4 Lineage-sorted piRNA Count Generation

Bryan Teefy

08/09/2019

Load Required Libraries

```
library(dplyr)
library(reshape2)
library(ggplot2)
library(ggpubr)
library(VennDiagram)
```

Explore Lineage-sorted piRNA Diversity

To explore the diversity of piRNAs in different lineages, we identified the unique piRNAs in each lineage as well as the piRNAs species that were present in multiple lineages.

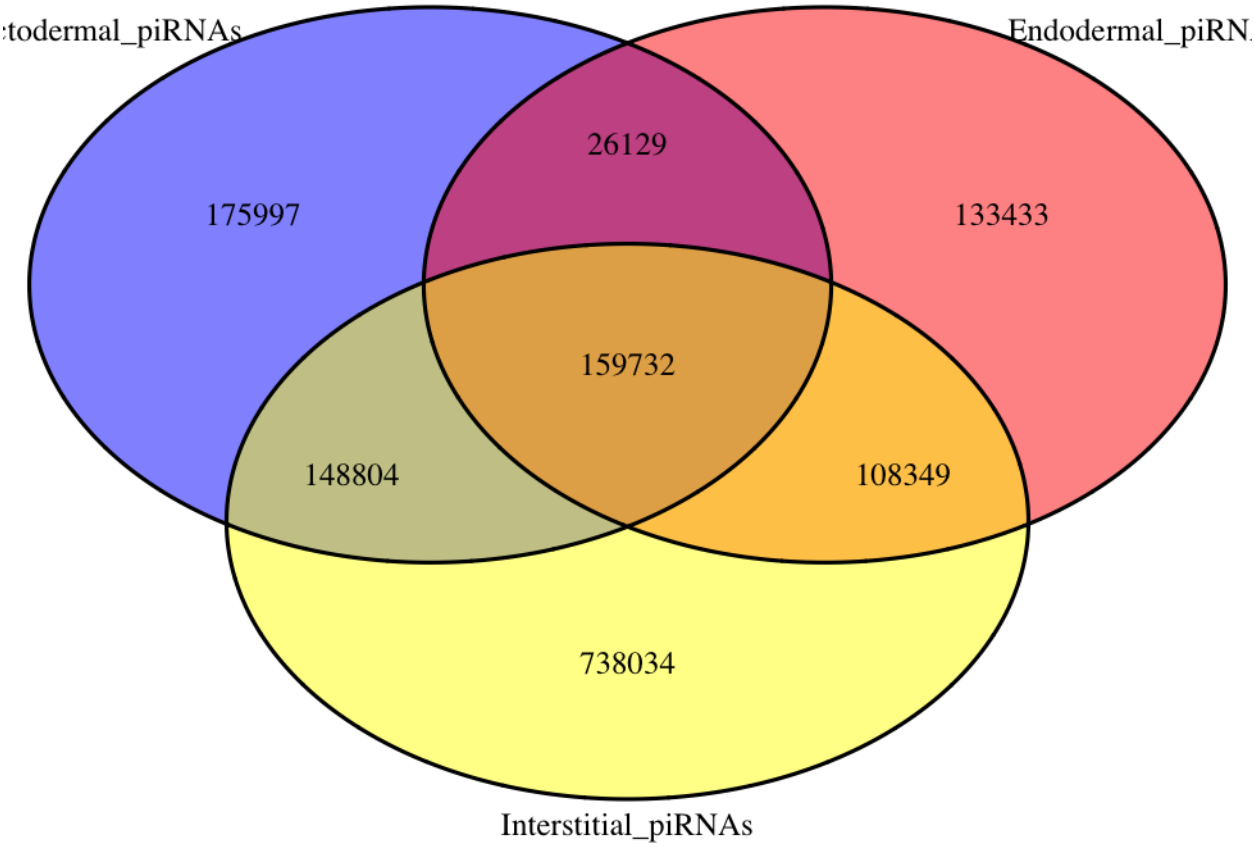
Trimmed piRNAs from Whole Animals were cross-referenced against lineage-sorted piRNA libraries (Juliano *et al.*, 2014) to retain lineage-specific piRNAs.

Lineage-specific piRNAs were saved in a R dataframe using the script, "Unique_piRNA_Generation.R" which uses the script, "run_lin_sorting.sh".

Unique and shared piRNAs were visualized using a Venn Diagram.

```
# Load unique piRNA sequences sorted by lineage and protein origin

load("objects/Unique_Ecto_Hyli_piRNAs.Rda")

load("objects/Unique_Ecto_Hywi_piRNAs.Rda")

load("objects/Unique_Endo_Hyli_piRNAs.Rda")

load("objects/Unique_Endo_Hywi_piRNAs.Rda")

load("objects/Unique_Int_Hyli_piRNAs.Rda")

load("objects/Unique_Int_Hywi_piRNAs.Rda")

# Count the number of unique piRNAs per lineage

PIWI_Ecto <- merge(EctoHyli, EctoHywi, by = "seq", all = TRUE)
rm(EctoHyli, EctoHywi)

PIWI_Endo <- merge(EndoHyli, EndoHywi, by = "seq", all = TRUE)
rm(EndoHyli, EndoHywi)

PIWI_Int <- merge(IntHyli, IntHywi, by = "seq", all = TRUE)
rm(IntHyli, IntHywi)
```

```
# Visualize shared and unique piRNA species by lineage

Venn <- list(Ectodermal_piRNAs = PIWI_Ecto$seq, Endodermal_piRNAs = PIWI_Endo$seq,
    Interstitial_piRNAs = PIWI_Int$seq)

venn.plot <- venn.diagram(Venn, filename = NULL, fill = c("blue", "red", "yellow"))
grid.draw(venn.plot)
```




```
rm(PIWI_Ecto, PIWI_Endo, PIWI_Int, Venn, venn.plot)
```

Generate Lineage-sorted piRNA Retaining Abundancies

To investigate piRNA targeting in each of the three cell lineages in *Hydra*, we cross-referenced lineage-specific small RNA libraries with the piRNA pulldown libraries from whole animals to generate lineage-specific piRNA libraries.

Trimmed piRNAs from Whole Animals were cross-referenced against lineage-sorted small RNA libraries (Juliano *et al.*, 2014) to retain only those piRNAs present in both libraries. Crucially, piRNA copy number from the piRNA pulldowns was maintained using the script, "small_RNA_piRNA_contrast.R" which uses the script, "run_contrast.sh".

The resultant lineage-sorted piRNA FASTA files were mapped to the *Hydra* transcriptome using RSEM functions rsem-calculate-expression and rsem-generate-data-matrix to generate a count matrix as in RMD 1.

Lineage-sorted piRNA mapping results are summarized in the table, "lin_rsem_piRNA_matrix.txt".

Load Transcriptome Annotation Matrix and Lineage-sorted piRNA Mapping Results

```
piRNA_Deg_counts <- read.table("objects/Annotated_piRNA_Degradome_Count_Matrix.txt",
    sep = "\t", check.names = FALSE, header = TRUE)

lin_matrix <- read.table("objects/lin_rsem_piRNA_matrix.txt", sep = "\t", header = T)
```

Generate Normalized piRNA Mapping Density Values

PIWI targets should have a high density of piRNA counts. We normalize piRNA counts by transcript length to determine piRNA count density.

To determine if the piRNA count density values were significantly different between classes of transcripts, we performed Tukey's Honest Significant Difference test to compare mean piRNA count density between each transcript type (i.e. TE, ncRNA, Unchar., Gene) for each piRNA class (i.e. Hywi Antisense-mapped, Hyli Sense-mapped, etc.).

```
# Merge lineage sorted piRNA counts with trancript length and transcript class
# data

lin_matrix <- merge(lin_matrix, piRNA_Deg_counts[, c(1, 3, 11)], by = "ID")

# To generate epithelial count values, take the mean of combined ectodermal and
# endodermal counts

lin_matrix$epi_Hyli_AS <- (lin_matrix$Ecto_Hyli_AS.isoforms.results + lin_matrix$Endo_Hyli_AS.isoforms.

lin_matrix$epi_Hyli_S <- (lin_matrix$Ecto_Hyli_S.isoforms.results + lin_matrix$Endo_Hyli_S.isoforms.resu

lin_matrix$epi_Hywi_AS <- (lin_matrix$Ecto_Hywi_AS.isoforms.results + lin_matrix$Endo_Hywi_AS.isoforms.

lin_matrix$epi_Hywi_S <- (lin_matrix$Ecto_Hywi_S.isoforms.results + lin_matrix$Endo_Hywi_S.isoforms.resu

# Calculate piRNA count density by dividing by counts by length in kilobases

norm <- (lin_matrix$Length/1000)

lin_matrix$epi_Hyli_AS_kb <- lin_matrix$epi_Hyli_AS/norm

lin_matrix$epi_Hyli_S_kb <- lin_matrix$epi_Hyli_S/norm

lin_matrix$epi_Hywi_AS_kb <- lin_matrix$epi_Hywi_AS/norm

lin_matrix$epi_Hywi_S_kb <- lin_matrix$epi_Hywi_S/norm

# Generate interstitial piRNA count density values

lin_matrix$int_Hyli_AS_kb <- lin_matrix$Int_Hyli_AS.isoforms.results/norm

lin_matrix$int_Hyli_S_kb <- lin_matrix$Int_Hyli_S.isoforms.results/norm
```

```
lin_matrix$int_Hywi_AS_kb <- lin_matrix$Int_Hywi_AS.isoforms.results/norm

lin_matrix$int_Hywi_S_kb <- lin_matrix$Int_Hywi_S.isoforms.results/norm

# Perform Tukey's Honest Significant Difference test

# Group normalized mapping counts

Normalized_Mapping_Counts_Matrix <- lin_matrix[, c(20:27, 15)]

Feeder_Plots <- melt(Normalized_Mapping_Counts_Matrix, id.var = "Transcript_Class")

# Subset count density based on piRNA origin

epi_Hyli_AS_kb_Stats <- subset(Feeder_Plots, variable == "epi_Hyli_AS_kb")
epi_Hyli_S_kb_Stats <- subset(Feeder_Plots, variable == "epi_Hyli_S_kb")
epi_Hywi_AS_kb_Stats <- subset(Feeder_Plots, variable == "epi_Hywi_AS_kb")
epi_Hywi_S_kb_Stats <- subset(Feeder_Plots, variable == "epi_Hywi_S_kb")

int_Hyli_AS_kb_Stats <- subset(Feeder_Plots, variable == "int_Hyli_AS_kb")
int_Hyli_S_kb_Stats <- subset(Feeder_Plots, variable == "int_Hyli_S_kb")
int_Hywi_AS_kb_Stats <- subset(Feeder_Plots, variable == "int_Hywi_AS_kb")
int_Hywi_S_kb_Stats <- subset(Feeder_Plots, variable == "int_Hywi_S_kb")

# Develop Tukey Test Function

Tukey_Test <- function(x) {
    res.aov <- aov(value ~ Transcript_Class, data = x)
    return(TukeyHSD(res.aov))
}

# Run Tukey Test

Tukey_Test(epi_Hyli_AS_kb_Stats)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                     diff         lwr       upr     p adj
## ncRNA-Gene    142.23248   94.595365  189.8696 0.0000000
## TE-Gene       731.49377  663.236286  799.7512 0.0000000
## Unchar-Gene   199.60642  152.900244  246.3126 0.0000000
## TE-ncRNA      589.26129  514.859206  663.6634 0.0000000
## Unchar-ncRNA   57.37394    2.074276  112.6736 0.0385121
## Unchar-TE    -531.88734 -605.696846 -458.0778 0.0000000
```

```
Tukey_Test(epi_Hyli_S_kb_Stats)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
```

```
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                      diff          lwr       upr     p adj
## ncRNA-Gene       7.227274   -6.3063246  20.76087 0.5170328
## TE-Gene        122.346413  102.9546143 141.73821 0.0000000
## Unchar-Gene     13.542257    0.2731351  26.81138 0.0433827
## TE-ncRNA       115.119140   93.9816730 136.25661 0.0000000
## Unchar-ncRNA     6.314984   -9.3955297  22.02550 0.7302595
## Unchar-TE     -108.804156 -129.7732718 -87.83504 0.0000000
```

```
Tukey_Test(epi_Hywi_AS_kb_Stats)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                     diff        lwr       upr     p adj
## ncRNA-Gene     261.78351  147.84255  375.7245 0.0000000
## TE-Gene       1145.66905  982.40721 1308.9309 0.0000000
## Unchar-Gene    330.86298  219.14868  442.5773 0.0000000
## TE-ncRNA       883.88554  705.92672 1061.8444 0.0000000
## Unchar-ncRNA    69.07947  -63.18919  201.3481 0.5362372
## Unchar-TE     -814.80607 -991.34752 -638.2646 0.0000000
```

```
Tukey_Test(epi_Hywi_S_kb_Stats)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                      diff        lwr        upr     p adj
## ncRNA-Gene      11.535023  -27.06414   50.13419 0.8690148
## TE-Gene        221.044393  165.73705  276.35173 0.0000000
## Unchar-Gene     21.036865  -16.80799   58.88172 0.4817158
## TE-ncRNA       209.509369  149.22321  269.79553 0.0000000
## Unchar-ncRNA     9.501842  -35.30610   54.30979 0.9479747
## Unchar-TE     -200.007528 -259.81353 -140.20152 0.0000000
```

```
Tukey_Test(int_Hyli_AS_kb_Stats)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                     diff         lwr       upr     p adj
## ncRNA-Gene     232.96181  154.949002  310.9746 0.0000000
## TE-Gene       1196.48078 1084.699076 1308.2625 0.0000000
## Unchar-Gene    327.20555  250.717283  403.6938 0.0000000
## TE-ncRNA       963.51897  841.674577 1085.3634 0.0000000
```

```
## Unchar-ncRNA   94.24374    3.682369  184.8051 0.0376586
## Unchar-TE    -869.27523 -990.149185 -748.4013 0.0000000
```

```
Tukey_Test(int_Hyli_S_kb_Stats)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                    diff          lwr        upr     p adj
## ncRNA-Gene     11.81402  -10.3373909   33.96543 0.5181634
## TE-Gene       199.98073  168.2407810  231.72068 0.0000000
## Unchar-Gene    22.15630    0.4377716   43.87482 0.0435122
## TE-ncRNA      188.16671  153.5695017  222.76392 0.0000000
## Unchar-ncRNA   10.34228  -15.3722483   36.05680 0.7298989
## Unchar-TE    -177.82443 -212.1460895 -143.50278 0.0000000
```

```
Tukey_Test(int_Hywi_AS_kb_Stats)
```

```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                    diff         lwr       upr     p adj
## ncRNA-Gene     348.3751   216.46844  480.2817 0.0000000
## TE-Gene       1644.3352  1455.33100 1833.3393 0.0000000
## Unchar-Gene    464.4270   335.09811  593.7558 0.0000000
## TE-ncRNA      1295.9601  1089.94162 1501.9786 0.0000000
## Unchar-ncRNA   116.0519   -37.07222  269.1761 0.2084706
## Unchar-TE    -1179.9082 -1384.28585 -975.5305 0.0000000
```

```
Tukey_Test(int_Hywi_S_kb_Stats)
```

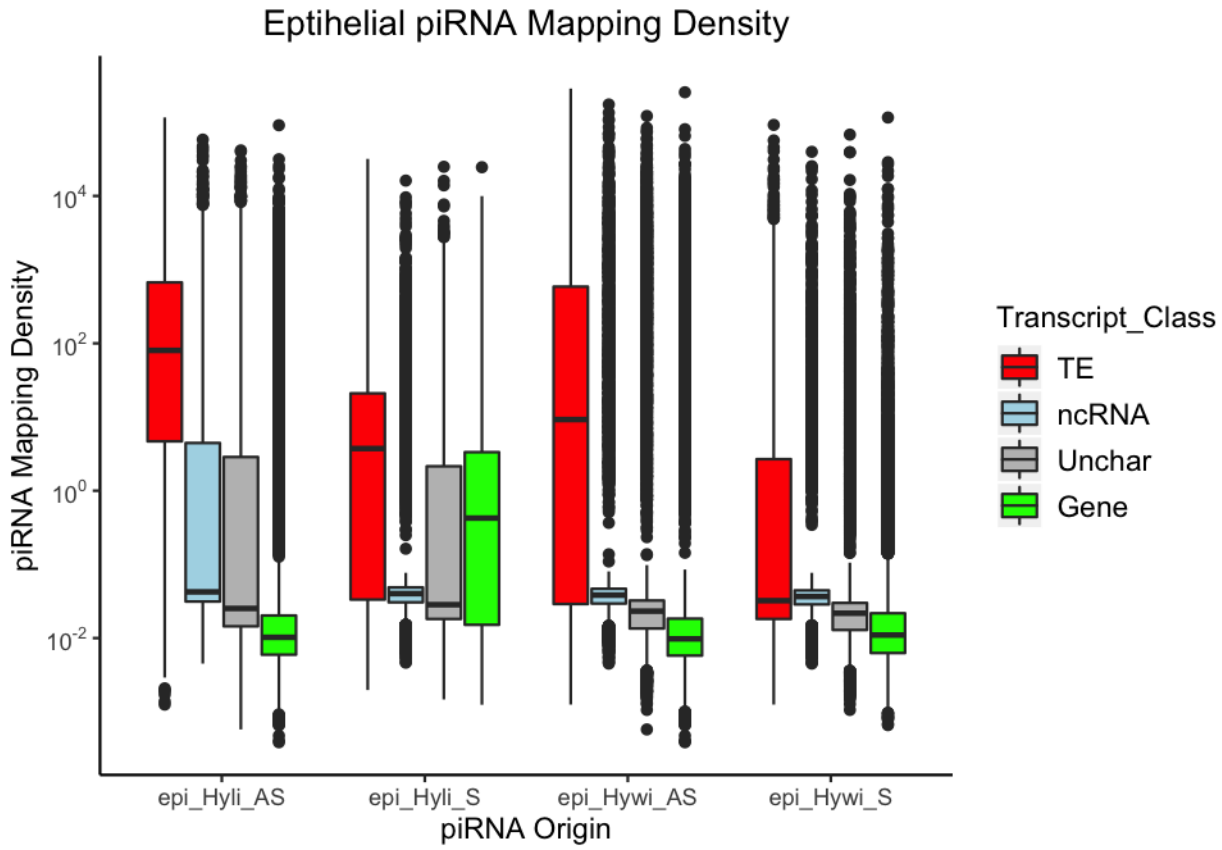
```
##   Tukey multiple comparisons of means
##     95% family-wise confidence level
##
## Fit: aov(formula = value ~ Transcript_Class, data = x)
##
## $Transcript_Class
##                    diff         lwr        upr     p adj
## ncRNA-Gene     22.49505  -19.922810   64.91291 0.5230711
## TE-Gene       297.64593  236.866932  358.42494 0.0000000
## Unchar-Gene    34.23743   -7.351492   75.82635 0.1482836
## TE-ncRNA      275.15089  208.900501  341.40127 0.0000000
## Unchar-ncRNA   11.74238  -37.498503   60.98326 0.9281041
## Unchar-TE    -263.40850 -329.131232 -197.68578 0.0000000
```

Visualizing piRNA Mapping

Since the range of observed count density values was large, we used a log scale to visualize piRNA count density. For boxplot visualization, we added a pseudocount to the raw piRNA counts to remove any 0 count

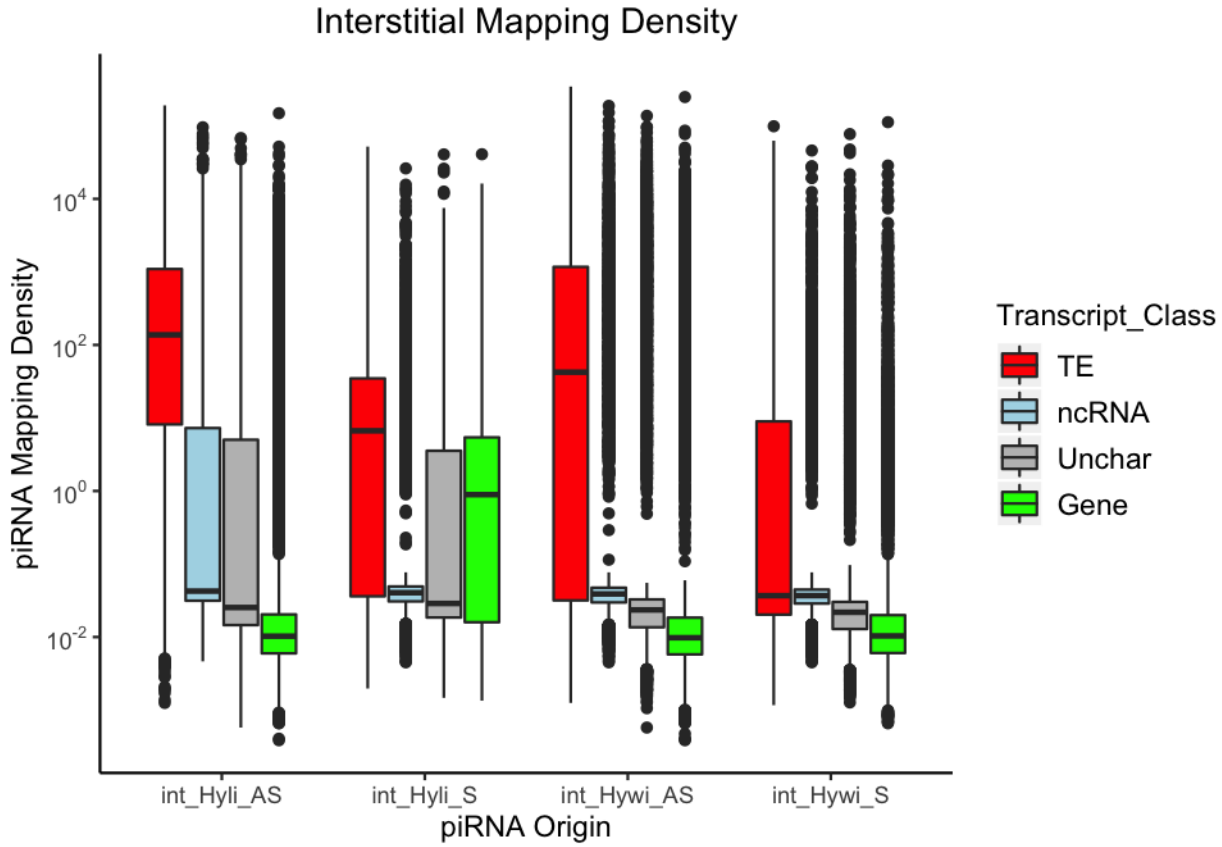
density values that would return infinite values on a log scale. The pseudocount we chose was 0.01 since that was the lowest fractional count administered by our counting strategy. We explored piRNA count density for 1) Epithelial piRNAs and 2) Interstitial piRNAs.

```
# Create pseudocount

pseudocount <- 0.01

# Add pseudocount to raw piRNA counts then generate piRNA count density values
# for epithelial piRNAs

boxplot_matrix <- lin_matrix[, c(2:15)]
boxplot_matrix[, c(1:12)] <- boxplot_matrix[, c(1:12)] + pseudocount

boxplot_matrix$epi_Hyli_AS <- (boxplot_matrix$Ecto_Hyli_AS.isoforms.results + boxplot_matrix$Endo_Hyli_A

boxplot_matrix$epi_Hyli_S <- (boxplot_matrix$Ecto_Hyli_S.isoforms.results + boxplot_matrix$Endo_Hyli_S.i

boxplot_matrix$epi_Hywi_AS <- (boxplot_matrix$Ecto_Hywi_AS.isoforms.results + boxplot_matrix$Endo_Hywi_A

boxplot_matrix$epi_Hywi_S <- (boxplot_matrix$Ecto_Hywi_S.isoforms.results + boxplot_matrix$Endo_Hywi_S.i

boxplot_matrix[, c(15:18)] <- boxplot_matrix[, c(15:18)]/(boxplot_matrix$Length/1000)

# Plot epithelial piRNA count density values

epi_matrix_feeder <- boxplot_matrix[, c(14, 15:18)]
epi_matrix_plotter <- melt(epi_matrix_feeder, id.var = "Transcript_Class")
colnames(epi_matrix_plotter) <- c("Transcript_Class", "piRNA_Origin", "piRNA_Mapping_Density")
epi_matrix_plotter$Transcript_Class <- factor(epi_matrix_plotter$Transcript_Class,
    levels = c("TE", "ncRNA", "Unchar", "Gene"))

epi_level_order <- c("epi_Hyli_AS", "epi_Hyli_S", "epi_Hywi_AS", "epi_Hywi_S")

epi_boxplot <- ggplot(data = epi_matrix_plotter, aes(x = factor(piRNA_Origin, level = epi_level_order),
    y = piRNA_Mapping_Density), log = "y") + geom_boxplot(aes(fill = Transcript_Class)) +
    scale_y_log10(breaks = scales::trans_breaks("log10", function(x) 10^x), labels = scales::trans_forma
        scales::math_format(10^.x)))

epi_boxplot + scale_fill_manual(values = c("red", "light blue", "grey", "green")) +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black")) +
    theme(legend.text = element_text(size = rel(1))) + ggtitle("Eptihelial piRNA Mapping Density") +
    theme(plot.title = element_text(hjust = 0.5)) + xlab("piRNA Origin") + ylab("piRNA Mapping Density")
```




```
# Add pseudocount to raw piRNA counts then generate piRNA count density values
# for interstitial piRNAs

boxplot_matrix$int_Hyli_AS <- (boxplot_matrix$Int_Hyli_AS.isoforms.results + boxplot_matrix$Int_Hyli_AS

boxplot_matrix$int_Hyli_S <- (boxplot_matrix$Int_Hyli_S.isoforms.results + boxplot_matrix$Int_Hyli_S.iso

boxplot_matrix$int_Hywi_AS <- (boxplot_matrix$Int_Hywi_AS.isoforms.results + boxplot_matrix$Int_Hywi_AS

boxplot_matrix$int_Hywi_S <- (boxplot_matrix$Int_Hywi_S.isoforms.results + boxplot_matrix$Int_Hywi_S.iso

boxplot_matrix[, c(19:22)] <- boxplot_matrix[, c(19:22)]/(boxplot_matrix$Length/1000)

# Plot interstitial piRNA count density values

int_matrix_feeder <- boxplot_matrix[, c(14, 19:22)]
int_matrix_plotter <- melt(int_matrix_feeder, id.var = "Transcript_Class")
colnames(int_matrix_plotter) <- c("Transcript_Class", "piRNA_Origin", "piRNA_Mapping_Density")
int_matrix_plotter$Transcript_Class <- factor(int_matrix_plotter$Transcript_Class,
    levels = c("TE", "ncRNA", "Unchar", "Gene"))

int_level_order <- c("int_Hyli_AS", "int_Hyli_S", "int_Hywi_AS", "int_Hywi_S")

int_boxplot <- ggplot(data = int_matrix_plotter, aes(x = factor(piRNA_Origin, level = int_level_order),
    y = piRNA_Mapping_Density), log = "y") + geom_boxplot(aes(fill = Transcript_Class)) +
    scale_y_log10(breaks = scales::trans_breaks("log10", function(x) 10^x), labels = scales::trans_forma
```

```
        scales::math_format(10^.x)))

int_boxplot + scale_fill_manual(values = c("red", "light blue", "grey", "green")) +
    theme(panel.grid.major = element_blank(), panel.grid.minor = element_blank(),
        panel.background = element_blank(), axis.line = element_line(colour = "black")) +
    theme(legend.text = element_text(size = rel(1))) + ggtitle("Interstitial Mapping Density") +
    theme(plot.title = element_text(hjust = 0.5)) + xlab("piRNA Origin") + ylab("piRNA Mapping Density")
```



Software versions

This document was computed on Fri Aug 09 19:23:44 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] grid      stats     graphics  grDevices utils     datasets  methods
[8] base

other attached packages:
[1] VennDiagram_1.6.20  futile.logger_1.4.3 ggpubr_0.2
[4] magrittr_1.5        ggplot2_3.2.0       reshape2_1.4.3
[7] dplyr_0.8.3         knitr_1.22

loaded via a namespace (and not attached):
 [1] Rcpp_1.0.1           munsell_0.5.0        tidyselect_0.2.5
 [4] colorspace_1.4-1     R6_2.4.0             rlang_0.4.0
 [7] stringr_1.4.0        plyr_1.8.4           tools_3.5.3
[10] gtable_0.3.0         xfun_0.5             lambda.r_1.2.3
[13] withr_2.1.2          htmltools_0.3.6      lazyeval_0.2.2
[16] yaml_2.2.0           assertthat_0.2.1     digest_0.6.20
[19] tibble_2.1.3         crayon_1.3.4         purrr_0.3.2
[22] formatR_1.7          futile.options_1.0.1 glue_1.3.1
[25] evaluate_0.13        rmarkdown_1.12       stringi_1.4.3
[28] compiler_3.5.3       pillar_1.4.2         scales_1.0.0
[31] pkgconfig_2.0.2
```