# Examination of the pipeline results

Contents

[Examination of the pipeline results 1](#_Toc532986907)

[Examination of the pipeline results 2](#_Toc532986908)

[1. Proportion of expressed TE candidates among annotated TEs 2](#_Toc532986909)

[2. Relations among the TE candidates captured by different sub-pipelines 2](#_Toc532986910)

[3. Comparison of the expression range of under-threshold TEs and expr. candidates of T=0 captured by different methods 3](#_Toc532986911)

[4. Calculation of total mapped reads 4](#_Toc532986912)

[5. Calculation of reads mapping to genes 4](#_Toc532986913)

[6. Calculation of reads mapping to TEs 5](#_Toc532986914)

[7. Calculation of reads mapping to TEs in sense orientation 5](#_Toc532986915)

[8. Calculation of reads mapping to TEs in antisense orientation 5](#_Toc532986916)

[9. Calculation of reads mapping to a TE in sense orientation but in antisense to another 6](#_Toc532986917)

[10. Calculation of reads mapping to the expressed TE candidates and Non-candidates 7](#_Toc532986918)

# Examination of the pipeline results

## Proportion of expressed TE candidates among annotated TEs

#R code

#ctrl

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

count1**=**count**[**,1**:**9**]**

count1**$**countMean**=**rowMeans**(**count1**[**,7**:**9**])**

df**=**count1

dfZero**=**subset**(**df, df**$**countMean**==**0**)** #nrow=195181; TEs with zero expression = 195181 TEs

dfNonZero**=**subset**(**df, df**$**countMean**>**0**)** #nrow=28230

data**=**read.table**(**"AllExpeCandidate\_ctrl\_tag\_new.txt", header**=**T, sep**=**"\t"**)** #nrow=3698; expr candidates

data**<-**data**[**order**(**data**$**id**)**,**]**

dfNonZero**<-**dfNonZero**[**order**(**dfNonZero**$**TEm**)**,**]**

dfNonZero**=**merge**(**dfNonZero, data, by.x**=**"TEm", by.y**=**"id", all.x**=TRUE)**

underthreshold**=**subset**(**dfNonZero, is.na**(**dfNonZero**$**C\_tracking**))** #nrow=24532; TEs under threshold

#mock

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

count1**=**count**[**,c**(**1**:**6, 10**:**21**)]**

count1**$**countMean**=**rowMeans**(**count1**[**,7**:**18**])**

df**=**count1

dfZero**=**subset**(**df, df**$**countMean**==**0**)** #nrow=179568; TEs with zero expression

dfNonZero**=**subset**(**df, df**$**countMean**>**0**)** #nrow=43843

data**=**read.table**(**"AllExpeCandidate\_mock\_tag\_new.txt", header**=**T, sep**=**"\t"**)** #nrow=5524; expr candidates

data**<-**data**[**order**(**data**$**id**)**,**]**

dfNonZero**<-**dfNonZero**[**order**(**dfNonZero**$**TEm**)**,**]**

dfNonZero**=**merge**(**dfNonZero, data, by.x**=**"TEm", by.y**=**"id", all.x**=TRUE)**

underthreshold**=**subset**(**dfNonZero, is.na**(**dfNonZero**$**C\_tracking**))** #nrow=38319; TEs under threshold

#yeast

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

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

count1**=**count**[**,c**(**1**:**6, 22**:**33**)]**

count1**$**countMean**=**rowMeans**(**count1**[**,7**:**18**])**

df**=**count1

dfZero**=**subset**(**df, df**$**countMean**==**0**)** #nrow= 168340; TEs with zero expression

dfNonZero**=**subset**(**df, df**$**countMean**>**0**)** #nrow= 55071

data**=**read.table**(**"AllExpeCandidate\_yeast\_tag\_new.txt", header**=**T, sep**=**"\t"**)** #nrow=5531; expr candidates

data**<-**data**[**order**(**data**$**id**)**,**]**

dfNonZero**<-**dfNonZero**[**order**(**dfNonZero**$**TEm**)**,**]**

dfNonZero**=**merge**(**dfNonZero, data, by.x**=**"TEm", by.y**=**"id", all.x**=TRUE)**

underthreshold**=**subset**(**dfNonZero, is.na**(**dfNonZero**$**C\_tracking**))** #nrow=49540; TEs under threshold

#botrytis

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

count1**=**count**[**,c**(**1**:**6, 34**:**45**)]**

count1**$**countMean**=**rowMeans**(**count1**[**,7**:**18**])**

df**=**count1

dfZero**=**subset**(**df, df**$**countMean**==**0**)** #nrow= 168829; TEs with zero expression

dfNonZero**=**subset**(**df, df**$**countMean**>**0**)** #nrow= 54582

data**=**read.table**(**"AllExpeCandidate\_botrytis\_tag\_new.txt", header**=**T, sep**=**"\t"**)** #nrow=5171; expr candidates

data**<-**data**[**order**(**data**$**id**)**,**]**

dfNonZero**<-**dfNonZero**[**order**(**dfNonZero**$**TEm**)**,**]**

dfNonZero**=**merge**(**dfNonZero, data, by.x**=**"TEm", by.y**=**"id", all.x**=TRUE)**

underthreshold**=**subset**(**dfNonZero, is.na**(**dfNonZero**$**C\_tracking**))** #nrow=49411; TEs under threshold

## Relations among the TE candidates captured by different sub-pipelines

#R code

#ctrl

RE**=**read.table**(**"k100Mm\_AllSenseTEm\_curatedGTF\_exprCtrl05\_V4.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 1

df**=**read.table**(**"BedCov\_OverlapBP\_PassC.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 2

dangler**=**read.table**(**"AllRep\_comp\_intersect\_exprCtrl\_V4.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 3

RE**=**RE**[!**grepl**(**"\_\_", RE**$**id**)**,**]**

RE**=**RE**[**order**(**RE**$**id**)**,**]**

dangler**=**dangler**[**order**(**dangler**$**id**)**,**]**

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

v**<-**venn.diagram**(**list**(**"count&depth"**=**df**[**,1**]**, "htseq-count"**=**RE**[**,1**]**,

"TEFingerprint"**=**dangler**[**,1**])**, filename**=NULL)**

grid.newpage**()**

pdf**(**file **=** "VennDiagramOverlap3groups\_ctrl.pdf", width**=**7, height**=**7**)**

pushViewport**(**viewport**(**width**=**unit**(**0.75, "npc"**)**, height **=** unit**(**0.75, "npc"**)))**

grid.draw**(**v**)**

dev.off**()**

#mock

RE**=**read.table**(**"k100Mm\_AllSenseTEm\_curatedGTF\_exprMock05\_V4. txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 1

dangler**=**read.table**(**"/AllRep\_comp\_intersect\_exprMock\_V4.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 2

df**=**read.table**(**"BedCov\_OverlapBP\_PassM.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 3

RE**=**RE**[!**grepl**(**"\_\_", RE**$**id**)**,**]**

RE**=**RE**[**order**(**RE**$**id**)**,**]**

dangler**=**dangler**[**order**(**dangler**$**id**)**,**]**

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

v**<-**venn.diagram**(**list**(**"count&depth"**=**df**[**,1**]**, "htseq-count"**=**RE**[**,1**]**, "TEFingerprint"**=**dangler**[**,1**])**, filename**=NULL)**

grid.newpage**()**

pdf**(**file **=** "VennDiagramOverlap3groups\_mock.pdf", width**=**7, height**=**7**)**

pushViewport**(**viewport**(**width**=**unit**(**0.75, "npc"**)**, height **=** unit**(**0.75, "npc"**)))**

grid.draw**(**v**)**

dev.off**()**

#yeast

RE**=**read.table**(**"k100Mm\_AllSenseTEm\_curatedGTF\_exprYeast05\_V4.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 1

dangler**=**read.table**(**"AllRep\_comp\_intersect\_exprYeast\_V4.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 2

df**=**read.table**(**"BedCov\_OverlapBP\_PassY.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 3

RE**=**RE**[!**grepl**(**"\_\_", RE**$**id**)**,**]**

RE**=**RE**[**order**(**RE**$**id**)**,**]**

dangler**=**dangler**[**order**(**dangler**$**id**)**,**]**

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

v**<-**venn.diagram**(**list**(**"count&depth"**=**df**[**,1**]**, "htseq-count"**=**RE**[**,1**]**, "TEFingerprint"**=**dangler**[**,1**])**, filename**=NULL)**

grid.newpage**()**

pdf**(**file **=** "VennDiagramOverlap3groups\_yeast.pdf", width**=**7, height**=**7**)**

pushViewport**(**viewport**(**width**=**unit**(**0.75, "npc"**)**, height **=** unit**(**0.75, "npc"**)))**

grid.draw**(**v**)**

dev.off**()**

#botrytis

RE**=**read.table**(**"k100Mm\_AllSenseTEm\_curatedGTF\_exprBotrytis05\_V4.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 1

dangler**=**read.table**(**"AllRep\_comp\_intersect\_exprBotrytis\_V4.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 2

df**=**read.table**(**"BedCov\_OverlapBP\_PassB.txt", header**=**T, sep**=**"\t"**)** #from sub-pipeline 3

RE**=**RE**[!**grepl**(**"\_\_", RE**$**id**)**,**]**

RE**=**RE**[**order**(**RE**$**id**)**,**]**

dangler**=**dangler**[**order**(**dangler**$**id**)**,**]**

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

v**<-**venn.diagram**(**list**(**"count&depth"**=**df**[**,1**]**, "htseq-count"**=**RE**[**,1**]**, "TEFingerprint"**=**dangler**[**,1**])**, filename**=NULL)**

grid.newpage**()**

pdf**(**file **=** "VennDiagramOverlap3groups\_botrytis.pdf", width**=**7, height**=**7**)**

pushViewport**(**viewport**(**width**=**unit**(**0.75, "npc"**)**, height **=** unit**(**0.75, "npc"**)))**

grid.draw**(**v**)**

dev.off**()**

## Comparison of the expression range of under-threshold TEs and expr. candidates of T=0 captured by different methods

#R code

#analyse control samples

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

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

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

count1**=**count**[**,1**:**9**]**

depth1**=**depth**[**,1**:**9**]**

coverage1**=**coverage**[**,1**:**9**]**

count1**$**countMean**=**rowMeans**(**count1**[**,7**:**9**])**

depth1**$**depthMean**=**rowMeans**(**depth1**[**,7**:**9**])**

coverage1**$**coverMean**=**rowMeans**(**coverage1**[**,7**:**9**])**

df**=**cbind**(**count1**[**,c**(**1**:**6,10**)]**, depthMean**=**depth1**$**depthMean, coverMean**=**coverage1**$**coverMean**)**

dfNonZero**=**subset**(**df, df**$**countMean**>**0**)** #nrow=28230

dfNonZero**<-**dfNonZero**[**order**(**dfNonZero**$**TEm**)**,**]**

data**=**read.table**(**"AllExpeCandidate\_ctrl\_tag\_new.txt", header**=**T, sep**=**"\t"**)** #nrow=3698

data**<-**data**[**order**(**data**$**id**)**,**]**

dfNonZero**=**merge**(**dfNonZero, data, by.x**=**"TEm", by.y**=**"id", all.x**=TRUE)**

underthreshold**=**subset**(**dfNonZero, is.na**(**dfNonZero**$**C\_tracking**))** #nrow=24532

dfNonZero**$**C\_tracking**<-** as.character**(**dfNonZero**$**C\_tracking**)**

dfNonZero**$**C\_tracking**<-**replace**(**dfNonZero**$**C\_tracking, is.na**(**dfNonZero**$**C\_tracking**)**, "Under threshold"**)**

dfNonZero**$**C\_tracking**<-**replace**(**dfNonZero**$**C\_tracking, dfNonZero**$**C\_tracking**==**"countdepthonly", "Bedtools only"**)**

dfNonZero**$**C\_tracking**<-**replace**(**dfNonZero**$**C\_tracking, dfNonZero**$**C\_tracking**==**"countdepth\_htseq", "Bedtools and Htseq"**)**

dfNonZero**$**C\_tracking**<-**replace**(**dfNonZero**$**C\_tracking, dfNonZero**$**C\_tracking**==**"countdepth\_htseq\_TEF", "Bedtools, Htseq and TEF"**)**

dfNonZero**$**C\_tracking**<-**replace**(**dfNonZero**$**C\_tracking, dfNonZero**$**C\_tracking**==**"countdepth\_TEF", "Bedtools and TEF"**)**

dfNonZero**$**C\_tracking**<-**replace**(**dfNonZero**$**C\_tracking, dfNonZero**$**C\_tracking**==**"htseqonly", "Htseq only"**)**

dfNonZero**$**C\_tracking**<-**replace**(**dfNonZero**$**C\_tracking, dfNonZero**$**C\_tracking**==**"htseqTEFonly", "Htseq and TEF"**)**

dfNonZero**$**C\_tracking**<-**factor**(**dfNonZero**$**C\_tracking, levels**=**c**(**"Htseq and TEF","Bedtools and TEF","Bedtools only","Bedtools and Htseq","Bedtools, Htseq and TEF","Htseq only","Under threshold"**)**, labels**=**c**(**"Htseq and TEF","Bedtools and TEF","Bedtools only","Bedtools and Htseq","Bedtools, Htseq and TEF","Htseq only","Under threshold"**))**

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

ggplot**(**dfNonZero, aes **(**log2**(**countMean**)**, log2**(**depthMean**)))+**

theme\_bw**()+**

geom\_point**(**aes**(**colour **=** C\_tracking**)**, alpha **=** 0.5, size**=**5**)+**

scale\_color\_manual**(**breaks **=** c**(**"Bedtools and TEF","Htseq and TEF","Bedtools only","Bedtools and Htseq","Bedtools, Htseq and TEF","Htseq only","Under threshold"**)**, values**=**c**(**"magenta", "forestgreen", "darkolivegreen3","goldenrod2", "red", "skyblue2", "gray84"**))+**

geom\_vline**(**xintercept **=** log2**(**10**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**8**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**6**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**4**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**2**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_hline**(**yintercept **=** log2**(**5**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_hline**(**yintercept **=** log2**(**4**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_hline**(**yintercept **=** log2**(**3**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_hline**(**yintercept **=** log2**(**2**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)**

dev.copy**(**png,"TEnonZero\_ctrl.png", width **=**1200, height **=** 900**)**

dev.off**()**

ggplot**(**dfNonZero, aes **(**log2**(**countMean**)))+**

geom\_density**(**size**=**1.5**)** **+**

theme\_bw**()+**

geom\_vline**(**xintercept **=** log2**(**10**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**8**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**6**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**4**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)+**

geom\_vline**(**xintercept **=** log2**(**2**)**, linetype **=** 2, size**=**1.5, colour **=** "gray35"**)**

dev.copy**(**png,"TEnonZero\_countDensity\_ctrl.png", width **=**1200, height **=** 300**)**

dev.off**()**

ggplot**(**dfNonZero, aes **(**log2**(**depthMean**)))+**

geom\_density**(**size**=**1.5**)** **+**

theme\_bw**()+**

geom\_vline**(**xintercept **=** log2**(**5**)**, colour **=** "gray35", linetype **=** 2, size**=**1.5**)+**

geom\_vline**(**xintercept **=** log2**(**4**)**, colour **=** "gray35", linetype **=** 2, size**=**1.5**)+**

geom\_vline**(**xintercept **=** log2**(**3**)**, colour **=** "gray35", linetype **=** 2, size**=**1.5**)+**

geom\_vline**(**xintercept **=** log2**(**2**)**, colour **=** "gray35", linetype **=** 2, size**=**1.5**)+**

coord\_flip**()**

dev.copy**(**png,"TEnonZero\_depthDensity\_ctrl.png", width **=**300, height **=** 900**)**

dev.off**()**

## Calculation of total mapped reads

#!/bin/bash

**for** i **in** **{**01**..**39**};** **do**

bedtools bamtobed **-**i ${i}\_k100Mm.bam **>** ${i}\_k100Mm.bed

**awk** '{print $4}' ${i}\_k100Mm.bed **|** **awk** '!x[$0]++' **-** **|** **wc** -l **>** ${i}\_TotalMappedRead.txt**;**

**done**

## Calculation of reads mapping to genes

#!/bin/bash

#gtf to bed

**awk** **-**F"\t" '{OFS="\t"; print $1,$4-1,$5,$9,$8,$7}' V2.1\_iso1\_exon.gtf **>** V2.1\_iso1\_exon\_StartMinus1.bed

#grab reads overlapping with gene features

**for** i **in** **{**01**..**39**};** **do**

samtools view **-**hL V2.1\_iso1\_exon\_StartMinus1.bed **../**bam\_file**/**${i}\_k100Mm.bam \

**|** samtools view **-**Su **-** \

**|** samtools sort **-** ${i}\_k100Mm.MappedToGenesV2**;**

**done**

#counting number of reads in the file

**for** i **in** **{**01**..**39**};** **do**

bedtools bamtobed **-**i ${i}\_k100Mm.MappedToGenesV2.bam **>** ${i}\_k100Mm.MappedToGenesV2.bed

**awk** '{print $4}' ${i}\_k100Mm.MappedToGenesV2.bed **|** **awk** '!x[$0]++' **-** **|** **wc** -l **>** ${i}\_MappedToGenesV2.txt**;**

**done**

## Calculation of reads mapping to TEs

#!/bin/bash

**awk** **-**F"\t" '{OFS="\t"; print $1,$4-1,$5,$9,$8,$7}' AllTEsExpanded\_curated.gtf \

**>** AllTEsExpanded\_curated\_StartMinus1.bed

**for** i **in** **{**01**..**39**};** **do**

bedtools bamtobed **-**split **-**i ${i}\_k100Mm.bam **>** ${i}\_k100Mm\_split.bed

bedtools intersect -a ${i}\_k100Mm\_split.bed -b AllTEsExpanded\_curated\_StartMinus1.bed **-**wa **-**wb \

**>** ${i}\_k100Mm.AllRESplitTag.bed**;**

**done**

**for** i **in** **{**01**..**39**};** **do**

**awk** '{print $4}' ${i}\_k100Mm.AllRESplitTag.bed **|** **sort** **|** **uniq** **|** **wc** -l \

**>** **./**AllREmappedRead**/**${i}ReadsMappedToRE.txt**;** **done**

## Calculation of reads mapping to TEs in sense orientation

**for** i **in** **{**01**..**39**};** **do**

**awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $6); print}' ${i}\_k100Mm.AllRESplitTag.bed **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $12); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $6); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $12); print}' **\**

**|** **grep** '/1' **\**

**|awk** '$6 != $12' **\**

**>** **./**sense**/**${i}ReadsMappedToRE\_sense1.txt #grep first reads aligned with TEs in different orientation

**awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $6); print}' ${i}\_k100Mm.AllRESplitTag.bed **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $12); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $6); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $12); print}' **\**

**|** **grep** '/2' **\**

**|awk** '$6 == $12' **\**

**>** **./**sense**/**${i}ReadsMappedToRE\_sense2.txt #grep second reads aligned with TEs in same orientation

**cat** **./**sense**/**${i}ReadsMappedToRE\_sense1.txt **./**sense**/**${i}ReadsMappedToRE\_sense2.txt **\**

**|sort** **-**k4 **\**

**>** **./**sense**/**${i}ReadsMappedToRE\_sense.txt**;** #combine read 1 and read 2

**done**

**for** i **in** **{**01**..**39**};** **do** **awk** '{print $4}' **./**sense**/**${i}ReadsMappedToRE\_sense.txt **\**

**|** **uniq** **\**

**|** **wc** -l **\**

**>** **./**sense**/**${i}NumberOfReadMappedToRE\_sense.txt**;**

**done**

## Calculation of reads mapping to TEs in antisense orientation

**for** i **in** **{**01**..**39**};** **do**

**awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $6); print}' ${i}\_k100Mm.AllRESplitTag.bed **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $12); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $6); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $12); print}' **\**

**|** **grep** '/1' **\**

**|awk** '$6 == $12' **\**

**>** **./**antisense**/**${i}ReadsMappedToRE\_antisense1.txt #grep first reads aligned with TEs in same orientation

**awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $6); print}' ${i}\_k100Mm.AllRESplitTag.bed **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\+']","1", $12); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $6); print}' **\**

**|** **awk** **-**F"\t" '{OFS="\t"; gsub( "['\-']","2", $12); print}' **\**

**|** **grep** '/2' **\**

**|awk** '$6 != $12' **\**

**>** **./**antisense**/**${i}ReadsMappedToRE\_antisense2.txt #grep second reads aligned with TEs in different orientation

**cat** **./**antisense**/**${i}ReadsMappedToRE\_antisense1.txt **./**antisense**/**${i}ReadsMappedToRE\_antisense2.txt **\**

**|sort** **-**k4 **\**

**>** **./**antisense**/**${i}ReadsMappedToRE\_antisense.txt**;** #combine read 1 and read 2

**done**

**for** i **in** **{**01**..**39**};** **do** **awk** '{print $4}' **./**antisense**/**${i}ReadsMappedToRE\_antisense.txt **\**

**|** **uniq** **\**

**|** **wc** -l **\**

**>** **./**antisense**/**${i}NumberOfReadMappedToRE\_antisense.txt**;**

**done**

## Calculation of reads mapping to a TE in sense orientation but in antisense to another

#Calculate reads lie sense to a TE but antisense to another

**for** i **in** **{**01**..**39**};** **do**

**awk** **-**F'\t' 'NR==FNR {c[$4]++;next}; c[$4]>0' **\**

**./**sense**/**${i}ReadsMappedToRE\_sense.txt **\**

**./**antisense**/**${i}ReadsMappedToRE\_antisense.txt **\**

**|** **awk** '!x[$4]++' **-** **\**

**|** **wc** -l **\**

**>** **./**senseantisense**/**${i}NumbOfSenseAntisenseRead.txt

**awk** **-**F'\t' 'NR==FNR {c[$4]++;next}; c[$4]==0' **\**

**./**antisense**/**${i}ReadsMappedToRE\_antisense.txt **\**

**./**sense**/**${i}ReadsMappedToRE\_sense.txt **\**

**|** **awk** '!x[$4]++' **-** **\**

**|** **wc** -l **\**

**>** **./**senseantisense**/**${i}NumbOfSenseReadOnly.txt

**awk** **-**F'\t' 'NR==FNR {c[$4]++;next}; c[$4]==0' **\**

**./**sense**/**${i}ReadsMappedToRE\_sense.txt **\**

**./**antisense**/**${i}ReadsMappedToRE\_antisense.txt **\**

**|** **awk** '!x[$4]++' **-** **\**

**|** **wc** -l **\**

**>** **./**senseantisense**/**${i}NumbOfAntisenseReadOnly.txt**;**

**done**

#Collect reads lie sense to a TE but antisense to another

**cd ./**senseantisense

**for** i **in** **{**01**..**39**};** **do**

**awk** **-**F'\t' 'NR==FNR {c[$4]++;next}; c[$4]>0' **\**

**../**sense**/**${i}ReadsMappedToRE\_sense.txt **\**

**../**antisense**/**${i}ReadsMappedToRE\_antisense.txt **\**

**>** ${i}ReadsMappedToRE\_senseantisense\_anti.txt

**awk** **-**F'\t' 'NR==FNR {c[$4]++;next}; c[$4]>0' **\**

**../**antisense**/**${i}ReadsMappedToRE\_antisense.txt **\**

**../**sense**/**${i}ReadsMappedToRE\_sense.txt **\**

**>** ${i}ReadsMappedToRE\_senseantisense\_sense.txt

**cat** ${i}ReadsMappedToRE\_senseantisense\_sense.txt ${i}ReadsMappedToRE\_senseantisense\_anti.txt **\**

**|** **sort** **-**k4 **\**

**>** ${i}ReadsMappedToRE\_senseantisense.txt**;**

**done**

#Calculate the three categories for reads that lie sense to a TE but antisense to another

#Category 1: A read mapped to at least two close TEs of different orientations.

# (it could be a unique-or multi-mapping read)

#Category 2: A read not only mapped to at least two close TEs of different orientations,

# but also mapped in antisense orientation to at least one TE in antisense orientation

# (i.e. it’s a multi-mapping read).

#Category 3: A read not only mapped to a TE in sense orientation,

# but also mapped in antisense orientation to the second TE that is in the same orientation of the # first TE (i.e. it’s a multi-mapping read).

**for** i **in** **{**01**..**39**};** **do**

**sort** **-**k4 ${i}ReadsMappedToRE\_senseantisense.txt **\**

**|** **awk** '!x[$4]++' **-** **\**

**|** **awk** '{print $4}' **-** **\**

**>** ${i}ReadListSenseAntisense.txt

**touch** ${i}SenseAntisenseREstrandDiff.txt

**cat** ${i}ReadListSenseAntisense.txt **\**

**|** **while** **read** line**;** **do**

**grep** "$line" ${i}ReadsMappedToRE\_senseantisense.txt **>** ${i}temp1.txt

**awk** '!x[$12]++' ${i}temp1.txt **>** ${i}temp2.txt #column 12 stores strand information

**cat** ${i}temp2.txt **\**

**|** **wc** -l **>** ${i}temp3.txt

#if the read mapped to TEs of different orientation

#the file ${i}temp3.txt would have more than 1 line

**awk** '{if ($1=="1") print "F"; else print "T"; }' ${i}temp3.txt **\**

**>>** ${i}SenseAntisenseREstrandDiff.txt**;**

**done**

**paste** ${i}ReadListSenseAntisense.txt ${i}SenseAntisenseREstrandDiff.txt **\**

**>** ${i}SenseAntisenseREstrandDiffList.txt

**grep** "F\b" ${i}SenseAntisenseREstrandDiffList.txt **\**

**|** **wc** -l **\**

**>** ${i}NumOfInconsistentRead.txt #should equal to the number category 3

**cat** ${i}SenseAntisenseREstrandDiffList.txt **\**

**|** **grep** "F\b" **-** **\**

**|** **awk** '{print $1}' **-** **\**

**>** ${i}ReadListSenseAntisenseIncon.txt

**touch** ${i}SenseAntisenseReadStrandDiff.txt

**cat** ${i}ReadListSenseAntisenseIncon.txt **\**

**|** **while** **read** line**;** **do**

**grep** "$line" ${i}ReadsMappedToRE\_senseantisense.txt **>** ${i}temp4.txt

**awk** '!x[$6]++' ${i}temp4.txt **>** ${i}temp5.txt #column 6 stores read’s strand info

**cat** ${i}temp5.txt **|** **wc** -l **>** ${i}temp6.txt

**awk** '{if ($1=="1") print "F"; else print "T"; }' ${i}temp6.txt **\**

**>>** ${i}SenseAntisenseReadStrandDiff.txt**;**

**done**

**paste** ${i}ReadListSenseAntisenseIncon.txt ${i}SenseAntisenseReadStrandDiff.txt **\**

**>** ${i}SenseAntisenseReadStrandDiffList.txt

**grep** "T\b" ${i}SenseAntisenseReadStrandDiffList.txt **\**

**|** **wc** -l **\**

**>** ${i}NumOfInconsistentRead\_DiffStrand.txt #category 3 (the read did not map to different oriented TEs, but the read itself was multi-mapped in different orientation)

**cat** ${i}SenseAntisenseREstrandDiffList.txt **\**

**|** **grep** "T\b" **-** **\**

**|** **awk** '{print $1}' **-** **\**

**>** ${i}ReadListSenseAntisenseCon.txt #these are reads mapping to different oriented TEs

**touch** ${i}SenseAntisenseReadStrandDiff\_REStrandDiff.txt

**cat** ${i}ReadListSenseAntisenseCon.txt **\**

**|** **while** **read** line**;** **do**

**grep** "$line" ${i}ReadsMappedToRE\_senseantisense.txt **>** ${i}temp7.txt

**awk** '!x[$6]++' ${i}temp7.txt **>** ${i}temp8.txt #column 6 stores read’s strand info

**cat** ${i}temp8.txt **|** **wc** -l **>** ${i}temp9.txt

**awk** '{if ($1=="1") print "F"; else print "T"; }' ${i}temp9.txt **\**

**>>** ${i}SenseAntisenseReadStrandDiff\_REStrandDiff.txt**;**

**done**

**paste** ${i}ReadListSenseAntisenseCon.txt ${i}SenseAntisenseReadStrandDiff\_REStrandDiff.txt **\**

**>** ${i}SenseAntisenseReadStrandDiff\_REStrandDiff\_List.txt

**grep** "T\b" ${i}SenseAntisenseReadStrandDiff\_REStrandDiff\_List.txt **\**

**|** **wc** –l **\**

**>** ${i}NumOfConsistentRead\_DiffStrand.txt #category 2 (the read did map to different oriented TEs, but the read itself was also multi-mapped in different orientation)

**grep** "F\b" ${i}SenseAntisenseReadStrandDiff\_REStrandDiff\_List.txt **\**

**|** **wc** -l **\**

**>** ${i}NumOfConsistentRead\_SameStrand.txt #category 1 (the read did map to different oriented TEs, and the read itself was not multi-mapped in different orientation)

**rm** -f ${i}temp**\*.**txt**;**

**done**

## Calculation of reads mapping to the expressed TE candidates and Non-candidates

#R code

#generate bed file of Expr. candidates at T=0

df**=**read.table**(**"AllExpeCandidate\_tracking\_ctrl.txt", header**=**T, sep**=**"\t"**)**

ref**=**read.table**(**"AllTEsExpanded\_curated\_StartMinus1.bed", header**=**F, sep**=**"\t"**)**

same.id**<-(**ref**$**V4 %in% df**$**id**)**

data**=**ref**[**same.id,**]**

write.table**(**data, "AllExpeCandidate\_ctrl.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

dataF**=**subset**(**data, data**$**V6**==**"+"**)**

dataR**=**subset**(**data, data**$**V6**==**"-"**)**

write.table**(**dataF, "AllExpeCandidate\_ctrl\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**dataR, "AllExpeCandidate\_ctrl\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

diff.id**<-(!**ref**$**V4 %in% df**$**id**)**

restdata**=**ref**[**diff.id,**]**

write.table**(**restdata, "NotExpeCandidate\_ctrl.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

restF**=**subset**(**restdata, restdata**$**V6**==**"+"**)**

restR**=**subset**(**restdata, restdata**$**V6**==**"-"**)**

write.table**(**restF, "NotExpeCandidate\_ctrl\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**restR, "NotExpeCandidate\_ctrl\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

#generate bed file of Expr. candidates in mock (20181214 updated)

df**=**read.table**(**"AllExpeCandidate\_tracking\_mock.txt", header**=**T, sep**=**"\t"**)**

ref**=**read.table**(**"AllTEsExpanded\_curated\_StartMinus1.bed", header**=**F, sep**=**"\t"**)**

same.id**<-(**ref**$**V4 %in% df**$**id**)**

data**=**ref**[**same.id,**]**

write.table**(**data, "AllExpeCandidate\_mock.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

dataF**=**subset**(**data, data**$**V6**==**"+"**)**

dataR**=**subset**(**data, data**$**V6**==**"-"**)**

write.table**(**dataF, "AllExpeCandidate\_mock\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**dataR, "AllExpeCandidate\_mock\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

diff.id**<-(!**ref**$**V4 %in% df**$**id**)**

restdata**=**ref**[**diff.id,**]**

write.table**(**restdata, "NotExpeCandidate\_mock.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

restF**=**subset**(**restdata, restdata**$**V6**==**"+"**)**

restR**=**subset**(**restdata, restdata**$**V6**==**"-"**)**

write.table**(**restF, "NotExpeCandidate\_mock\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**restR, "NotExpeCandidate\_mock\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

#generate bed file of Expr. candidates in yeast treatment (20181214 updated)

df**=**read.table**(**"AllExpeCandidate\_tracking\_yeast.txt", header**=**T, sep**=**"\t"**)**

ref**=**read.table**(**"AllTEsExpanded\_curated\_StartMinus1.bed", header**=**F, sep**=**"\t"**)**

same.id**<-(**ref**$**V4 %in% df**$**id**)**

data**=**ref**[**same.id,**]**

write.table**(**data, "AllExpeCandidate\_yeast.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

dataF**=**subset**(**data, data**$**V6**==**"+"**)**

dataR**=**subset**(**data, data**$**V6**==**"-"**)**

write.table**(**dataF, "AllExpeCandidate\_yeast\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**dataR, "AllExpeCandidate\_yeast\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

diff.id**<-(!**ref**$**V4 %in% df**$**id**)**

restdata**=**ref**[**diff.id,**]**

write.table**(**restdata, "NotExpeCandidate\_yeast.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

restF**=**subset**(**restdata, restdata**$**V6**==**"+"**)**

restR**=**subset**(**restdata, restdata**$**V6**==**"-"**)**

write.table**(**restF, "NotExpeCandidate\_yeast\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**restR, "NotExpeCandidate\_yeast\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

#generate bed file of Expr. candidates in botrytis treatment (20180613 updated)

df**=**read.table**(**"AllExpeCandidate\_tracking\_botrytis.txt", header**=**T, sep**=**"\t"**)**

ref**=**read.table**(**"AllTEsExpanded\_curated\_StartMinus1.bed", header**=**F, sep**=**"\t"**)**

same.id**<-(**ref**$**V4 %in% df**$**id**)**

data**=**ref**[**same.id,**]**

write.table**(**data, "AllExpeCandidate\_botrytis.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

dataF**=**subset**(**data, data**$**V6**==**"+"**)**

dataR**=**subset**(**data, data**$**V6**==**"-"**)**

write.table**(**dataF, "AllExpeCandidate\_botrytis\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**dataR, "AllExpeCandidate\_botrytis\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

diff.id**<-(!**ref**$**V4 %in% df**$**id**)**

restdata**=**ref**[**diff.id,**]**

write.table**(**restdata, "NotExpeCandidate\_botrytis.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

restF**=**subset**(**restdata, restdata**$**V6**==**"+"**)**

restR**=**subset**(**restdata, restdata**$**V6**==**"-"**)**

write.table**(**restF, "NotExpeCandidate\_botrytis\_Forward.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

write.table**(**restR, "NotExpeCandidate\_botrytis\_Reverse.bed", col.names**=**F, row.names**=**F, sep**=**"\t", quote**=**F**)**

#Capture sense reads mapping to TE candidates

#!/bin/bash

**cd** **/**media**/**ting-hsuan**/**ExtraDrive1**/**PhD**/**analysis**/**ECstress\_TEalignment**/**ExprCandidate

#Forward-oriented TEs

**for** i **in** **{**01**..**03**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b AllExpeCandidate\_ctrl\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF2R1.bed**;**

**done**

**for** i **in** **{**04**..**15**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b AllExpeCandidate\_mock\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF2R1.bed**;**

**done**

**for** i **in** **{**16**..**27**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b AllExpeCandidate\_yeast\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF2R1.bed**;**

**done**

**for** i **in** **{**28**..**39**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b AllExpeCandidate\_botrytis\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF2R1.bed**;**

**done**

#Reverse-oriented TEs

**for** i **in** **{**01**..**03**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b AllExpeCandidate\_ctrl\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF1R2.bed**;**

**done**

**for** i **in** **{**04**..**15**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b AllExpeCandidate\_mock\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF1R2.bed**;**

**done**

**for** i **in** **{**16**..**27**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b AllExpeCandidate\_yeast\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF1R2.bed**;**

**done**

**for** i **in** **{**28**..**39**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b AllExpeCandidate\_botrytis\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmCandidatesSplitF1R2.bed**;**

**done**

#Capture sense reads mapping to non-candidates

#!/bin/bash

**cd** **/**media**/**ting-hsuan**/**ExtraDrive1**/**PhD**/**analysis**/**ECstress\_TEalignment**/**ExprCandidate

#Forward-oriented TEs

**for** i **in** **{**01**..**03**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b NotExpeCandidate\_ctrl\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF2R1.bed**;**

**done**

**for** i **in** **{**04**..**15**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b NotExpeCandidate\_mock\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF2R1.bed**;**

**done**

**for** i **in** **{**16**..**27**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b NotExpeCandidate\_yeast\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF2R1.bed**;**

**done**

**for** i **in** **{**28**..**39**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F2R1\_split.bed **\**

-b NotExpeCandidate\_botrytis\_Forward.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF2R1.bed**;**

**done**

#Reverse-oriented TEs

**for** i **in** **{**01**..**03**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b NotExpeCandidate\_ctrl\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF1R2.bed**;**

**done**

**for** i **in** **{**04**..**15**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b NotExpeCandidate\_mock\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF1R2.bed**;**

**done**

**for** i **in** **{**16**..**27**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b NotExpeCandidate\_yeast\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF1R2.bed**;**

**done**

**for** i **in** **{**28**..**39**};** **do** bedtools intersect **\**

-a **../**${i}\_k100Mm\_F1R2\_split.bed **\**

-b NotExpeCandidate\_botrytis\_Reverse.bed **-**wa **-**wb **\**

**>** **../**AllTEmappedRead**/**${i}\_k100MmNonCandidatesSplitF1R2.bed**;**

**done**

#calculate sense read mapping to candidates or non-candidates

**cd** **/**media**/**ting-hsuan**/**ExtraDrive1**/**PhD**/**analysis**/**ECstress\_TEalignment**/**AllTEmappedRead

**for** i **in** **{**01**..**39**};** **do**

**cat** ${i}\_k100MmCandidatesSplitF2R1.bed ${i}\_k100MmCandidatesSplitF1R2.bed **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**>** ${i}\_senseReadMapToCand.txt

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**>** ${i}\_senseReadMapToNonCand.txt**;**

**done**

**for** i **in** **{**01**..**39**};** **do**

**awk** **-**F'\t' 'NR==FNR {c[$1]++;next}; c[$1]==0' ${i}\_senseReadMapToNonCand.txt ${i}\_senseReadMapToCand.txt **\**

**|** **sort** **|** **uniq** **\**

**>** ${i}\_senseReadMapToCandOnly.txt #capture sense read mapping to candidate only

**awk** **-**F'\t' 'NR==FNR {c[$1]++;next}; c[$1]==1' ${i}\_senseReadMapToNonCand.txt ${i}\_senseReadMapToCand.txt **\**

**|** **sort** **|** **uniq** **\**

**>** ${i}\_senseReadMapToCandAndNonCand.txt #capture sense read mapping to both candidate and non-candidate

**awk** **-**F'\t' 'NR==FNR {c[$1]++;next}; c[$1]==0' ${i}\_senseReadMapToCand.txt ${i}\_senseReadMapToNonCand.txt **\**

**|** **sort** **|** **uniq** **\**

**>** ${i}\_senseReadMapToNonCandOnly.txt**;** **done** #capture sense read mapping to non-candidate only

**for** i **in** **{**01**..**39**};** **do**

**cat** ${i}\_senseReadMapToCandOnly.txt **|** **sort** **|** **uniq** **|** **wc** -l **>** ${i}\_NumOfsenseReadMapToCandOnly.txt

**cat** ${i}\_senseReadMapToCandAndNonCand.txt **|** **sort** **|** **uniq** **|** **wc** -l **>** ${i}\_NumOfsenseReadMapToCandAndNonCand.txt

**cat** ${i}\_senseReadMapToNonCandOnly.txt **|** **sort** **|** **uniq** **|** **wc** -l **>** ${i}\_NumOfsenseReadMapToNonCandOnly.txt**;**

**done**

#capture under-threshold TEs, and categorize them by read cound >10 or <=10

#R code

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

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

data**=**count

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

#ctrl

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

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

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

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

temp**=**subset**(**newdata**,** newdata**$CountMean>**0**)**

df1**=**subset**(**temp**,** temp**$CountMean<=**10**)**

df2**=**subset**(**newdata**,** newdata**$CountMean>**10**)**

df1**=**data.frame**(**id**=**df1**[,**1**])**

df2**=**data.frame**(**id**=**df2**[,**1**])**

write.table**(**df1**,** "UnderThreshold\_countLthan10\_ctrl.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

write.table**(**df2**,** "UnderThreshold\_countHthan10\_ctrl.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

#mock

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

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

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

newdata**$CountMean1h=**rowMeans**(**newdata**[,**10**:**21**])**

temp**=**subset**(**newdata**,** newdata**$CountMean>**0**)**

df1**=**subset**(**temp**,** temp**$CountMean<=**10**)**

df2**=**subset**(**newdata**,** newdata**$CountMean>**10**)**

df1**=**data.frame**(**id**=**df1**[,**1**])**

df2**=**data.frame**(**id**=**df2**[,**1**])**

write.table**(**df1**,** "UnderThreshold\_countLthan10\_mock.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

write.table**(**df2**,** "UnderThreshold\_countHthan10\_mock.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

#yeast

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

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

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

newdata**$CountMean=**rowMeans**(**newdata**[,**22**:**33**])**

temp**=**subset**(**newdata**,** newdata**$CountMean>**0**)**

df1**=**subset**(**temp**,** temp**$CountMean<=**10**)**

df2**=**subset**(**newdata**,** newdata**$CountMean>**10**)**

df1**=**data.frame**(**id**=**df1**[,**1**])**

df2**=**data.frame**(**id**=**df2**[,**1**])**

write.table**(**df1**,** "UnderThreshold\_countLthan10\_yeast.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

write.table**(**df2**,** "UnderThreshold\_countHthan10\_yeast.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

#botrytis

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

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

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

newdata**$CountMean=**rowMeans**(**newdata**[,**34**:**45**])**

temp**=**subset**(**newdata**,** newdata**$CountMean>**0**)**

df1**=**subset**(**temp**,** temp**$CountMean<=**10**)**

df2**=**subset**(**newdata**,** newdata**$CountMean>**10**)**

df1**=**data.frame**(**id**=**df1**[,**1**])**

df2**=**data.frame**(**id**=**df2**[,**1**])**

write.table**(**df1**,** "UnderThreshold\_countLthan10\_botrytis.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

write.table**(**df2**,** "UnderThreshold\_countHthan10\_botrytis.txt"**,** col.names**=**F**,** row.names**=**F**,** sep**=**"\t"**,** quote**=**F**)**

#capture reads mapping to under-threshold TEs that were categorized by read count threshold

#!/bin/bash

**cd** **/**media**/**ting-hsuan**/**ExtraDrive1**/**PhD**/**analysis**/**ECstress\_TEalignment**/**AllTEmappedRead

#ctrl

**for** i **in** **{**01**..**03**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countHthan10\_ctrl.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **\**

**>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountHthan10.txt**;**

**done**

**for** i **in** **{**01**..**03**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countLthan10\_ctrl.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **\**

**>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountLthan10.txt**;**

**done**

#mock

**for** i **in** **{**04**..**15**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countHthan10\_mock.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **\**

**>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountHthan10.txt**;**

**Done**

**for** i **in** **{**04**..**15**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countLthan10\_mock.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **\**

**>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountLthan10.txt**;**

**done**

#yeast

**for** i **in** **{**16**..**27**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countHthan10\_yeast.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **\**

**>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountHthan10.txt**;**

**done**

**for** i **in** **{**16**..**27**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countLthan10\_yeast.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **\**

**>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountLthan10.txt**;**

**done**

#botrytis

**for** i **in** **{**28**..**39**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countHthan10\_botrytis.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountHthan10.txt**;** **done**

**for** i **in** **{**28**..**39**};** **do**

**cat** ${i}\_k100MmNonCandidatesSplitF2R1.bed ${i}\_k100MmNonCandidatesSplitF1R2.bed **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$10]>0' **../**ExprCandidate**/**UnderThreshold\_countLthan10\_botrytis.txt **-** **\**

**|** **awk** '{print $4}' **-** **|** **sort** **|** **uniq** **\**

**|** **awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **-** ${i}\_senseReadMapToNonCandOnly.txt **\**

**>** ${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountLthan10.txt**;** **done**

#calculate reads mapping to under-threshold TEs that were categorized by read count threshold

#!/bin/bash

**for** i **in** **{**01**..**39**};** **do**

**awk** 'NR==FNR {c[$1]++;next}; c[$1]==0' **\**

${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountHthan10.txt **\**

${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountLthan10.txt **\**

**|** **wc** -l **>** ${i}\_NumOfSenseReadMapToNonCandOnly\_MapTEnonCandLthan10only.txt

**awk** 'NR==FNR {c[$1]++;next}; c[$1]==0' **\**

${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountLthan10.txt **\**

${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountHthan10.txt **\**

**|** **wc** -l **>** ${i}\_NumOfSenseReadMapToNonCandOnly\_MapTEnonCandHthan10only.txt

**awk** 'NR==FNR {c[$1]++;next}; c[$1]>0' **\**

${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountHthan10.txt **\**

${i}\_senseReadMapToNonCandOnly\_MapUnderThresholdCountLthan10.txt **\**

**|** **wc** -l **>** ${i}\_NumOfSenseReadMapToNonCandOnly\_MapTEnonCandHLintersect.txt**;**

**done**

#check the read count and average read depth of mapped region of non-candidates

#ctrl

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

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

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

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

count**$**depth\_ctrl\_a**=**depth**$**Ctrl\_a

count**$**depth\_ctrl\_b**=**depth**$**Ctrl\_b

count**$**depth\_ctrl\_c**=**depth**$**Ctrl\_c

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**])**

newdata**$**DepthMean**=**rowMeans**(**newdata**[**,46**:**48**])**

df1**=**subset**(**newdata, newdata**$**CountMean**==**"0"**)** #nrow=195181

df1.1**=**subset**(**newdata, newdata**$**DepthMean**==**"0"**)** #nrow=195181

df2**=**subset**(**newdata, newdata**$**DepthMean**>**0 **)**

df2.1**=**subset**(**df2, df2**$**DepthMean**<=**1 **)** #nrow=12838

df2.2**=**subset**(**df2.1, df2.1**$**CountMean**<=**10 **)** #nrow=12838

df3**=**subset**(**newdata, newdata**$**DepthMean**>**1 **)**

df3.1**=**subset**(**df3, df3**$**DepthMean**<=**2 **)** #nrow=7054

df3.2**=**subset**(**df3.1, df3.1**$**CountMean**<=**10 **)** #nrow=6906

df4**=**subset**(**newdata, newdata**$**DepthMean**>**2 **)**

df4.1**=**subset**(**df4, df4**$**DepthMean**<=**3 **)** #nrow=2875

df4.2**=**subset**(**df4.1, df4.1**$**CountMean**<=**10 **)** #nrow=2329

df5**=**subset**(**newdata, newdata**$**DepthMean**>**3 **)**

df5.1**=**subset**(**df5, df5**$**DepthMean**<=**4 **)** #nrow=1046

df5.2**=**subset**(**df5.1, df5.1**$**CountMean**<=**10 **)** #nrow=680

df6**=**subset**(**newdata, newdata**$**DepthMean**>**4 **)**

df6.1**=**subset**(**df6, df6**$**DepthMean**<=**5 **)** #nrow=539

df6.2**=**subset**(**df6.1, df6.1**$**CountMean**<=**10 **)** #nrow=294

df7**=**subset**(**newdata, newdata**$**DepthMean**>**5 **)** #nrow=180

#mock

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

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

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

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

data**=**cbind**(**count, depth**[**,10**:**21**])**

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**[**,10**:**21**])**

newdata**$**DepthMean**=**rowMeans**(**newdata**[**,46**:**57**])**

df1**=**subset**(**newdata, newdata**$**CountMean**==**"0"**)** #nrow=179568

df1.1**=**subset**(**newdata, newdata**$**DepthMean**==**"0"**)** #nrow=179568

df2**=**subset**(**newdata, newdata**$**DepthMean**>**0 **)**

df2.1**=**subset**(**df2, df2**$**DepthMean**<=**1 **)** #nrow=27796

df2.2**=**subset**(**df2.1, df2.1**$**CountMean**<=**10 **)** #nrow=27796

df3**=**subset**(**newdata, newdata**$**DepthMean**>**1 **)**

df3.1**=**subset**(**df3, df3**$**DepthMean**<=**2 **)** #nrow=7312

df3.2**=**subset**(**df3.1, df3.1**$**CountMean**<=**10 **)** #nrow=7252

df4**=**subset**(**newdata, newdata**$**DepthMean**>**2 **)**

df4.1**=**subset**(**df4, df4**$**DepthMean**<=**3 **)** #nrow=2341

df4.2**=**subset**(**df4.1, df4.1**$**CountMean**<=**10 **)** #nrow=1894

df5**=**subset**(**newdata, newdata**$**DepthMean**>**3 **)**

df5.1**=**subset**(**df5, df5**$**DepthMean**<=**4 **)** #nrow=707

df5.2**=**subset**(**df5.1, df5.1**$**CountMean**<=**10 **)** #nrow=408

df6**=**subset**(**newdata, newdata**$**DepthMean**>**4 **)**

df6.1**=**subset**(**df6, df6**$**DepthMean**<=**5 **)** #nrow=138

df6.2**=**subset**(**df6.1, df6.1**$**CountMean**<=**10 **)** #nrow=96

df7**=**subset**(**newdata, newdata**$**DepthMean**>**5 **)** #nrow=25

#yeast

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

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

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

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

data**=**cbind**(**count, depth**[**,22**:**33**])**

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**[**,22**:**33**])**

newdata**$**DepthMean**=**rowMeans**(**newdata**[**,46**:**57**])**

df1**=**subset**(**newdata, newdata**$**CountMean**==**"0"**)** #nrow=168340

df1.1**=**subset**(**newdata, newdata**$**DepthMean**==**"0"**)** #nrow=168340

df2**=**subset**(**newdata, newdata**$**DepthMean**>**0 **)**

df2.1**=**subset**(**df2, df2**$**DepthMean**<=**1 **)** #nrow=38306

df2.2**=**subset**(**df2.1, df2.1**$**CountMean**<=**10 **)** #nrow=38306

df3**=**subset**(**newdata, newdata**$**DepthMean**>**1 **)**

df3.1**=**subset**(**df3, df3**$**DepthMean**<=**2 **)** #nrow=8710

df3.2**=**subset**(**df3.1, df3.1**$**CountMean**<=**10 **)** #nrow=8586

df4**=**subset**(**newdata, newdata**$**DepthMean**>**2 **)**

df4.1**=**subset**(**df4, df4**$**DepthMean**<=**3 **)** #nrow=1956

df4.2**=**subset**(**df4.1, df4.1**$**CountMean**<=**10 **)** #nrow=1401

df5**=**subset**(**newdata, newdata**$**DepthMean**>**3 **)**

df5.1**=**subset**(**df5, df5**$**DepthMean**<=**4 **)** #nrow=511

df5.2**=**subset**(**df5.1, df5.1**$**CountMean**<=**10 **)** #nrow=265

df6**=**subset**(**newdata, newdata**$**DepthMean**>**4 **)**

df6.1**=**subset**(**df6, df6**$**DepthMean**<=**5 **)** #nrow=50

df6.2**=**subset**(**df6.1, df6.1**$**CountMean**<=**10 **)** #nrow=40

df7**=**subset**(**newdata, newdata**$**DepthMean**>**5 **)** #nrow=7

#botrytis

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

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

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

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

data**=**cbind**(**count, depth**[**,34**:**45**])**

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**[**,34**:**45**])**

newdata**$**DepthMean**=**rowMeans**(**newdata**[**,46**:**57**])**

df1**=**subset**(**newdata, newdata**$**CountMean**==**"0"**)** #nrow=168829

df1.1**=**subset**(**newdata, newdata**$**DepthMean**==**"0"**)** #nrow=168829

df2**=**subset**(**newdata, newdata**$**DepthMean**>**0 **)**

df2.1**=**subset**(**df2, df2**$**DepthMean**<=**1 **)** #nrow=36742

df2.2**=**subset**(**df2.1, df2.1**$**CountMean**<=**10 **)** #nrow=36742

df3**=**subset**(**newdata, newdata**$**DepthMean**>**1 **)**

df3.1**=**subset**(**df3, df3**$**DepthMean**<=**2 **)** #nrow=9568

df3.2**=**subset**(**df3.1, df3.1**$**CountMean**<=**10 **)** #nrow=9425

df4**=**subset**(**newdata, newdata**$**DepthMean**>**2 **)**

df4.1**=**subset**(**df4, df4**$**DepthMean**<=**3 **)** #nrow=2408

df4.2**=**subset**(**df4.1, df4.1**$**CountMean**<=**10 **)** #nrow=1813

df5**=**subset**(**newdata, newdata**$**DepthMean**>**3 **)**

df5.1**=**subset**(**df5, df5**$**DepthMean**<=**4 **)** #nrow=594

df5.2**=**subset**(**df5.1, df5.1**$**CountMean**<=**10 **)** #nrow=323

df6**=**subset**(**newdata, newdata**$**DepthMean**>**4 **)**

df6.1**=**subset**(**df6, df6**$**DepthMean**<=**5 **)** #nrow=91

df6.2**=**subset**(**df6.1, df6.1**$**CountMean**<=**10 **)** #nrow=61

df7**=**subset**(**newdata, newdata**$**DepthMean**>**5 **)** #nrow=8