# Relationships between genes and TEs

Contents

[Relationships between genes and TEs 1](#_Toc533513336)

[Relationships between genes and TEs 1](#_Toc533513337)

[1. Generate gtf file of gene’s exons, introns, and 2kb-upstream/downstream regions 1](#_Toc533513338)

[2. Collect the intersections of genic regions and TEs 2](#_Toc533513339)

[3. Merge the intersection results 2](#_Toc533513340)

[4. Calculate TEs co-localizing with genes 4](#_Toc533513341)

# Relationships between genes and TEs

## Generate gtf file of gene’s exons, introns, and 2kb-upstream/downstream regions

#!/bin/bash

**awk** 'BEGIN{FS="\t| "}{OFS="\t"; print $12}' V2.1\_iso1\_exon.gtf **|** **uniq** **\**

**>** V2.1\_iso1\_exon\_genelist.txt

#R code

#get intron.gtf

setwd**(**"/home/thchen/project/nzgl00762/scratch/Bedtools\_intersect"**)**

gtf**<-**read.table**(**"/home/thchen/project/nzgl00762/active/TH\_Reference/curated/V2.1\_iso1\_exon.gtf", header**=**F, sep**=**"\t"**)**

gene**<-**read.table**(**"/home/thchen/project/nzgl00762/active/TH\_Reference/curated/V2.1\_iso1\_exon\_genelist.txt", header**=**F, sep**=**"\t"**)**

intron **<-** data.frame**(**chr**=** numeric**(**0**)**, source**=** numeric**(**0**)**, feature**=** numeric**(**0**)**, start**=** numeric**(**0**)**, end**=** numeric**(**0**)**, score**=** numeric**(**0**)**, strand**=** numeric**(**0**)**, frame**=**numeric**(**0**)**, id**=** numeric**(**0**))**

**for** **(**i **in** 1**:**nrow**(**gene**)){**

id**=**gene**[**i,**]**

x**<-**gtf**[**grepl**(**paste**(**id, collapse **=** "|"**)**, gtf**$**V9**)**,**]**

y**<-**nrow**(**x**)**

**if** **(**y**>**1**){**

z**=**y**-**1

**for** **(**j **in** c**(**1**:**z**)){**intron**[**nrow**(**intron**)+**1, **]** **<-** c**(**as.character**(**x**[**j,1**])**, as.character**(**x**[**j,2**])**, "intron", x**[**j,5**]+**1, x**[**j**+**1,4**]-**1, as.character**(**x**[**j,6**])**, as.character**(**x**[**j,7**])**, as.character**(**x**[**j,8**])**, as.character**(**x**[**j,9**]))}}}**

intron\_sorted **<-** intron**[**order**((**intron**$**chr**)**, as.numeric**(**as.character**(**intron**$**start**)))**, **]**

write.table**(**intron\_sorted, "VvGene\_intron.gtf", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

#get exon.gtf

write.table**(**gtf, "VvGene\_exon.gtf", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

#get flank

flank **<-** data.frame**(**chr**=** numeric**(**0**)**, source**=** numeric**(**0**)**, feature**=** numeric**(**0**)**, start**=** numeric**(**0**)**, end**=** numeric**(**0**)**, score**=** numeric**(**0**)**, strand**=** numeric**(**0**)**, frame**=**numeric**(**0**)**, id**=** numeric**(**0**))**

**for** **(**i **in** 1**:**nrow**(**gene**))** **{**

id**=**gene**[**i,**]**

x**<-**gtf**[**grepl**(**paste**(**id, collapse **=** "|"**)**, gtf**$**V9**)**,**]**

N**=**x**[**1,4**]-**2000

**if** **(**N**<**0**)** **{**N**=**1**}**

y**<-**nrow**(**x**)**

flank**[**nrow**(**flank**)+**1, **]** **<-** c**(**as.character**(**x**[**1,1**])**, as.character**(**x**[**1,2**])**, "N\_flank", N, x**[**1,4**]-**1, as.character**(**x**[**1,6**])**, as.character**(**x**[**1,7**])**, as.character**(**x**[**1,8**])**, as.character**(**x**[**1,9**]))**

flank**[**nrow**(**flank**)+**1, **]** **<-** c**(**as.character**(**x**[**y,1**])**, as.character**(**x**[**y,2**])**, "C\_flank", x**[**y,5**]+**1, x**[**y,5**]+**2000, as.character**(**x**[**y,6**])**, as.character**(**x**[**y,7**])**, as.character**(**x**[**y,8**])**, as.character**(**x**[**y,9**]))**

**}**

flank\_sorted **<-** flank**[**order**((**flank**$**chr**)**, as.numeric**(**as.character**(**flank**$**start**)))**, **]**

N\_flank **=** subset**(**flank\_sorted, feature **==** "N\_flank"**)**

C\_flank **=** subset**(**flank\_sorted, feature **==** "C\_flank"**)**

write.table**(**N\_flank, "VvGene\_Nflank.gtf", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**C\_flank, "VvGene\_Cflank.gtf", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

## Collect the intersections of genic regions and TEs

#!/bin/bash

bedtools intersect -a "VvGene\_exon.gtf" -b AllTEsExpanded\_curated\_V2.gtf **-**wa **-**wb **\**

**>** "Vv\_exon\_TE\_intersect.gtf"

bedtools intersect -a "VvGene\_intron.gtf" -b AllTEsExpanded\_curated\_V2.gtf **-**wa **-**wb **\**

**>** "Vv\_intron\_TE\_intersect.gtf"

bedtools intersect -a "VvGene\_Nflank.gtf" -b AllTEsExpanded\_curated\_V2.gtf **-**wa **-**wb **\**

**>** "Vv\_Nflank\_TE\_intersect.gtf"

bedtools intersect -a "VvGene\_Cflank.gtf" -b **/**home**/**thchen**/**project**/**nzgl00762**/**active**/**TH\_Reference**/**curated**/**AllTEsExpanded\_curated\_V2.gtf **-**wa **-**wb **\**

**>** "Vv\_Cflank\_TE\_intersect.gtf"

## Merge the intersection results

#R code

#Establish a table of genes and co-localizing TEs

#Extract information from the gtf file generated by “bedtools intersect”

#Note that Some TEs overlap with multiple genic features.

#So, TEs overlapping with genic regions would be further examined and reassigned by tagging the corresponding intersection events with "keep" or "remove"

#Role of tagging:

#If a TE has >= 5% of itself overlapping with an exon --> exonic TE

#If a TE has > 95% of itself overlapping with an intron --> intronic TE

#If a TE 1) situates within the range of a N-flank,

# 2) overlaps with a N-flank and extend outward to the 5’ end of the N-flank,

# 3) overlaps with a N-flank and the exon and has < 5% of itself overlapping with the exon

# --> N-flank TE

#If a TE 1) situates within the range of a C-flank,

# 2) overlaps with a C-flank and extend outward to the 3’ end of the C-flank,

# 3) overlaps with a C-flank and the exon and has < 5% of itself overlapping with the exon

# --> C-flank TE

#whole genome

setwd**(**"/media/ting-hsuan/ExtraDrive1/PhD/analysis/Bedtools\_intersect\_Vvgenome\_V2"**)**

#need to add " for gene\_id and transcript\_id, using text editor

df1**=**read.table**(**textConnection**(**gsub**(**" ", "\t", readLines**(**"Vv\_exon\_TE\_intersect.gtf"**))))**

df2**=**read.table**(**textConnection**(**gsub**(**" ", "\t", readLines**(**"Vv\_intron\_TE\_intersect.gtf"**))))**

df3**=**read.table**(**textConnection**(**gsub**(**" ", "\t", readLines**(**"Vv\_Nflank\_TE\_intersect.gtf"**))))**

df4**=**read.table**(**textConnection**(**gsub**(**" ", "\t", readLines**(**"Vv\_Cflank\_TE\_intersect.gtf"**))))**

df1**=**df1**[**,c**(**1,12,3,4,5,24,18,19,7,21,26**)]**

df2**=**df2**[**,c**(**1,12,3,4,5,24,18,19,7,21,26**)]**

df3**=**df3**[**,c**(**1,12,3,4,5,24,18,19,7,21,26**)]**

df4**=**df4**[**,c**(**1,12,3,4,5,24,18,19,7,21,26**)]**

df\_all**=**rbind**(**df1,df2,df3,df4**)**

colnames**(**df\_all**)=**c**(**"chr","g\_id","g\_feature","gfeature\_start","gfeature\_end","TEfamily","TE\_start","TE\_end","gene\_strand","TE\_strand", "TE\_id"**)**

#chr=chromosome; g=gene; ftr=feature; sta=start; str=strand; TE\_id=TE\_transcript\_id

#x\_bdr=cross\_boundry

cross.boundry.test**<-function(**dat**){**

y**=**nrow**(**dat**)**

dat**$**x\_bdr**=**numeric**(**y**)**

x**=**ncol**(**dat**)**

**for** **(**i **in** c**(**1**:**y**)){**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)|(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

dat**[**i,x**]=**"T"**}**

**else{**dat**[**i,x**]=**"F"**}}**

return**(**dat**)**

**}**

cross.boundry.select**<-function(**dat**){**

y**=**nrow**(**dat**)**

dat**$**select**=**numeric**(**y**)**

x**=**ncol**(**dat**)**

**for** **(**i **in** c**(**1**:**y**)){**

**if** **(**dat**[**i,3**]==**"exon"**){**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

dat**[**i,x**]=**"keep"**}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

**if** **(((**dat**[**i,5**]-**dat**[**i,7**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>=**0.05**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

**if** **(((**dat**[**i,8**]-**dat**[**i,4**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>=**0.05**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

**if** **(((**dat**[**i,5**]-**dat**[**i,4**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>=**0.05**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"keep"**}}}**

**if** **(**dat**[**i,3**]==**"intron"**){**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

dat**[**i,x**]=**"keep"**}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

**if** **(((**dat**[**i,5**]-**dat**[**i,7**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>**0.95**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

**if** **(((**dat**[**i,8**]-**dat**[**i,4**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>**0.95**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

**if** **(((**dat**[**i,5**]-**dat**[**i,4**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>**0.95**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}}**

**if** **(**dat**[**i,3**]==**"N\_flank"**){**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

dat**[**i,x**]=**"keep"**}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

**if** **(((**dat**[**i,5**]-**dat**[**i,7**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>**0.95**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

dat**[**i,x**]=**"keep"**}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

**if** **(((**dat**[**i,8**]-**dat**[**i,5**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)<**0.05**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}}**

**if** **(**dat**[**i,3**]==**"C\_flank"**){**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

dat**[**i,x**]=**"keep"**}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]>=**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

dat**[**i,x**]=**"keep"**}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]<=**0**)){**

**if** **(((**dat**[**i,8**]-**dat**[**i,4**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)>**0.95**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}**

**if** **((**dat**[**i,7**]-**dat**[**i,4**]<**0**)&(**dat**[**i,8**]-**dat**[**i,5**]>**0**)){**

**if** **(((**dat**[**i,4**]-**dat**[**i,7**]+**1**)/(**dat**[**i,8**]-**dat**[**i,7**]+**1**)<**0.05**)){**

dat**[**i,x**]=**"keep"**}**

**else{**dat**[**i,x**]=**"remove"**}}}**

**}**

return**(**dat**)**

**}**

df\_all**<-**cross.boundry.test**(**df\_all**)**

df\_all**<-**cross.boundry.select**(**df\_all**)**

#fix flank and gene strand (direction) problem

df**=**df\_all

new**=**data.frame**(**id**=**numeric**(**0**)**, fixed\_g\_ftr**=**numeric**(**0**))**

x**=**nrow**(**df**)**

**for** **(**i **in** c**(**1**:**x**)){**

**if(**df**[**i,3**]==**"exon"**){**new**[**nrow**(**new**)+**1,**]<-**c**(**as.character**(**df**[**i,2**])**, as.character**(**df**[**i,3**]))}**

**if(**df**[**i,3**]==**"intron"**){**new**[**nrow**(**new**)+**1,**]<-**c**(**as.character**(**df**[**i,2**])**, as.character**(**df**[**i,3**]))}**

**if(**df**[**i,3**]==**"N\_flank"**){**

**if(**df**[**i,9**]==**"+"**){**new**[**nrow**(**new**)+**1,**]<-**c**(**as.character**(**df**[**i,2**])**, as.character**(**df**[**i,3**]))}**

**if(**df**[**i,9**]==**"-"**){**new**[**nrow**(**new**)+**1,**]<-**c**(**as.character**(**df**[**i,2**])**, "C\_flank"**)}**

**}**

**if(**df**[**i,3**]==**"C\_flank"**){**

**if(**df**[**i,9**]==**"+"**){**new**[**nrow**(**new**)+**1,**]<-**c**(**as.character**(**df**[**i,2**])**, as.character**(**df**[**i,3**]))}**

**if(**df**[**i,9**]==**"-"**){**new**[**nrow**(**new**)+**1,**]<-**c**(**as.character**(**df**[**i,2**])**, "N\_flank"**)}**

**}}**

df**=**cbind**(**df, new**)**

df**=**df**[**,c**(**1,2,15,4**:**13**)]**

write.table**(**df, "Vv\_gene\_TE\_intersect.txt", col.names**=**T, row.names**=**F, sep**=**"\t", quote**=**F**)**

#open the file in the text editor, change the TE\_id into the right format e.g. LINE-VLINE10\_chr1\_13799-15411

#check for TEs that has been excluded in the above steps manually

setwd**(**"/media/ting-hsuan/ExtraDrive1/PhD/analysis/Bedtools\_intersect\_Vvgenome\_V2"**)**

df**=**read.table**(**"Vv\_gene\_TE\_intersect.txt", header**=**T, sep**=**"\t"**)**

df2**=**subset**(**df, df**$**select**==**"keep"**)**

before**=**df**[!**duplicated**(**df**$**TE\_id**)**,**]** #nrow=96435

after**=**df2**[!**duplicated**(**df2**$**TE\_id**)**,**]** #nrow=96071

diff.id**=!(**before**$**TE\_id %in% after**$**TE\_id**)** #nrow=364

data**=**before**[**diff.id,**]** #nrow=364, 364 TEs were lost after selection of location, 361 of them overlap with short exon that is shorter than 5% of the TE length. -> asign this TE as colocalizing with exon

write.table**(**data, "missingGenicTEs.txt", col.names**=**T, row.names**=**F, sep**=**"\t", quote**=**F**)**

data**$**geneL**=**data**$**gfeature\_end **-** data**$**gfeature\_start **+** 1

data**$**TE\_Length**=**data**$**TE\_end **-** data**$**TE\_start **+** 1

data**$**percent**=**data**$**geneL **/** data**$**TE\_Length

over5percent**=**subset**(**data, data**$**percent**>**0.05**)** #nrow=3

over5percent

#chr g\_id fixed\_g\_ftr gfeature\_start gfeature\_end TEfamily TE\_start TE\_end gene\_strand TE\_stran TE\_id x\_bdr select geneL TE\_Length percent

#chr11 VIT\_211s0149g00360 exon 7257069 7257370 Copia-67 7257314 7262032 + + Copia-67\_chr11\_7257314-7262032 TRUE remove 302 4719 0.06399661

#chr14 VIT\_214s0060g00025 exon 12840 13164 LINE-VLINE8 13022 16570 + + LINE-VLINE8\_chr14\_13022-16570 TRUE remove 325 3549 0.09157509

#chr7 VIT\_207s0005g06240 exon 11264351 11265820 Copia-30 11265798 11268441 - - Copia-30\_chr7\_11265798-11268441TRUE remove 1470 2644 0.55597579

#check these TEs on IGV:

#Copia-67\_chr11\_7257314-7262032 partially overlap with a longer exon (overlapping region < 5% of TE length) and totally cover the last exon (the exon's length (195bp) is shorter than 5% of TE length) -> asign this TE as colocalizing with exon

#LINE-VLINE8\_chr14\_13022-16570 was excluded as same resion as described above -> asign this TE as colocalizing with exon

#Copia-30 11265798 11268441 was excluded as same resion as described above -> asign this TE as colocalizing with exon

df**$**uniqueID**=**paste**(**df**[**,1**]**, df**[**,2**]**, df**[**,3**]**, df**[**,4**]**, df**[**,5**]**, df**[**,6**]**, df**[**,7**]**, df**[**,8**]**, df**[**,9**]**, df**[**,10**]**, df**[**,11**]**, df**[**,12**]**, df**[**,13**]**, sep **=** "\_"**)** #nrow=113176

df2**$**uniqueID**=**paste**(**df2**[**,1**]**, df2**[**,2**]**, df2**[**,3**]**, df2**[**,4**]**, df2**[**,5**]**, df2**[**,6**]**, df2**[**,7**]**, df2**[**,8**]**, df2**[**,9**]**, df2**[**,10**]**, df2**[**,11**]**, df2**[**,12**]**, df2**[**,13**]**, sep **=** "\_"**)** #nrow=107645

diff.id**=!(**df**$**uniqueID %in% df2**$**uniqueID**)**

df3**=**df**[**diff.id,**]** #nrow=5531

lostTE**=**data**$**TE\_id

df3.1**=**df3**[**grepl**(**paste**(**lostTE, collapse**=**"|"**)**, df3**$**TE\_id**)**,**]** #nrow=1323

df3.1**=**df3.1**[!**duplicated**(**df3.1**$**TE\_id**)**,**]** #nrow=364

diff.id**=!(**df3**$**uniqueID %in% df3.1**$**uniqueID**)**

df3.2**=**df3**[**diff.id,**]** #nrow=5167 (nrow(df3.2)5167+nrow(df3.1)364=nrow(df3)5531 )

df3.1**$**select**=**"keep"

newdf3**=**rbind**(**df3.1, df3.2**)**

newdf**=**rbind**(**newdf3, df2**)** #nrow=113176

newdf**=**newdf**[**,**-**14**]**

write.table**(**newdf, "Vv\_gene\_TE\_intersect\_V2.txt", col.names**=**T, row.names**=**F, sep**=**"\t", quote**=**F**)**

## Calculate TEs co-localizing with genes

########calculate annotated TEs co-localizing with genes

setwd**(**"/media/ting-hsuan/ExtraDrive1/PhD/analysis/Bedtools\_intersect\_Vvgenome\_V2/"**)**

df**=**read.table**(**"Vv\_gene\_TE\_intersect\_V2.txt", header**=**T, sep**=**"\t"**)**

df**=**subset**(**df, df**$**select**==**"keep"**)**

dedup**=**df**[!**duplicated**(**df**$**TE\_id**)**,**]**

TEcoGene**=**dedup**[**order**(**dedup**$**TE\_id**)**,**]**

TEcoGene**=**data.frame**(**id**=**TEcoGene**[**,11**])** #nrow=96435 (96435 TEs co-localizing with genes in the genome)

#########test position of TEs co-localizing with expr or not-expr genes

####calculate expr. candidates co-localizing with genes

setwd**(**"/media/ting-hsuan/ExtraDrive1/PhD/analysis/ECstress\_TEalignment/ExprCandidate"**)**

cand**=**read.table**(**"AllExpeCandidate\_ctrl\_tag\_new.txt", header**=**T, sep**=**"\t"**)**

flcand**=**subset**(**cand, cand**$**Lratio**>**0.9**)** #nrow=338

candcoGene**=**merge**(**cand, TEcoGene, by**=**"id"**)** #nrow=2643 (2643 candidates co-localizing with genes at T=0)

df**=**read.table**(**"NotExpeCandidate\_ctrl.bed", header**=**F, sep**=**"\t"**)**

count**=**read.table**(**"../BedCov\_OverlapBP\_sense\_count.txt", header**=**T, sep**=**"\t"**)**

data**=**count

data**=**data**[**order**(**data**$**TEm**)**,**]**

id**=**data.frame**(**id**=**df**[**,4**])**

newdata**=**merge**(**id, data, by.x**=**"id", by.y**=**"TEm"**)**

newdata**$**CountMean**=**rowMeans**(**newdata**[**,7**:**9**])**

df1**=**subset**(**newdata, newdata**$**CountMean**==**"0"**)** #nrow=195181 (195181 TEs with no expression at T=0)

noExprcoGene**=**merge**(**df1, TEcoGene, by**=**"id"**)** #nrow=78635 (78635 no-expression-TEs co-localizing with genes at T=0)

df2**=**subset**(**newdata, newdata**$**CountMean**>**0**)** #nrow=24532 (24532 non-candidates (with background expression) at T=0)

nonCandcoGene**=**merge**(**df2, TEcoGene, by**=**"id"**)** #nrow=15157 (15157 non-candidates co-localizing with genes at T=0)

setwd**(**"/media/ting-hsuan/ExtraDrive1/PhD/analysis/Bedtools\_intersect\_Vvgenome\_V2/"**)**

df**=**read.table**(**"Vv\_gene\_TE\_intersect\_V2.txt", header**=**T, sep**=**"\t"**)**

df**=**subset**(**df, df**$**select**==**"keep"**)**

data**=**merge**(**df, candcoGene, by.x**=**"TE\_id", by.y**=**"id"**)**

data**=**data**[!**duplicated**(**data**$**TE\_id**)**,**]** #nrow=2643, number of expression candidates co-localizing with gene

setwd**(**"/media/ting-hsuan/ExtraDrive1/PhD/analysis/Htseq-count/Gene\_k100Mm/GeneV2GtfReadCount"**)**

gene**=**read.table**(**"k100Mm\_Gene\_Ctrl\_exprTag.txt", header**=**T, sep**=**"\t"**)**

exprGene**=**subset**(**gene, gene**$**FPKM\_tag**==**"expr"**)**

noexprGene**=**subset**(**gene, gene**$**FPKM\_tag**==**"noexpr"**)**

TEcoExprGene**=**merge**(**data, exprGene, by.x**=**"g\_id", by.y**=**"id"**)** #nrow=2257, number of expr. candidates co-localizing with expr. gene

fl1**=**subset**(**TEcoExprGene, TEcoExprGene**$**Lratio**>**0.9**)** #nrow=256, full length candidates co-localizing with expressed genes

TEcoNoExprGene**=**merge**(**data, noexprGene, by.x**=**"g\_id", by.y**=**"id"**)** #nrow=386, number of expr. candidates co-localizing with non-expr. gene

fl2**=**subset**(**TEcoNoExprGene, TEcoNoExprGene**$**Lratio**>**0.9**)** #nrow=40

#exon, intron, flanks

exon1**=**subset**(**TEcoExprGene, TEcoExprGene**$**fixed\_g\_ftr**==**"exon"**)** #nrow=248

intron1**=**subset**(**TEcoExprGene, TEcoExprGene**$**fixed\_g\_ftr**==**"intron"**)** #nrow=1639

flank1**=**subset**(**TEcoExprGene, TEcoExprGene**$**fixed\_g\_ftr**==**"N\_flank" **|** TEcoExprGene**$**fixed\_g\_ftr**==**"C\_flank"**)** #nrow=370

Nflank1**=**subset**(**TEcoExprGene, TEcoExprGene**$**fixed\_g\_ftr**==**"N\_flank"**)** #nrow=160

Cflank1**=**subset**(**TEcoExprGene, TEcoExprGene**$**fixed\_g\_ftr**==**"C\_flank"**)** #nrow=210

#check duplicated TEs

dup**=**exon1**[**duplicated**(**exon1**$**TE\_id**)**,**]** #nrow=0

dup**=**intron1**[**duplicated**(**intron1**$**TE\_id**)**,**]** #nrow=0

Ndup**=**Nflank1**[**duplicated**(**Nflank**$**TE\_id**)**,**]** #nrow=0

Cdup**=**Cflank1**[**duplicated**(**Cflank**$**TE\_id**)**,**]** #nrow=0

dup**=**flank1**[**duplicated**(**flank1**$**TE\_id**)**,**]** #nrow=0

setwd**(**"/media/ting-hsuan/ExtraDrive1/PhD/analysis/ECstress\_TEalignment/ExprCandidate"**)**

sink**(**"t-test\_candidatesCoExprGene\_ins\_length\_C.txt", append**=FALSE**, split**=FALSE)**

t.test**(**exon1**$**length,intron1**$**length**)**

t.test**(**exon1**$**length,Nflank1**$**length**)**

t.test**(**exon1**$**length,Cflank1**$**length**)**

t.test**(**intron1**$**length,Nflank1**$**length**)**

t.test**(**intron1**$**length,Cflank1**$**length**)**

t.test**(**Nflank1**$**length,Cflank1**$**length**)**

sink**(**file **=** **NULL)**

#boxplot

exon1**$**newftr**=**"Exon"

intron1**$**newftr**=**"Intron"

Nflank1**$**newftr**=**"N-Flank"

Cflank1**$**newftr**=**"C-Flank"

dfnew**=**rbind**(**exon1,intron1,Nflank1,Cflank1**)**

dfnew**$**newftr**<-**factor**(**dfnew**$**newftr, levels**=**c**(**"Exon","Intron","N-Flank","C-Flank"**)**, labels**=**c**(**"Exon","Intron","N-Flank","C-Flank"**))**

library**(**ggplot2**)**

ggplot**(**dfnew, aes**(**x**=**newftr, y**=**length**))+**

theme**(**panel.background **=** element\_rect**(**fill **=** "white", colour **=** "grey20"**))+**

geom\_boxplot**(**aes**(**fill **=** newftr, color**=** newftr**)**, outlier.size **=** 6, outlier.colour **=** **NULL**, size**=**2**)+**

stat\_summary**(**geom **=** "crossbar", width**=**0.76, fatten**=**3, color**=**"white", fun.data **=** **function(**x**){** return**(**c**(**y**=**median**(**x**)**, ymin**=**median**(**x**)**, ymax**=**median**(**x**)))** **})+**

ylim**(**0,16000**)+**

labs**(**title**=**"Candidate length vs. position ", x**=**"position", y**=**"TE length (bp)"**)+**

theme**(**axis.title**=**element\_text**(**size**=**40**)**, title**=**element\_text**(**size**=**40**)**, axis.text**=**element\_text**(**size**=**36**)**, legend.title**=**element\_blank**()**,legend.text**=**element\_text**(**size**=**36**))**

dev.copy**(**png,"candidatesCoExprGene\_ins\_length\_C.png", width **=** 1500, height **=** 1800**)**

dev.off**()**

exon2**=**subset**(**TEcoNoExprGene, TEcoNoExprGene**$**fixed\_g\_ftr**==**"exon"**)** #nrow=50

intron2**=**subset**(**TEcoNoExprGene, TEcoNoExprGene**$**fixed\_g\_ftr**==**"intron"**)** #nrow=84

flank2**=**subset**(**TEcoNoExprGene, TEcoNoExprGene**$**fixed\_g\_ftr**==**"N\_flank" **|** TEcoNoExprGene**$**fixed\_g\_ftr**==**"C\_flank"**)** #nrow=252

Nflank2**=**subset**(**TEcoNoExprGene, TEcoNoExprGene**$**fixed\_g\_ftr**==**"N\_flank"**)** #nrow=118

Cflank2**=**subset**(**TEcoNoExprGene, TEcoNoExprGene**$**fixed\_g\_ftr**==**"C\_flank"**)** #nrow=134

#check duplicated TEs

dup**=**exon2**[**duplicated**(**exon2**$**TE\_id**)**,**]** #nrow=0

dup**=**intron2**[**duplicated**(**intron2**$**TE\_id**)**,**]** #nrow=0

Ndup**=**Nflank2**[**duplicated**(**Nflank2**$**TE\_id**)**,**]** #nrow=0

Cdup**=**Cflank2**[**duplicated**(**Cflank2**$**TE\_id**)**,**]** #nrow=0

dup**=**flank2**[**duplicated**(**flank2**$**TE\_id**)**,**]** #nrow=0

sink**(**"t-test\_candidatesCoNoExprGene\_ins\_length\_C.txt", append**=FALSE**, split**=FALSE)**

t.test**(**exon2**$**length,intron2**$**length**)**

t.test**(**exon2**$**length,Nflank2**$**length**)**

t.test**(**exon2**$**length,Cflank2**$**length**)**

t.test**(**intron2**$**length,Nflank2**$**length**)**

t.test**(**intron2**$**length,Cflank2**$**length**)**

t.test**(**Nflank2**$**length,Cflank2**$**length**)**

sink**(**file **=** **NULL)**

#boxplot

exon2**$**newftr**=**"Exon"

intron2**$**newftr**=**"Intron"

Nflank2**$**newftr**=**"N-Flank"

Cflank2**$**newftr**=**"C-Flank"

dfnew**=**rbind**(**exon2,intron2,Nflank2,Cflank2**)**

dfnew**$**newftr**<-**factor**(**dfnew**$**newftr, levels**=**c**(**"Exon","Intron","N-Flank","C-Flank"**)**, labels**=**c**(**"Exon","Intron","N-Flank","C-Flank"**))**

library**(**ggplot2**)**

ggplot**(**dfnew, aes**(**x**=**newftr, y**=**length**))+**

theme**(**panel.background **=** element\_rect**(**fill **=** "white", colour **=** "grey20"**))+**

geom\_boxplot**(**aes**(**fill **=** newftr, color**=** newftr**)**, outlier.size **=** 6, outlier.colour **=** **NULL**, size**=**2**)+**

stat\_summary**(**geom **=** "crossbar", width**=**0.76, fatten**=**3, color**=**"white", fun.data **=** **function(**x**){** return**(**c**(**y**=**median**(**x**)**, ymin**=**median**(**x**)**, ymax**=**median**(**x**)))** **})+**

ylim**(**0,16000**)+**

labs**(**title**=**"Candidate length vs. position ", x**=**"position", y**=**"TE length (bp)"**)+**

theme**(**axis.title**=**element\_text**(**size**=**40**)**, title**=**element\_text**(**size**=**40**)**, axis.text**=**element\_text**(**size**=**36**)**, legend.title**=**element\_blank**()**,legend.text**=**element\_text**(**size**=**36**))**

dev.copy**(**png,"candidatesCoNoExprGene\_ins\_length\_C.png", width **=** 1500, height **=** 1800**)**

dev.off**()**

sink**(**"t-test\_candidatesCoExpr\_Vs\_NoExprGene\_ins\_length\_C.txt", append**=FALSE**, split**=FALSE)**

t.test**(**exon1**$**length,exon2**$**length**)**

t.test**(**intron1**$**length,intron2**$**length**)**

t.test**(**Nflank1**$**length,Nflank2**$**length**)**

t.test**(**Cflank1**$**length,Cflank2**$**length**)**

sink**(**file **=** **NULL)**

#boxplot

exon1**$**geneExpr**=**"ExprGene"

exon2**$**geneExpr**=**"NonExprGene"

intron1**$**geneExpr**=**"ExprGene"

intron2**$**geneExpr**=**"NonExprGene"

Nflank1**$**geneExpr**=**"ExprGene"

Nflank2**$**geneExpr**=**"NonExprGene"

Cflank1**$**geneExpr**=**"ExprGene"

Cflank2**$**geneExpr**=**"NonExprGene"

dfnew**=**rbind**(**exon1,exon2**)**

dfnew**$**geneExpr**<-**factor**(**dfnew**$**geneExpr, levels**=**c**(**"ExprGene","NonExprGene"**)**, labels**=**c**(**"ExprGene","NonExprGene"**))**

library**(**ggplot2**)**

ggplot**(**dfnew, aes**(**geneExpr, length**))+**

theme**(**panel.background **=** element\_rect**(**fill **=** "white", colour **=** "grey20"**))+**

geom\_boxplot**(**aes**(**fill **=** geneExpr, color**=** geneExpr**)**, outlier.size **=** 6, outlier.colour **=** **NULL**, size**=**2**)+**

stat\_summary**(**geom **=** "crossbar", width**=**0.76, fatten**=**3, color**=**"white", fun.data **=** **function(**x**){** return**(**c**(**y**=**median**(**x**)**, ymin**=**median**(**x**)**, ymax**=**median**(**x**)))** **})+**

ylim**(**0,16000**)+**

labs**(**title**=**"Candidate length vs. gene expression \n (exon insertion)", y**=**"TE length (bp)"**)+**

theme**(**axis.title**=**element\_text**(**size**=**40**)**, title**=**element\_text**(**size**=**40**)**, axis.text**=**element\_text**(**size**=**36**)**, legend.title**=**element\_blank**()**,legend.text**=**element\_text**(**size**=**36**))**

dev.copy**(**png,"candidatesCoExpr\_Vs\_NoExprGene\_exonIns\_length\_C.png", width **=** 1500, height **=** 1800**)**

dev.off**()**

dfnew**=**rbind**(**intron1,intron2**)**

dfnew**$**geneExpr**<-**factor**(**dfnew**$**geneExpr, levels**=**c**(**"ExprGene","NonExprGene"**)**, labels**=**c**(**"ExprGene","NonExprGene"**))**

library**(**ggplot2**)**

ggplot**(**dfnew, aes**(**geneExpr, length**))+**

theme**(**panel.background **=** element\_rect**(**fill **=** "white", colour **=** "grey20"**))+**

geom\_boxplot**(**aes**(**fill **=** geneExpr, color**=** geneExpr**)**, outlier.size **=** 6, outlier.colour **=** **NULL**, size**=**2**)+**

stat\_summary**(**geom **=** "crossbar", width**=**0.76, fatten**=**3, color**=**"white", fun.data **=** **function(**x**){** return**(**c**(**y**=**median**(**x**)**, ymin**=**median**(**x**)**, ymax**=**median**(**x**)))** **})+**

ylim**(**0,16000**)+**

labs**(**title**=**"Candidate length vs. gene expression \n (intron insertion)", y**=**"TE length (bp)"**)+**

theme**(**axis.title**=**element\_text**(**size**=**40**)**, title**=**element\_text**(**size**=**40**)**, axis.text**=**element\_text**(**size**=**36**)**, legend.title**=**element\_blank**()**,legend.text**=**element\_text**(**size**=**36**))**

dev.copy**(**png,"candidatesCoExpr\_Vs\_NoExprGene\_intronIns\_length\_C.png", width **=** 1500, height **=** 1800**)**

dev.off**()**

dfnew**=**rbind**(**Nflank1,Nflank2**)**

dfnew**$**geneExpr**<-**factor**(**dfnew**$**geneExpr, levels**=**c**(**"ExprGene","NonExprGene"**)**, labels**=**c**(**"ExprGene","NonExprGene"**))**

library**(**ggplot2**)**

ggplot**(**dfnew, aes**(**geneExpr, length**))+**

theme**(**panel.background **=** element\_rect**(**fill **=** "white", colour **=** "grey20"**))+**

geom\_boxplot**(**aes**(**fill **=** geneExpr, color**=** geneExpr**)**, outlier.size **=** 6, outlier.colour **=** **NULL**, size**=**2**)+**

stat\_summary**(**geom **=** "crossbar", width**=**0.76, fatten**=**3, color**=**"white", fun.data **=** **function(**x**){** return**(**c**(**y**=**median**(**x**)**, ymin**=**median**(**x**)**, ymax**=**median**(**x**)))** **})+**

ylim**(**0,16000**)+**

labs**(**title**=**"Candidate length vs. gene expression \n (Nflank insertion)", y**=**"TE length (bp)"**)+**

theme**(**axis.title**=**element\_text**(**size**=**40**)**, title**=**element\_text**(**size**=**40**)**, axis.text**=**element\_text**(**size**=**36**)**, legend.title**=**element\_blank**()**,legend.text**=**element\_text**(**size**=**36**))**

dev.copy**(**png,"candidatesCoExpr\_Vs\_NoExprGene\_NflankIns\_length\_C.png", width **=** 1500, height **=** 1800**)**

dev.off**()**

dfnew**=**rbind**(**Cflank1,Cflank2**)**

dfnew**$**geneExpr**<-**factor**(**dfnew**$**geneExpr, levels**=**c**(**"ExprGene","NonExprGene"**)**, labels**=**c**(**"ExprGene","NonExprGene"**))**

library**(**ggplot2**)**

ggplot**(**dfnew, aes**(**geneExpr, length**))+**

theme**(**panel.background **=** element\_rect**(**fill **=** "white", colour **=** "grey20"**))+**

geom\_boxplot**(**aes**(**fill **=** geneExpr, color**=** geneExpr**)**, outlier.size **=** 6, outlier.colour **=** **NULL**, size**=**2**)+**

stat\_summary**(**geom **=** "crossbar", width**=**0.76, fatten**=**3, color**=**"white", fun.data **=** **function(**x**){** return**(**c**(**y**=**median**(**x**)**, ymin**=**median**(**x**)**, ymax**=**median**(**x**)))** **})+**

ylim**(**0,16000**)+**

labs**(**title**=**"Candidate length vs. gene expression \n (Cflank insertion)", y**=**"TE length (bp)"**)+**

theme**(**axis.title**=**element\_text**(**size**=**40**)**, title**=**element\_text**(**size**=**40**)**, axis.text**=**element\_text**(**size**=**36**)**, legend.title**=**element\_blank**()**,legend.text**=**element\_text**(**size**=**36**))**

dev.copy**(**png,"candidatesCoExpr\_Vs\_NoExprGene\_CflankIns\_length\_C.png", width **=** 1500, height **=** 1800**)**

dev.off**()**