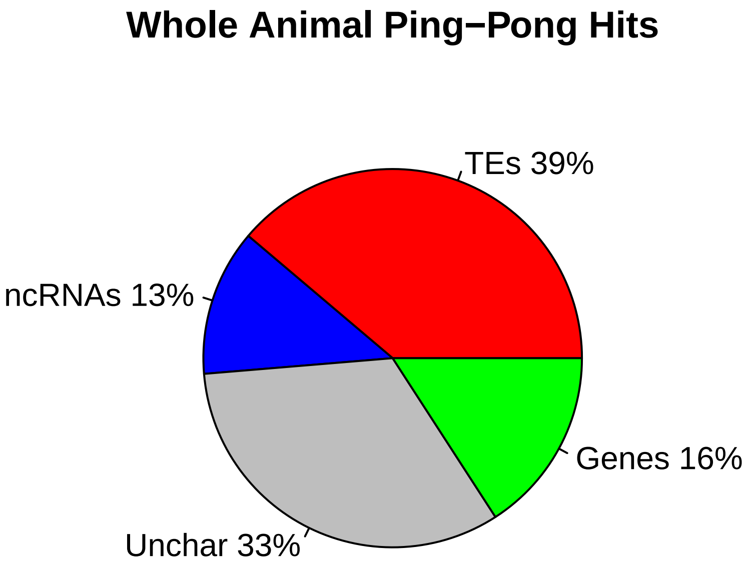
Load Necessary Packages

```
###load packages###

library(ggplot2)

library(reshape2)
```

Visualizing Ping-Pong Hits

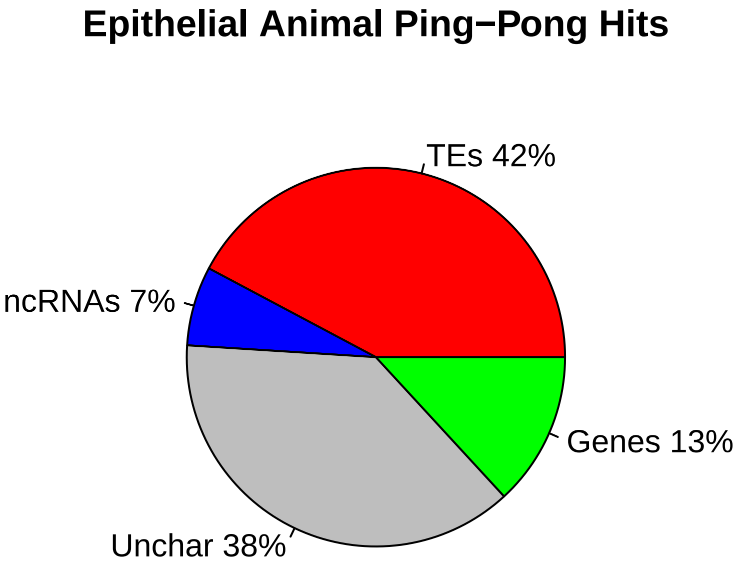
To visualize ping-pong hit distribution, we generated pie charts consisting of all transcripts that have at least one ping-pong hit and plot the output by transcript class for 1) Whole Animals and 2) Epithelial Animals.

```
Read_Counts_Master_DF <- merge(Read_Counts_Master_DF, Ping_Pong_Matrix, by = "ID")

#retrieve hits

WT_ping_hits <- c(sum(Read_Counts_Master_DF$Whole_Ping_Pong & Read_Counts_Master_DF$Transcript_Class ==

Colch_ping_hits <- c(sum(Read_Counts_Master_DF$Epi_Ping_Pong & Read_Counts_Master_DF$Transcript_Class =

#create pie chart with percentages

#Whole Animal
lbls <- c("TEs", "ncRNAs", "Unchar", "Genes")
colors = c("red", "blue", "gray", "green")
pct <- round(WT_ping_hits/sum(WT_ping_hits)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(WT_ping_hits,labels = lbls, col=colors,
    main="Whole Animal Ping-Pong Hits")
```



```
#Epithelial Animal
lbls <- c("TEs", "ncRNAs", "Unchar", "Genes")
colors = c("red", "blue", "gray", "green")
```

```
pct <- round(Colch_ping_hits/sum(Colch_ping_hits)*100)
lbls <- paste(lbls, pct) # add percents to labels
lbls <- paste(lbls,"%",sep="") # ad % to labels
pie(Colch_ping_hits,labels = lbls, col=colors,
    main="Epithelial Animal Ping-Pong Hits")
```



piRNA Positional Overlap Frequency Interrogation

Ping-pong processing of transcripts is known to occur in *Hydra* but whether this type of processing is active in somatic stem cells remains unknown. piRNAs from Whole Animals and Epithelial Animals were used to address this problem by generating piRNA overlap frequency plots. Ping-pong processing should consist of a preponderance of 10 bp overlaps between antisense- and sense-mapped piRNAs.

Positional information was extracted from piRNA BAM files and exported as a .txt file using the script: get_rsem_position.sh This script uses: "get_rep_piRNA_sense_BT.perl" and "get_rep_piRNA_antisense_BT.perl".

txt files were converted into BED files using the script: olap_bed_rsem_all.sh.

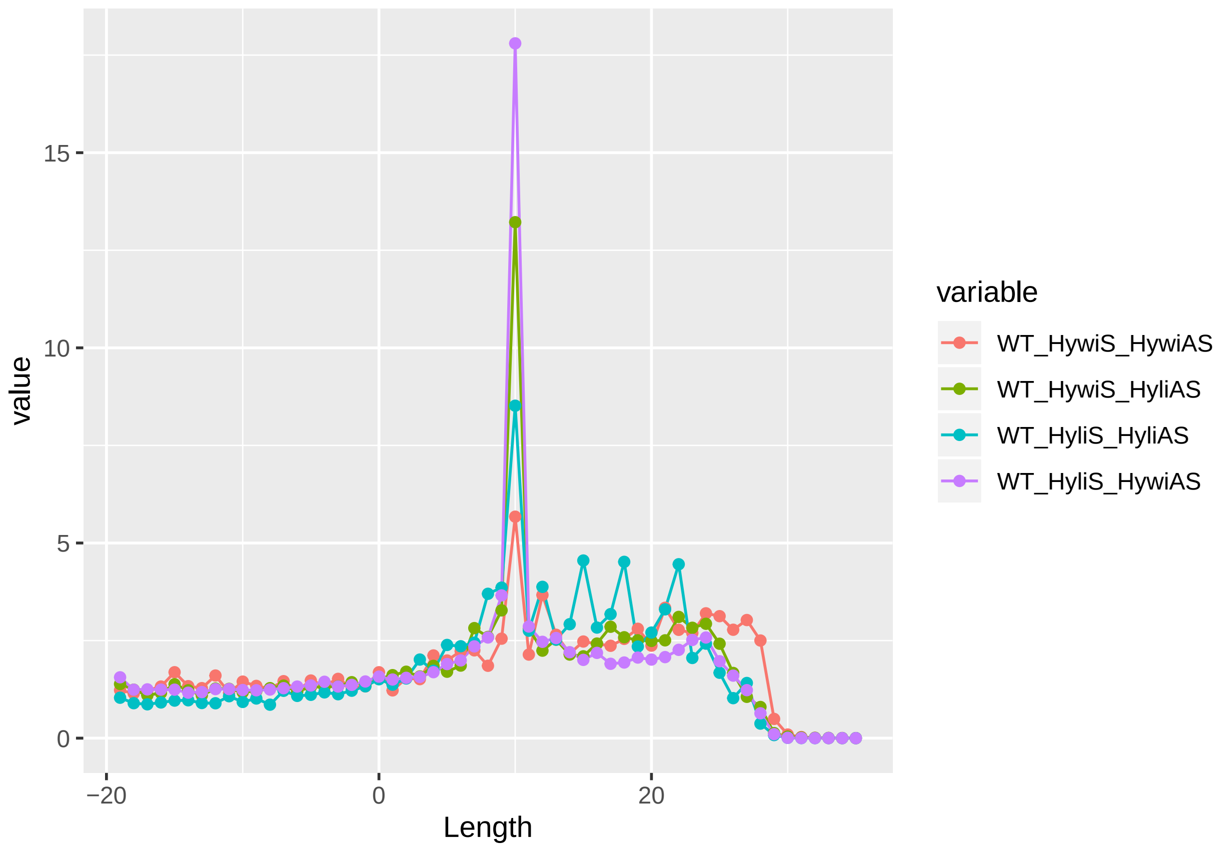
BED files containing piRNA overlap positions and overlap length between pairs of piRNAs (i.e. Hywi antisense/Hyli sense) were generated using the script windowbedall_rsem.sh.

The output of windowbedall_rsem.sh are BED files consisting of rows indexing a piRNA overlap event (ex. an overlap event between piRNA_1 and piRNA_2) within a 30 bp window.
The file contains has 11 columns:
1) Transcript on which piRNA_1 is mapped 2) Start position of piRNA_1 3) End position of piRNA_1 4) Sequence of piRNA_1 5) Copy number of piRNA_1

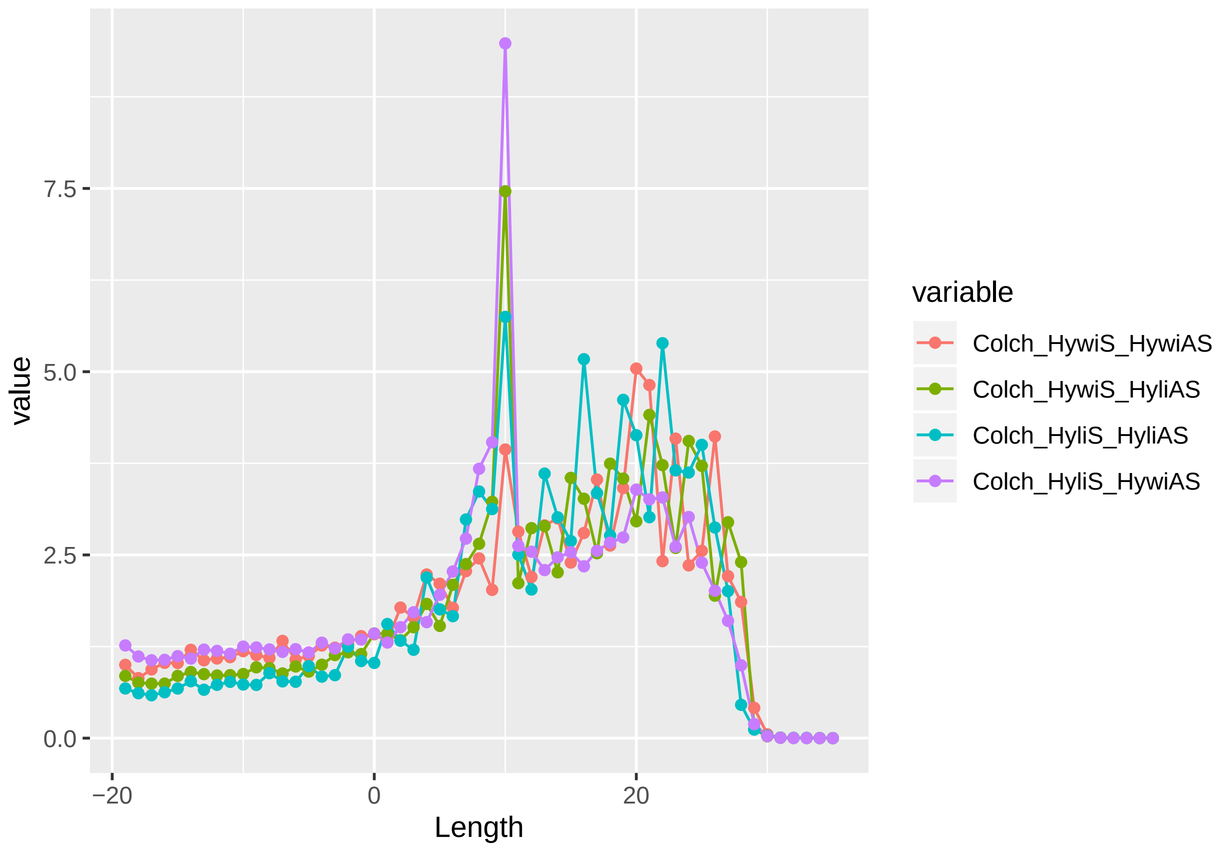
6) Transcript on which piRNA_2 is mapped 7) Start position of piRNA_2 8) End position of piRNA_2 9) Sequence of piRNA_2 10) Copy number of piRNA_2 11) piRNA overlap length

BED files are converted to txt files before importing into R.

```
#Load Matrices of piRNA overlap events

overlap_matrices <- list(read.table("objects/ping_pong/WT_HywiS_HywiAS_rsem.overlap.txt"),read.table("o

#Generate a function that provides the frequency of overlap events at a given distance between piRNA 5'

Generate_Freq <- function(x){
  x <- x[,c(5,10,11)]
  x.sub <- x[,1:2]
  x.min <- apply(x.sub, 1, min)
  x <- data.frame(frequency = x.min, Length = x[,3])
  x.table <- aggregate(frequency ~ Length, data = x, FUN = sum)
  x.table$Percentage <- (x.table$frequency/sum(x.table$frequency))*100
  x.table <- x.table[,c(1,3)]
  return(x.table)
}

Ping_Pong_List <- vector("list",length(overlap_matrices))

for (i in 1:length(overlap_matrices)){
  Ping_Pong_List[[i]] <- Generate_Freq(overlap_matrices[[i]])
  perc <- paste("Percentage_", i)
  colnames(Ping_Pong_List[[i]]) <- c("Length", perc)
}

Col_Names <- c("Length", "WT_HywiS_HywiAS","WT_HywiS_HyliAS", "WT_HyliS_HyliAS", "WT_HyliS_HywiAS", "Co

Ping_Pong_Overlap_Matrix <- Reduce(function(...) merge(..., by = "Length"), Ping_Pong_List)

colnames(Ping_Pong_Overlap_Matrix) <- paste(Col_Names, sep = "")

#Plot piRNA overlap frequency in whole animals

WT_Matrix <- Ping_Pong_Overlap_Matrix[,c(1:5)]

WTolapmelt <- melt(WT_Matrix, id.vars = "Length")

ggplot(data=WTolapmelt, aes(x=Length, y=value, group=variable, color= variable)) +
  geom_line()+
  geom_point()
```



```
#Plot piRNA overlap frequency in epithelial animals

Colch_Matrix <- Ping_Pong_Overlap_Matrix[,c(1,6:9)]

Colcholapmelt <- melt(Colch_Matrix, id.vars = "Length")

ggplot(data=Colcholapmelt, aes(x=Length, y=value, group=variable, color= variable)) +
  geom_line()+
  geom_point()
```



Software versions

This document was computed on Fri Aug 09 19:22:40 2019 with the following R package versions.

```
R version 3.5.3 (2019-03-11)
Platform: x86_64-apple-darwin15.6.0 (64-bit)
Running under: macOS Mojave 10.14.5

Matrix products: default
BLAS: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRblas.0.dylib
LAPACK: /Library/Frameworks/R.framework/Versions/3.5/Resources/lib/libRlapack.dylib

locale:
[1] en_US.UTF-8/en_US.UTF-8/en_US.UTF-8/C/en_US.UTF-8/en_US.UTF-8

attached base packages:
[1] stats     graphics  grDevices utils     datasets  methods   base

other attached packages:
[1] reshape2_1.4.3 ggplot2_3.2.0

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.1       knitr_1.22       magrittr_1.5     tidyselect_0.2.5
 [5] munsell_0.5.0    colorspace_1.4-1 R6_2.4.0         rlang_0.4.0
 [9] plyr_1.8.4        stringr_1.4.0     dplyr_0.8.3       tools_3.5.3
[13] grid_3.5.3        gtable_0.3.0      xfun_0.5          withr_2.1.2
[17] htmltools_0.3.6   yaml_2.2.0        lazyeval_0.2.2    digest_0.6.20
[21] assertthat_0.2.1 tibble_2.1.3      crayon_1.3.4      purrr_0.3.2
[25] glue_1.3.1        evaluate_0.13     rmarkdown_1.12    labeling_0.3
[29] stringi_1.4.3     compiler_3.5.3    pillar_1.4.2      scales_1.0.0
[33] pkgconfig_2.0.2
```