Phylogenomics of Pyrenodesmia

Fernando Fernandez Mendoza

7/6/2021

Contents

1 Introduction

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see http://rmarkdown.rstudio.com.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

1.1 Data input

The first parts of the script retrieve data sources,

```
## Registered S3 method overwritten by 'ggtree':
##   method      from
##   identify.gg ggfun
```

```
## ggtree v3.2.1  For help: https://yulab-smu.top/treedata-book/
##
## If you use ggtree in published research, please cite the most appropriate paper(s):
##
## 1. Guangchuang Yu. Using ggtree to visualize data on tree-like structures. Current Protocols in Bioi
## 2. Guangchuang Yu, Tommy Tsan-Yuk Lam, Huachen Zhu, Yi Guan. Two methods for mapping and visualizing
## 3. Guangchuang Yu, David Smith, Huachen Zhu, Yi Guan, Tommy Tsan-Yuk Lam. ggtree: an R package for vi
```

```
##
## Attaching package: 'ggtree'
```

```
## The following object is masked from 'package:ape':
##
##     rotate
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.6-2
```

```
## Package 'mclust' version 5.4.9
## Type 'citation("mclust")' for citing this R package in publications.
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
## Loading required package: clst
```

```
##
## Attaching package: 'clst'
```

```
## The following object is masked from 'package:dplyr':
##
```

```
##     pull
```

```
## Loading required package: rjson
```

2 The phylogenomic dataset

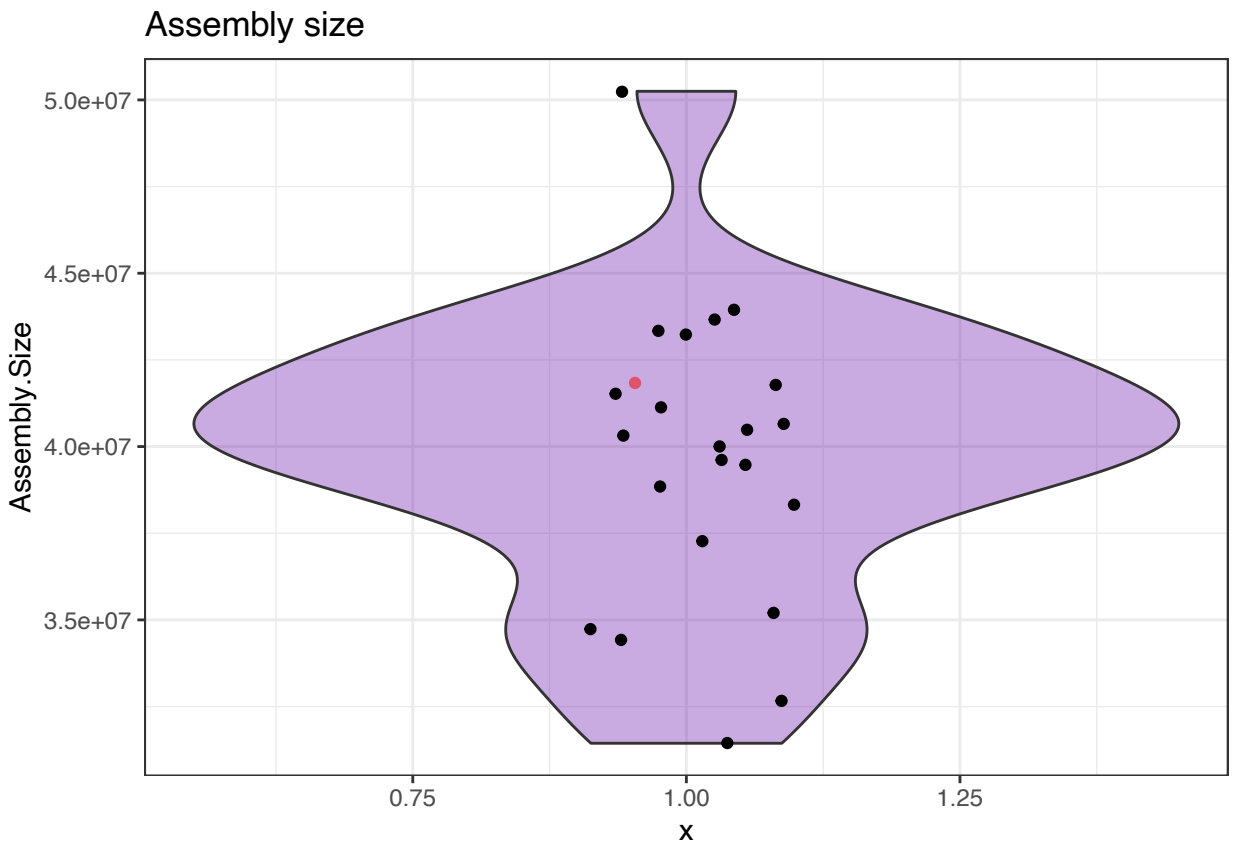
```
synopsis<-read.delim2("/Users/Fernando/Desktop/01_paper_sanger_drive/genome_stats.txt")
kable(synopsis)
```

Species	Assembly_size	Longest_Scaffold	Average_Scaffold	Number_Scaffolds	Scaffold_N50	Percent_GC	Num_Genes	Num_Unique_Genes	tRNAs	Proteins	Proteins_at_least_1_ortholog	Single_copy_orthologs
Xanthoria_parietina	31900637	3929677	817965	39	1731186	49.73	9488	9431	57	1390	8041	3021
Gyalolechia_flavorubescens	34468238	3816824	957451	36	1693300	41.79	9880	9823	57	1572	8251	3021
Pyrenodesmia_erodens	41851372	3781653	1162538	36	1704480	44.05	9416	9370	46	386	8984	3021
Pyrenodesmia_X2	43348608	3837403	240826	180	376967	42.31	10320	10272	48	474	9798	3021
Pyrenodesmia_X3	37275908	3320638	248506	150	419601	45.53	9898	9850	48	212	9638	3021
Pyrenodesmia_X4	41798402	2794185	435400	96	731853	43.38	10056	10010	46	319	9691	3021
Pyrenodesmia_X5	34428752	2072771	180255	191	312859	47.01	9695	9652	43	278	9374	3021
Pyrenodesmia_X6	35186806	1032653	129363	272	189335	45.97	9112	9070	42	281	8789	3021
Pyrenodesmia_X7	40660310	0499439	280416	145	429105	42.90	9722	9625	97	256	9369	3021
Pyrenodesmia_X8	39614804	1433674	309491	128	468261	44.39	10050	10006	44	223	9783	3021
Pyrenodesmia_X9	43659406	6131763	263008	166	423490	42.62	9808	9762	46	389	9371	3021
Pyrenodesmia_Y1	34734307	959625	149074	233	219802	47.43	10006	9960	46	342	9615	3021
Pyrenodesmia_Y2	50248551	5799656	380671	132	745020	39.44	9228	9178	50	330	8848	3021
Pyrenodesmia_Y3	41509736	6945962	334756	124	571577	44.22	9989	9943	46	258	9685	3021
Pyrenodesmia_Y4	31441282	297770	40206	782	65651	44.38	8107	8069	38	420	7646	3021
Pyrenodesmia_Y5	41119244	1867145	345540	119	583469	43.92	9939	9873	66	202	9671	3021
Pyrenodesmia_Y6	39489925	1532758	278098	142	426766	43.50	9955	9908	47	237	9671	3021
Pyrenodesmia_Y7	43221143	3712518	294021	147	584803	42.04	9960	9883	77	329	9552	3021
Pyrenodesmia_Y8	43948905	1539690	240158	183	418742	43.08	10115	10058	57	573	9480	3021
Pyrenodesmia_Y9	38308282	3270234	517680	74	905955	44.72	9585	9448	137	328	9120	3021
Pyrenodesmia_Y10	38841502	2412088	227143	171	360919	45.12	10123	10075	48	267	9808	3021
Pyrenodesmia_Y11	40472199	1156999	249828	162	349194	43.08	10018	9971	47	319	9652	3021
Pyrenodesmia_Y12	32661815	960224	102388	319	183671	48.72	10022	9976	46	473	9501	3021

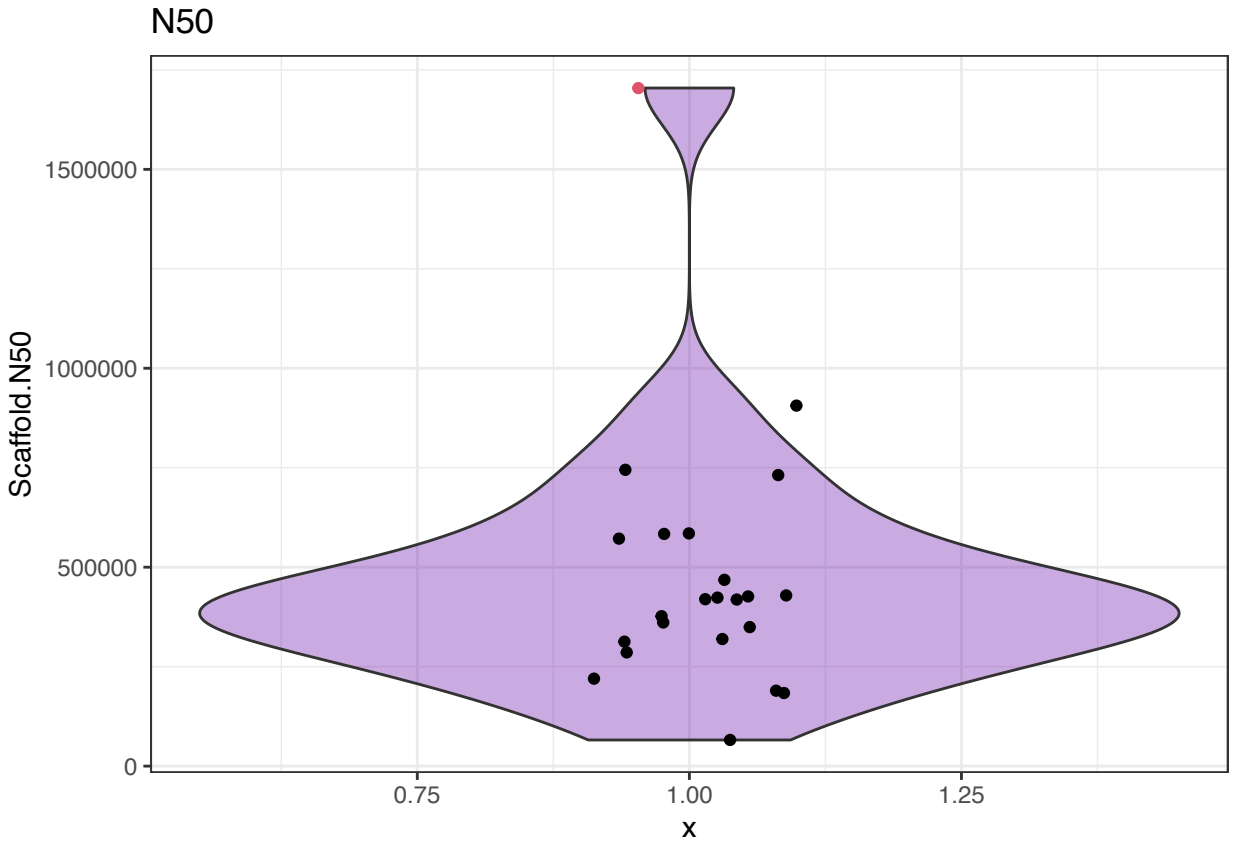
Species	Assembly.Size	Largest.Scaffold	Average.Scaffold	Number.Scaffolds	Scaffold.N50	Percent.GC	Num.Genes	Num.Unique.Genes	Num.tRNAs	Num.Unique.Proteins	Proteins.at.least.Single.copy.orthologs
Pyrenodesmia Y13	40315951	1335582	180789	223	285734	44.40	9041	8986	55	295	8691	3021
Pyrenodesmia Y14	39991836	1021580	186009	215	319254	43.42	9817	9773	44	365	9408	3021

```
synopsis_backup<-synopsis
synopsis<-synopsis[-c(1,2),]
```

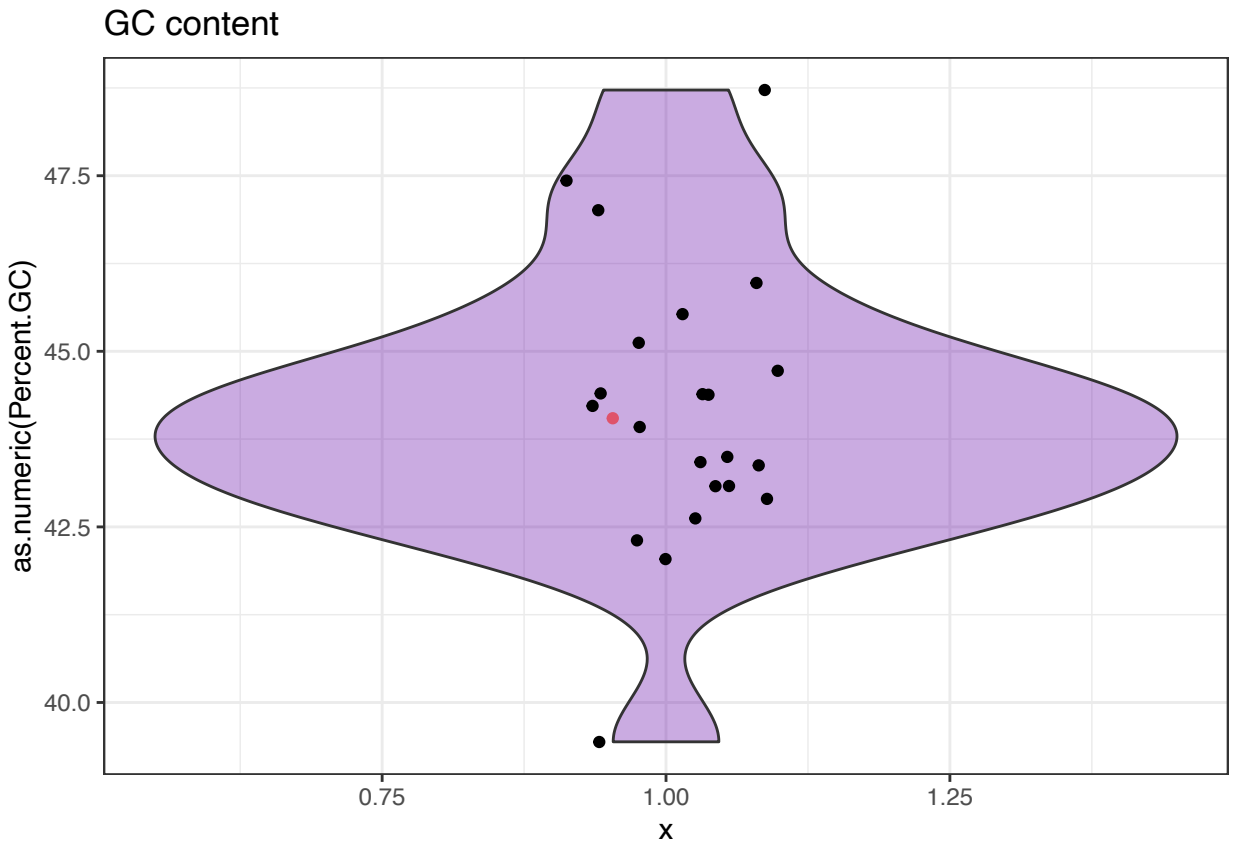
```
ggplot(synopsis,aes(x=1,y=Assembly.Size))+geom_violin(fill = "#6001A655")+geom_point(position = positio
```



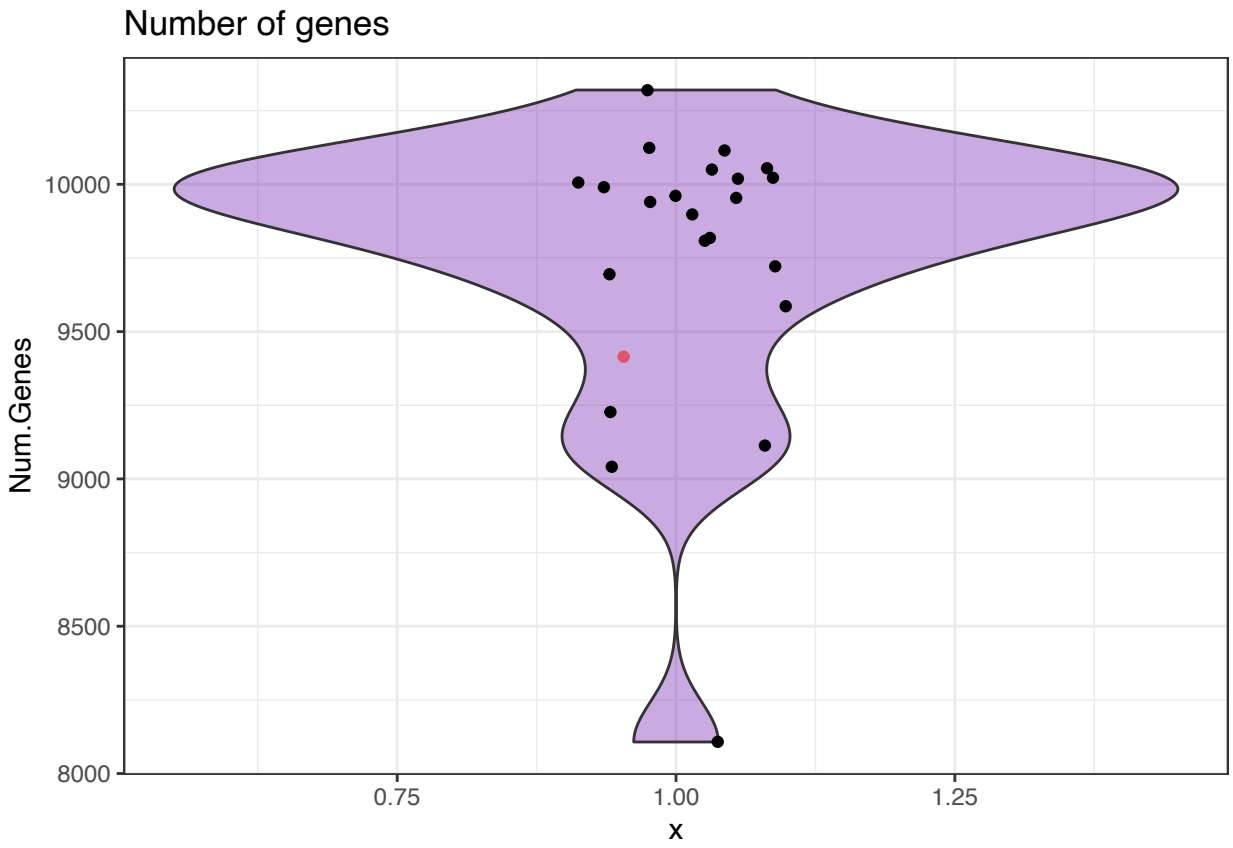

```
ggplot(synopsis,aes(x=1,y=Scaffold.N50))+geom_violin(fill = "#6001A655")+geom_point(position = position_
```



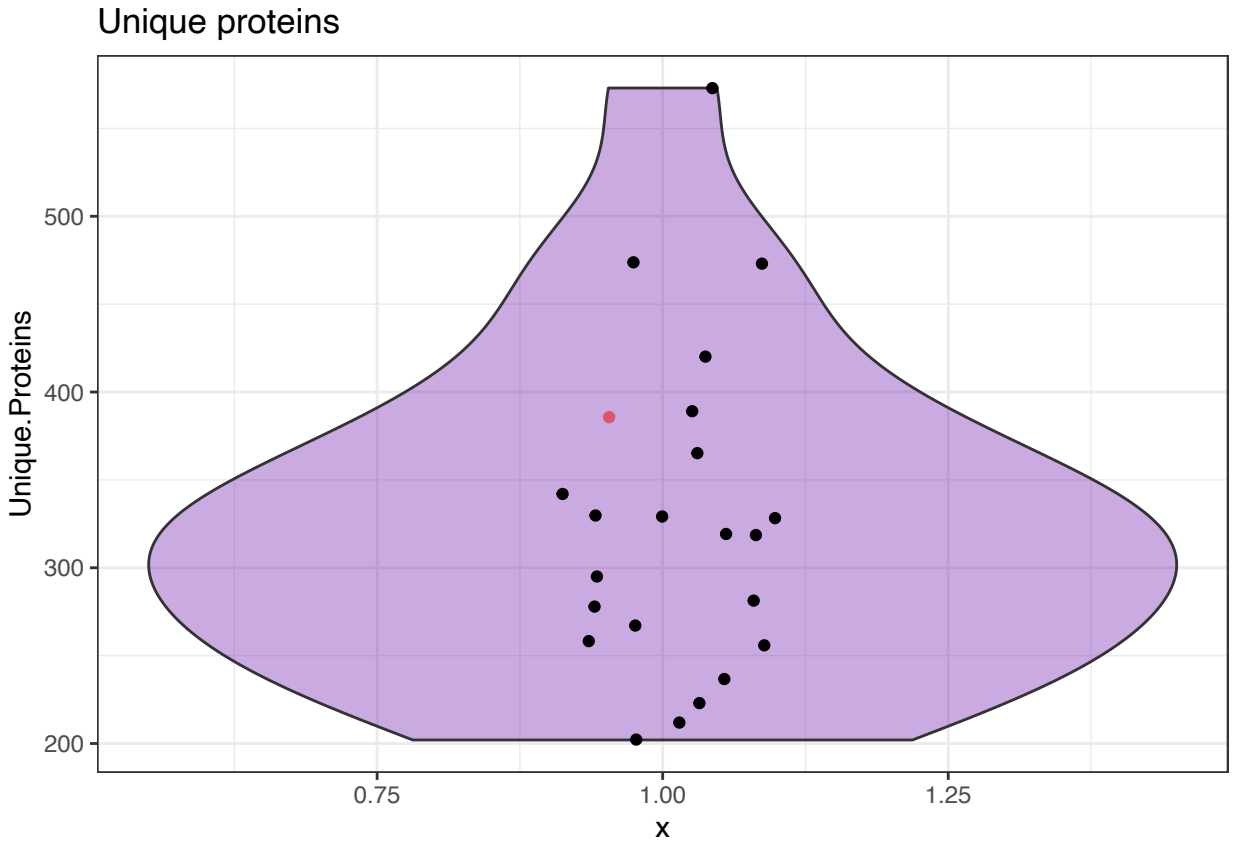

```
ggplot(synopsis,aes(x=1,y=as.numeric(Percent.GC)))+geom_violin(fill = "#6001A655")+geom_point(position =
```




```
ggplot(synopsis,aes(x=1,y=Num.Genes))+geom_violin(fill = "#6001A655")+geom_point(position = position_ji
```

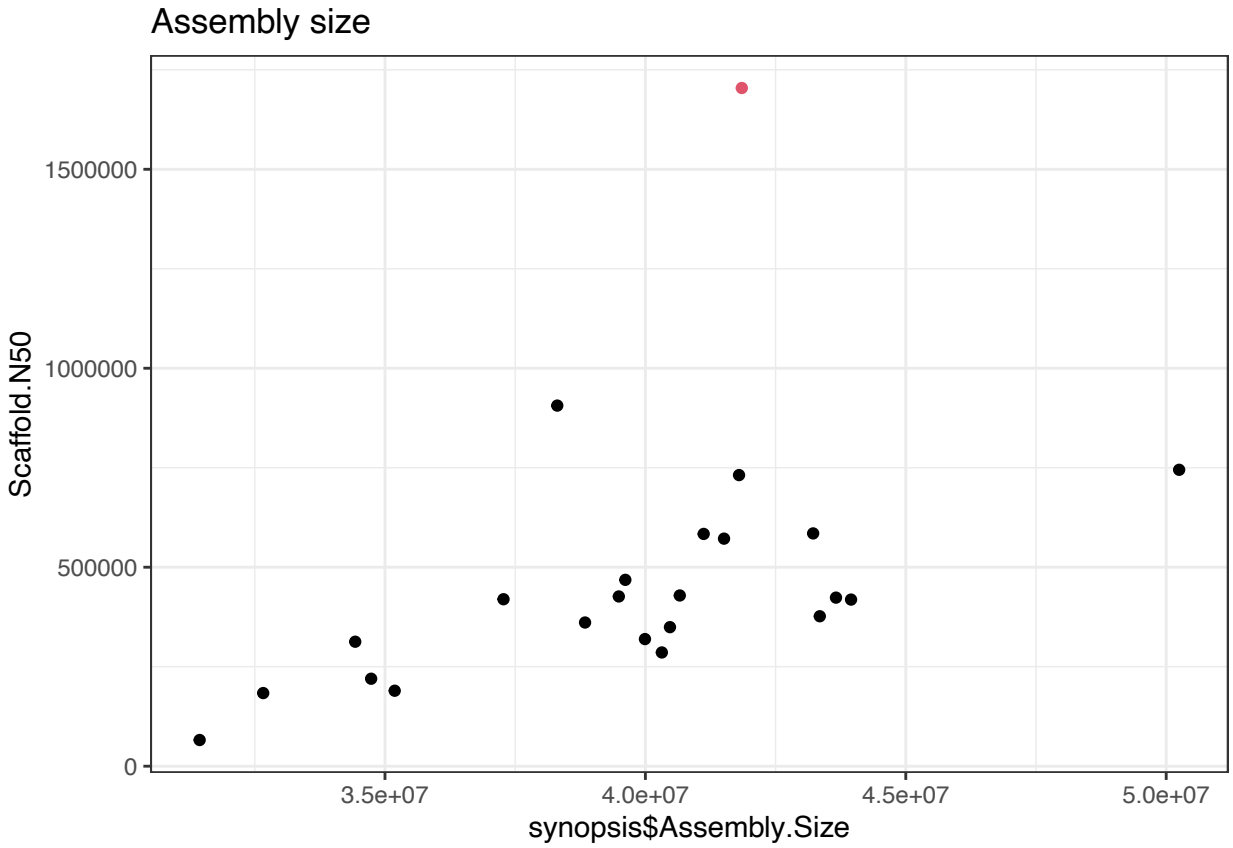



```
ggplot(synopsis,aes(x=1,y=Unique.Proteins))+geom_violin(fill = "#6001A655")+geom_point(position = posit
```




```
ggplot(synopsis,aes(x=synopsis$Assembly.Size,y=Scaffold.N50))+geom_point(position = position_jitter(seed
```

```
## Warning: Use of `synopsis$Assembly.Size` is discouraged. Use `Assembly.Size`
## instead.
```



3 Description of the Pyrenodesmia erodens genome

```
library(karyoploteR)
```

```
## Loading required package: regioneR
## Loading required package: GenomicRanges
## Loading required package: stats4
## Loading required package: BiocGenerics
##
## Attaching package: 'BiocGenerics'
## The following objects are masked from 'package:dplyr':
##
##     combine, intersect, setdiff, union
## The following object is masked from 'package:TreeTools':
##
##     match
## The following objects are masked from 'package:stats':
##
##     IQR, mad, sd, var, xtabs
## The following objects are masked from 'package:base':
##
##     anyDuplicated, append, as.data.frame, basename, cbind, colnames,
```

```
##     dirname, do.call, duplicated, eval, evalq, Filter, Find, get, grep,
##     grepl, intersect, is.unsorted, lapply, Map, mapply, match, mget,
##     order, paste, pmax, pmax.int, pmin, pmin.int, Position, rank,
##     rbind, Reduce, rownames, sapply, setdiff, sort, table, tapply,
##     union, unique, unsplit, which.max, which.min

## Loading required package: S4Vectors

##
## Attaching package: 'S4Vectors'

## The following objects are masked from 'package:dplyr':
##
##     first, rename

## The following object is masked from 'package:ggtree':
##
##     expand

## The following objects are masked from 'package:base':
##
##     expand.grid, I, unname

## Loading required package: IRanges

##
## Attaching package: 'IRanges'

## The following objects are masked from 'package:dplyr':
##
##     collapse, desc, slice

## The following object is masked from 'package:ggtree':
##
##     collapse

## Loading required package: GenomeInfoDb
```

```
library(GenomicRanges)
repbase<-read.delim("/Users/Fernando/Desktop/01_paper_sanger_drive/P_erodens_sorted_relimited.fas.out",
repbase<-repbase[!(is.na(repbase$start)),]
repbase$strand[repbase$strand=="C"]<-"-"
repetitive_sequences<-repbase[repbase$class=="Simple_repeat",]
mobile_elements<-repbase[repbase$class!="Simple_repeat",]
mobile_elements<-mobile_elements[mobile_elements$class!="Low_complexity",]
# Smith Watermann score
mobile_elements<-mobile_elements[mobile_elements$SW_score>=100,]
#mobile_elements<-mobile_elements[mobile_elements$class!="Unknown",]
#mobile_elements<-mobile_elements[mobile_elements$class!="Unspecified",]
dna1<-makeGRangesFromDataFrame(mobile_elements[mobile_elements$class%in%c("DNA/hAT-Ac","DNA/MULE-MuDR")
                         keep.extra.columns=FALSE,
                         ignore.strand=FALSE,
                         seqinfo=NULL,
                         seqnames.field="query_seq",
                         start.field="start",
                         end.field="end",
                         strand.field="strand",
                         starts.in.df.are.0based=FALSE)
ltrcopia<-makeGRangesFromDataFrame(mobile_elements[mobile_elements$class=="LTR/Copia",],
```

```
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="query_seq",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
ltrgypsy<-makeGRangesFromDataFrame(mobile_elements[mobile_elements$class=="LTR/Gypsy",],
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="query_seq",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
ltrNgaro<-makeGRangesFromDataFrame(mobile_elements[mobile_elements$class=="LTR/Ngaro",],
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="query_seq",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
helitron<-makeGRangesFromDataFrame(mobile_elements[mobile_elements$class=="RC/Helitron",],
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="query_seq",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)


telomeres<-repetitive_sequences[repetitive_sequences$rpeat=="(AACCCT)n",]
telomeres<-makeGRangesFromDataFrame(telomeres,
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="query_seq",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)

#repetitive_sequences<-repetitive_sequences[repetitive_sequences$rpeat!="(AACCCT)n",]
repetitive_sequences<-makeGRangesFromDataFrame(repetitive_sequences,
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
```

```
                          seqnames.field="query_seq",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
mobile_elements<-makeGRangesFromDataFrame(mobile_elements,
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="query_seq",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
ranges_genes2<-makeGRangesFromDataFrame(genes2,
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="scaffold",
                          start.field="start",
                          end.field="end",
                          starts.in.df.are.0based=FALSE)

pe.genome <- toGRanges(data.frame(chr=names(pe_genome), start=rep(1,length(pe_genome)), end=sapply(pe_g
```

3.1 Group Large RIP Affected Regions based on their content in repeatable elements

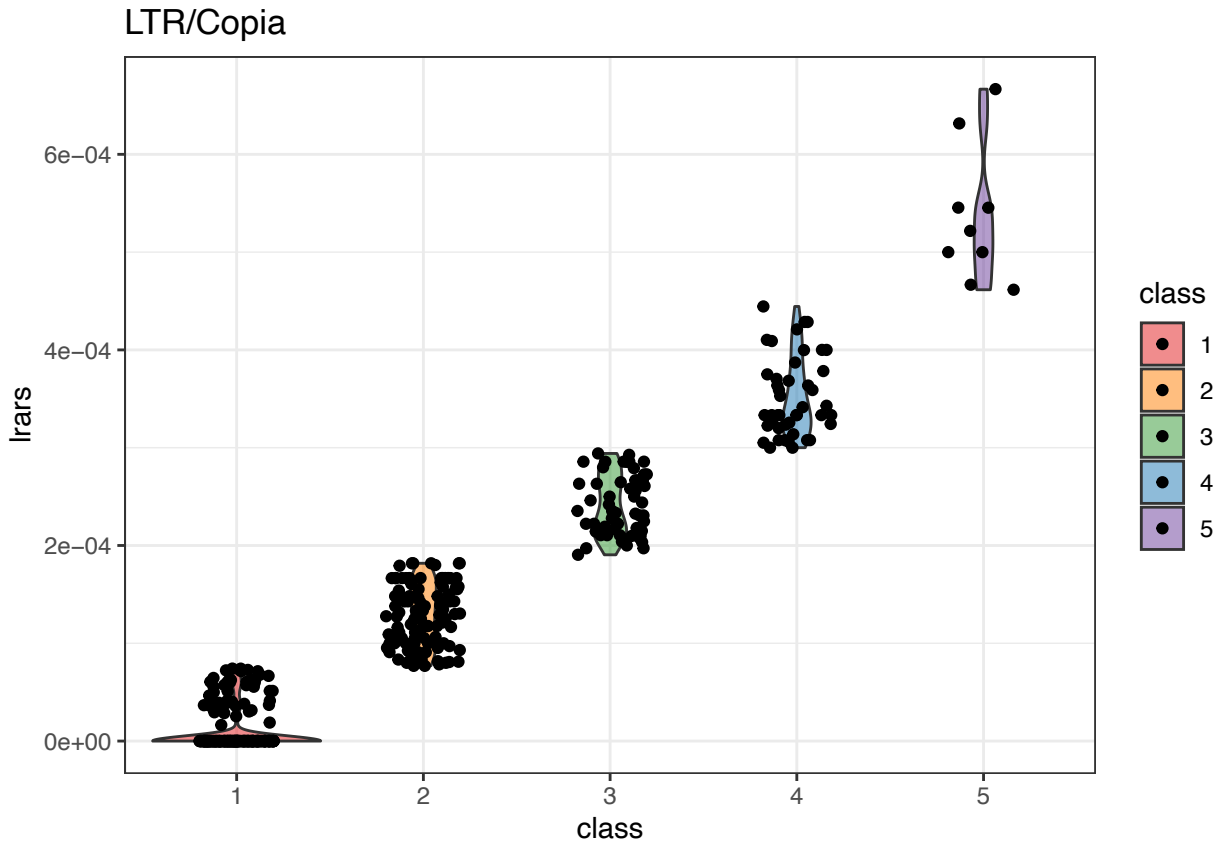
In higher eukaryotes the TE content has been shown to be directly related to the effective population size of the host organism. Lynch M, Conery JS (2003) The origins of genome complexity. Science 302: 1401–1404

```
tabulate_rep_lrar<-table(repbase$class)
tabulate_rep_lrar<-rbind(tabulate_rep_lrar,tabulate_rep_lrar)
for (LRAR in c(1:dim(lrar)[1]))
{
  foo.scaffold<-lrar[LRAR,1]
  foo.start<-lrar[LRAR,2]
  foo.end<-lrar[LRAR,3]
  foo.repbase<-repbase[repbase$query_seq==foo.scaffold&repbase$start>=foo.start&repbase$end<=foo.end,]
  tabulate_rep_lrar<-rbind(tabulate_rep_lrar,table(foo.repbase$class)[colnames(tabulate_rep_lrar)])
}

tabulate_rep_lrar<-tabulate_rep_lrar[-1,]
tabulate_rep_lrar[is.na(tabulate_rep_lrar)]<-0
tabulate_rep_lrar1<-tabulate_rep_lrar[1,]
tabulate_rep_lrar<-tabulate_rep_lrar[-1,]
tabulate_rep_lrar<-cbind(lrar,tabulate_rep_lrar/lrar$Size)
Coding_regions<-NULL
tabulate_regions<-rbind(tabulate_rep_lrar1,tabulate_rep_lrar1)
#tabulate_regions[]<-0
for (LRAR in c(levels(factor(lrar[,1]))))
{
foo.lrar<-lrar[lrar[,1]==LRAR,]
Coding_regions<-rbind(Coding_regions,cbind(Scaf=LRAR,Start=c(0,foo.lrar$End),End=c(foo.lrar$Start,lengt
```

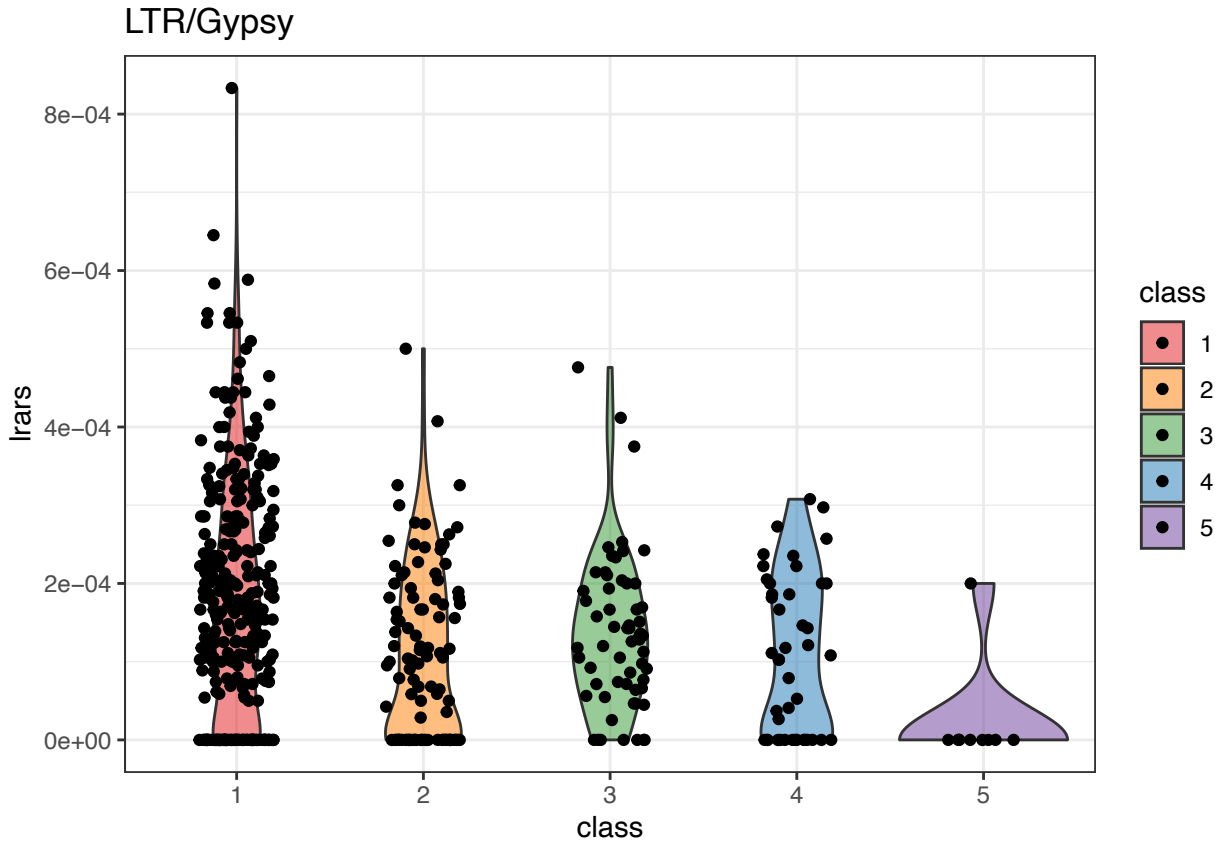
```
}
Coding_regions<-Coding_regions[!(Coding_regions[,2]==0&Coding_regions[,3]==0),]
for (LRAR in c(1:dim(Coding_regions)[1]))
{
  foo.scaffold<-Coding_regions[LRAR,1]
  foo.start<-Coding_regions[LRAR,2]
  foo.end<-Coding_regions[LRAR,3]
  foo.repbase<-repbase[repbase$query_seq==foo.scaffold&repbase$start>=foo.start&repbase$end<=foo.end,]
  tabulate_regions<-rbind(tabulate_regions,table(foo.repbase$class)[colnames(tabulate_regions)])
rm (foo.scaffold)
rm (foo.start)
rm (foo.end)
}
tabulate_regions<-tabulate_regions[-c(1,2),]
tabulate_regions[is.na(tabulate_regions)]<-0
Coding_regions<-data.frame(Coding_regions[,1],matrix(as.numeric(Coding_regions[,-1]),ncol = ncol(Coding_
Coding_regions[,4:13]<-Coding_regions[,4:13]/(Coding_regions[,3]-Coding_regions[,2])
Coding_regions<-Coding_regions[(Coding_regions[,3]-Coding_regions[,2])!=0,]
```

3.2 Cluster Large RIP Affected Regions based on their LTR content

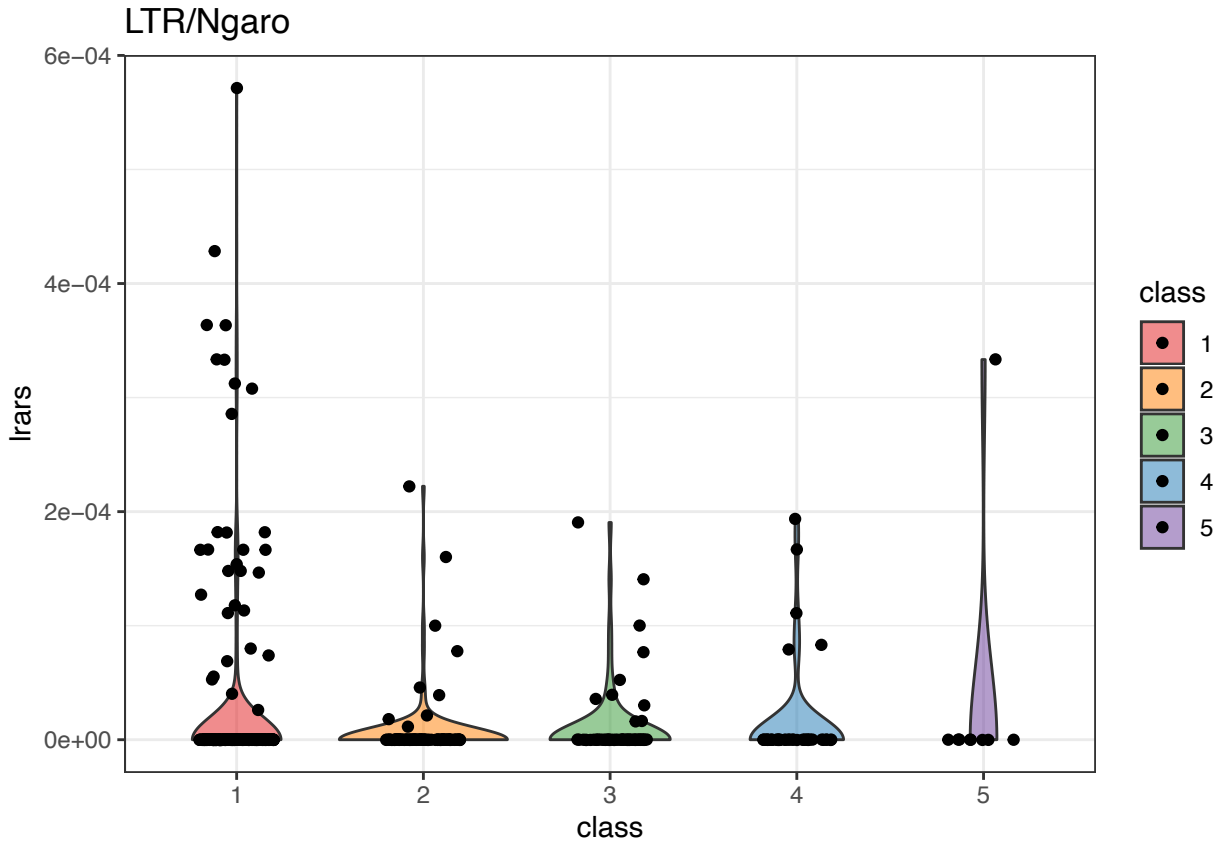
```
grupos<-Mclust(tabulate_rep_lrar[,13],1:5)
ggplot(data.frame(lrars=tabulate_rep_lrar[,13],class=as.factor(grupos$classification)),aes(x=class,y=lra
```




```
ggplot(data.frame(lrars=tabulate_rep_lrar[,14],class=as.factor(grupos$classification)),aes(x=class,y=lra
```

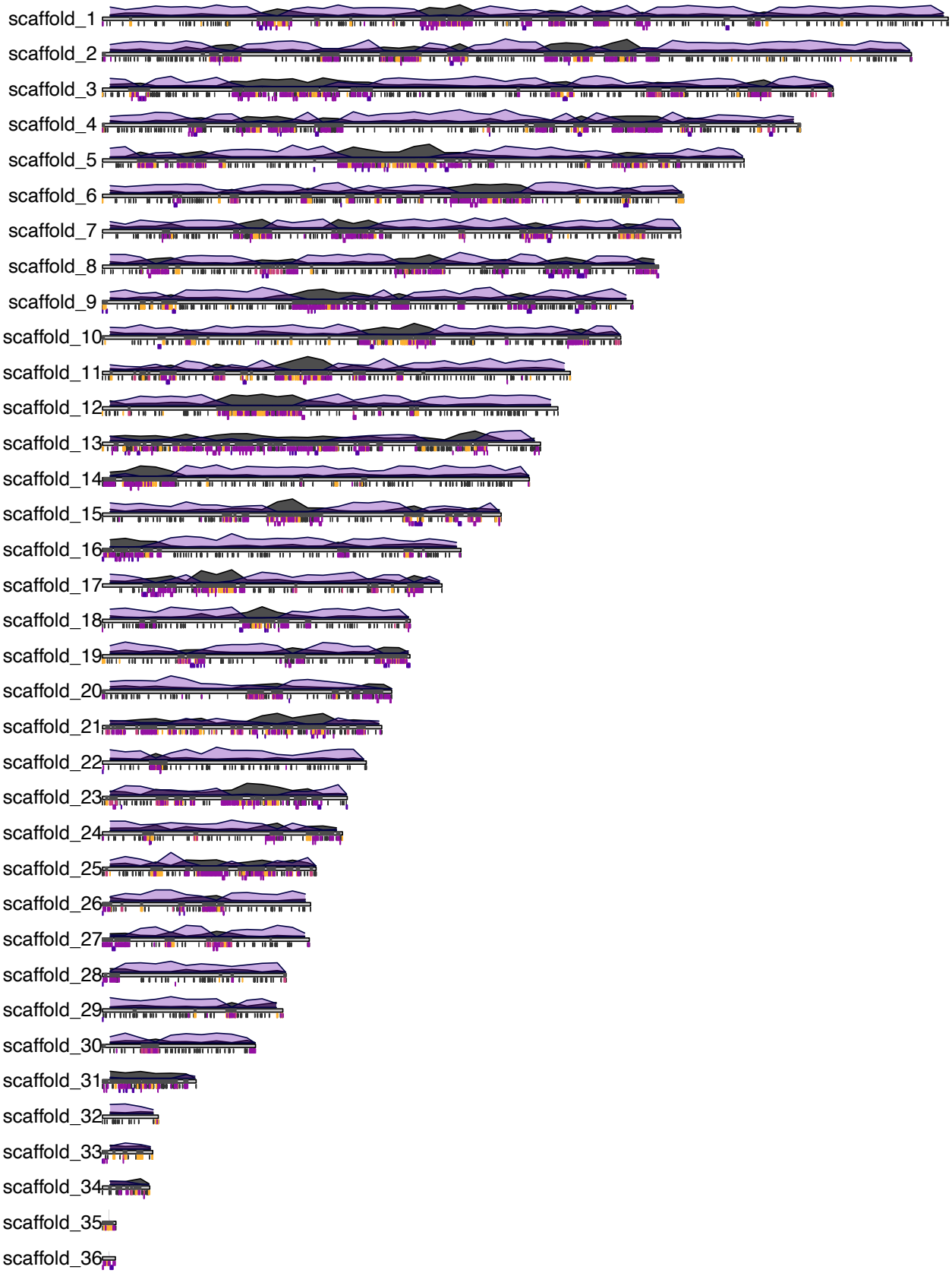



```
ggplot(data.frame(lrars=tabulate_rep_lrar[,15],class=as.factor(grupos$classification)),aes(x=class,y=lra
```

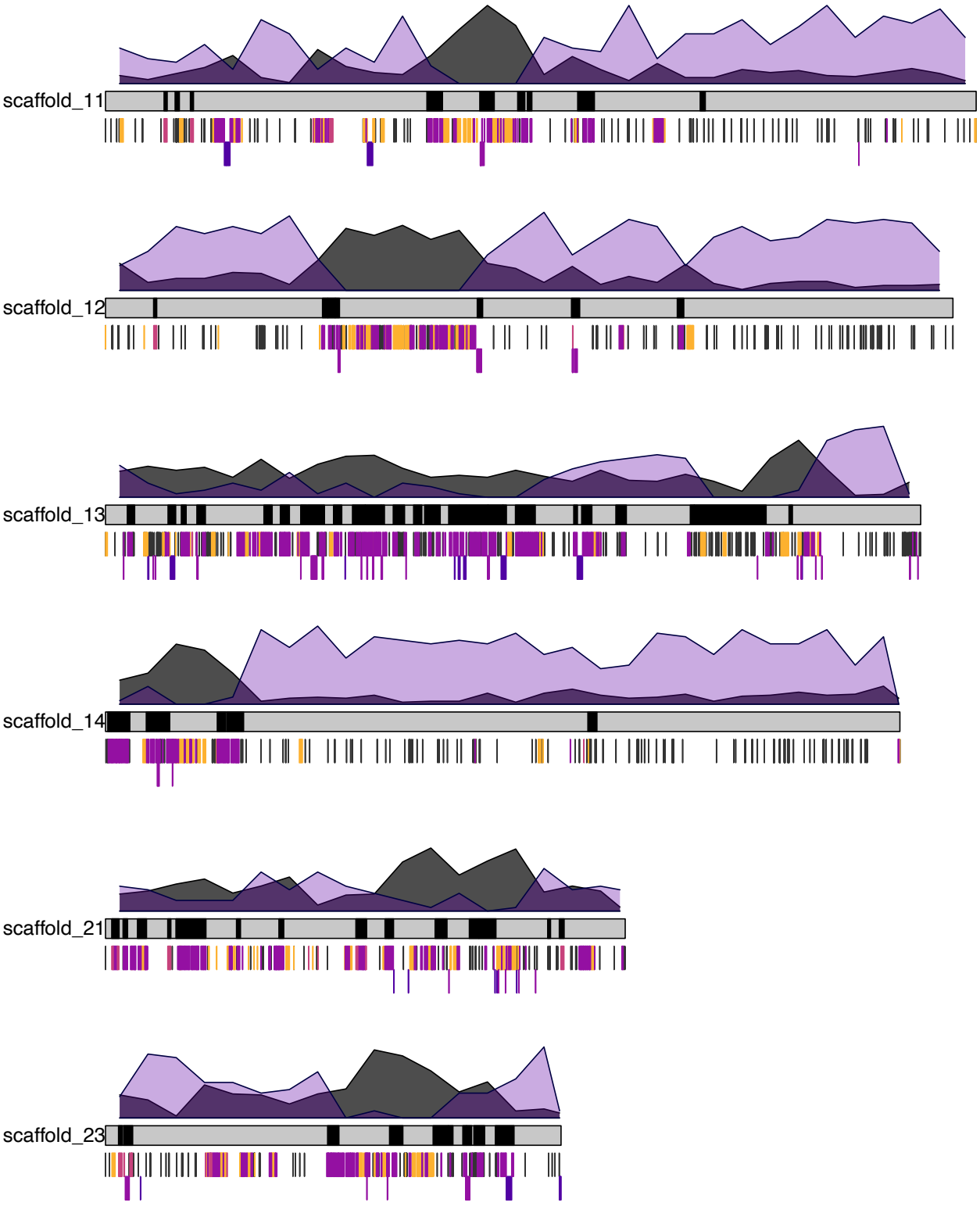


3.3 Graphical description

```
kp <- plotKaryotype(genome = pe.genome, plot.type= 2)
kpPlotRegions(kp, data=paste(lrar$Name,":",lrar$Start+1,"-",lrar$End+1,sep=""),data.panel = 2,
              col=grey(0.3), r0=-0.35, r1=-0.1, avoid.overlapping=FALSE)
dens_Me<-kpPlotDensity(kp, data=mobile_elements,data.panel = 1,col=grey(0.3), r0=0, r1=1,window.size=50
coding_dens<-kpPlotDensity(kp, ranges_genes2, col="#6001A655",data.panel = 1, r0=0, r1=1, window.size =
kpPlotRegions(kp, data=repetitive_sequences,data.panel = 2,col=grey(0.2), r0=0, r1=0.3, avoid.overlappi
kpPlotRegions(kp, data=dna1,data.panel = 2,col=spectrum[5], r0=0.3, r1=0.6, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=ltrcopia,data.panel = 2,col=spectrum[25], r0=0, r1=0.3, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=ltrgypsy,data.panel = 2,col=spectrum[10], r0=0, r1=0.3, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=ltrNgaro,data.panel = 2,col=spectrum[15], r0=0, r1=0.3, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=helitron,data.panel = 2,col=spectrum[10], r0=0.3, r1=0.6, avoid.overlapping=FALS
```


scaffold_1
scaffold_2
scaffold_3
scaffold_4
scaffold_5
scaffold_6
scaffold_7
scaffold_8
scaffold_9
scaffold_10
scaffold_11
scaffold_12
scaffold_13
scaffold_14
scaffold_15
scaffold_16
scaffold_17
scaffold_18
scaffold_19
scaffold_20
scaffold_21
scaffold_22
scaffold_23
scaffold_24
scaffold_25
scaffold_26
scaffold_27
scaffold_28
scaffold_29
scaffold_30
scaffold_31
scaffold_32
scaffold_33
scaffold_34
scaffold_35
scaffold_36

```
kp <- plotKaryotype(genome = pe.genome,plot.type=2,chromosomes=c("scaffold_11","scaffold_12","scaffold_
kpPlotRegions(kp, data=paste(lrar$Name,":",lrar$Start+1,"-",lrar$End+1,sep=""),data.panel = 2,
              col=grey(0:0.5)[grupos$classification], r0=-0.35, r1=-0.1, avoid.overlapping=FALSE)
kpPlotDensity(kp, data=mobile_elements,data.panel = 1,col=grey(0.3), r0=0, r1=1,window.size=50000)
kpPlotDensity(kp, ranges_genes2, col="#6001A655",data.panel = 1, r0=0, r1=1, window.size = 50000)
kpPlotRegions(kp, data=repetitive_sequences,data.panel = 2,col=grey(0.2), r0=0, r1=0.3, avoid.overlappi
kpPlotRegions(kp, data=dna1,data.panel = 2,col=spectrum[5], r0=0.3, r1=0.6, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=ltrcopia,data.panel = 2,col=spectrum[25], r0=0, r1=0.3, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=ltrgypsy,data.panel = 2,col=spectrum[10], r0=0, r1=0.3, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=ltrNgaro,data.panel = 2,col=spectrum[15], r0=0, r1=0.3, avoid.overlapping=FALSE)
kpPlotRegions(kp, data=helitron,data.panel = 2,col=spectrum[10], r0=0.3, r1=0.6, avoid.overlapping=FALSE
```




```
#, border=spectrum[10], r0=1, r1=-0.5, avoid.overlapping=FALSE)
```

3.4 Centromeres

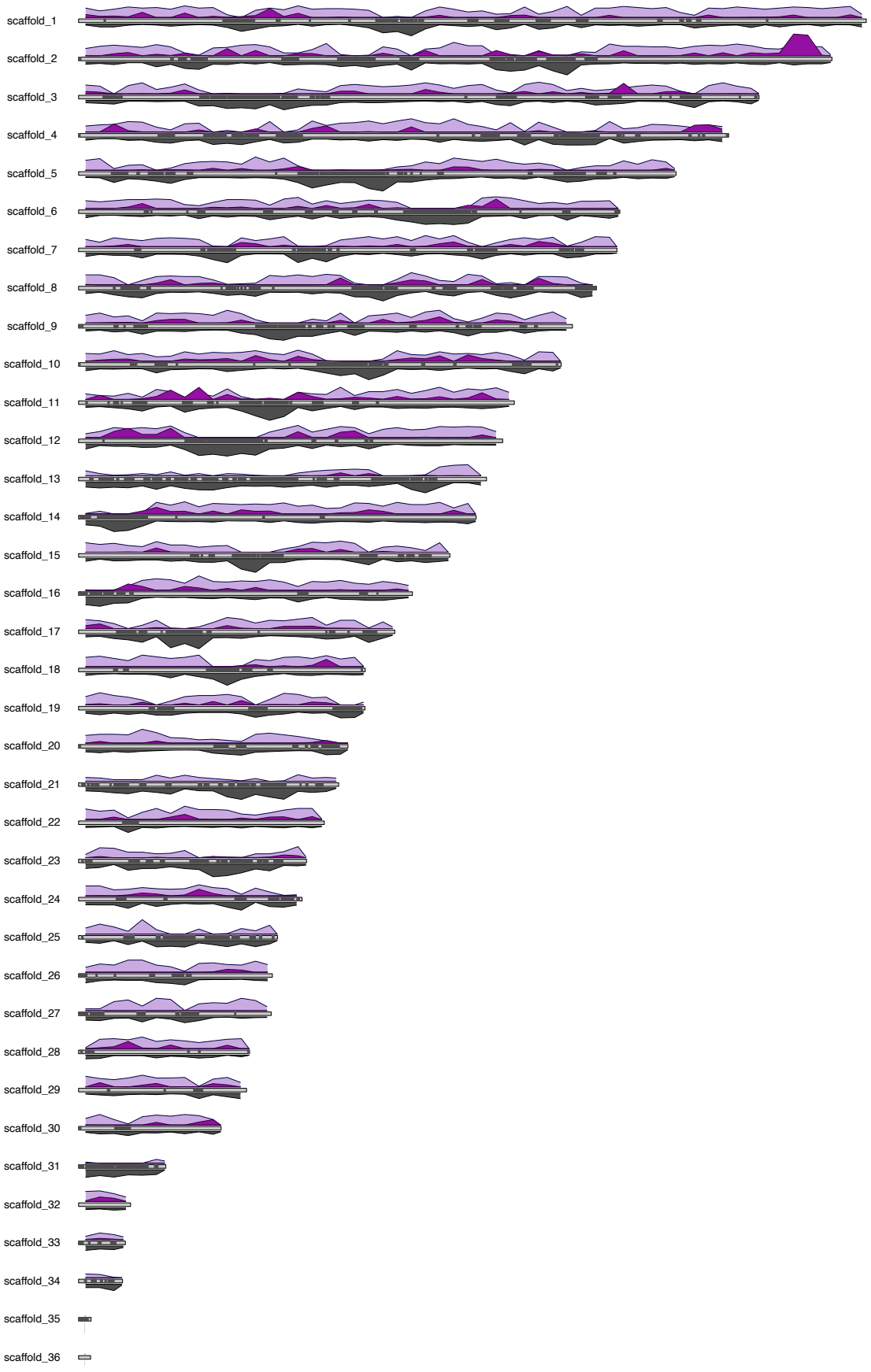
Based on only 16 kb of sequence, it was concluded that Neurospora centromeres are composed of degenerate transposons, mostly retrotransposons, and simple sequence repeats. The degenerate nature of the transposons is due to the action of a premeiotic process called "Repeat-Induced Point mutation" (RIP), which through an unknown mechanism recognizes repeated DNA and mutates both copies, yielding numerous C:T and G:A transition mutations (Cambareri et al., 1989, Selker, 1990). RIP will continue in successive sexual cycles until sequence identity between two copies decreases below ~85% but it will begin again if such regions become re-duplicated (Cambareri et al., 1991). Presumably these cycles of duplication and mutagenesis can continue until no Cs remain. Combined with potential gene conversion or recombination events, RIP appears to provide an interesting mechanism for diversification of centromeric DNA in many filamentous fungi.

3.5 Loss of synteny across the genome

```
#meko<-"| |"
#lost_out<-NULL
#for (FILE in list.files("/Users/Fernando/Desktop/01_paper_sanger_drive/06_Synteny/all.html/")[-1])
#{
#salida<-NULL
#prueba<-XML::readHTMLTable(paste("/Users/Fernando/Desktop/01_paper_sanger_drive/06_Synteny/all.html/",
#foo<-grep("Pyrenodesmiaerodens_",prueba[[1]])
#  if(length(foo)!=0)
#    {
#    prueba<-prueba[[1]][,c(1,2,foo)]
#    for (i in 3:dim(prueba)[2])
#      {
#      salida<-rbind(salida,as.matrix(prueba[min(grep("Pyrenodesmiaerodens",prueba[,i])):max(grep("Pyre
#      }
#    salida[,3]<-gsub("-T1","",salida[,3])
#   salida<-cbind(salida,genes2[salida[,3],c("scaffold","start","end")])
#    for (i in 1:dim(salida)[1])
#      {
#      if (salida[i,3]==meko)
#        {
#        salida[i,3]<-paste(salida[i-1,3],"lost",sep = "-")
#        scaff_pre<-salida[i-1,4]
#        j<-i
#        while(is.na(salida[j,4]))
#          {
#          j=j+1
#          }
#        scaff_post<-salida[j,4]
#        if(!(is.na(scaff_pre)|is.na(scaff_post)))
#          {
#          if(scaff_pre==scaff_post)
#          {
#          salida[i,4]<-scaff_pre
#          salida[i,5]<-as.integer(mean(c(as.numeric(salida[j,5]),as.numeric(salida[i-1,5]))))
#          salida[i,6]<-as.numeric(salida[i,5])+100
#          }
#        }
#        }
#   }
#    salida<-salida[sapply(strsplit(salida[,3],"-"),`[`,2)=="lost"&!is.na(sapply(strsplit(salida[,3],"-
```

```
#    if(dim(salida)[1]>0)
#      {
#      lost_out<-rbind(lost_out,cbind(FILE,unname(salida)))
#    }
#  }
#}
#colnames(lost_out)<-c("file","depth","ref","name","scaffold","start","end")
```

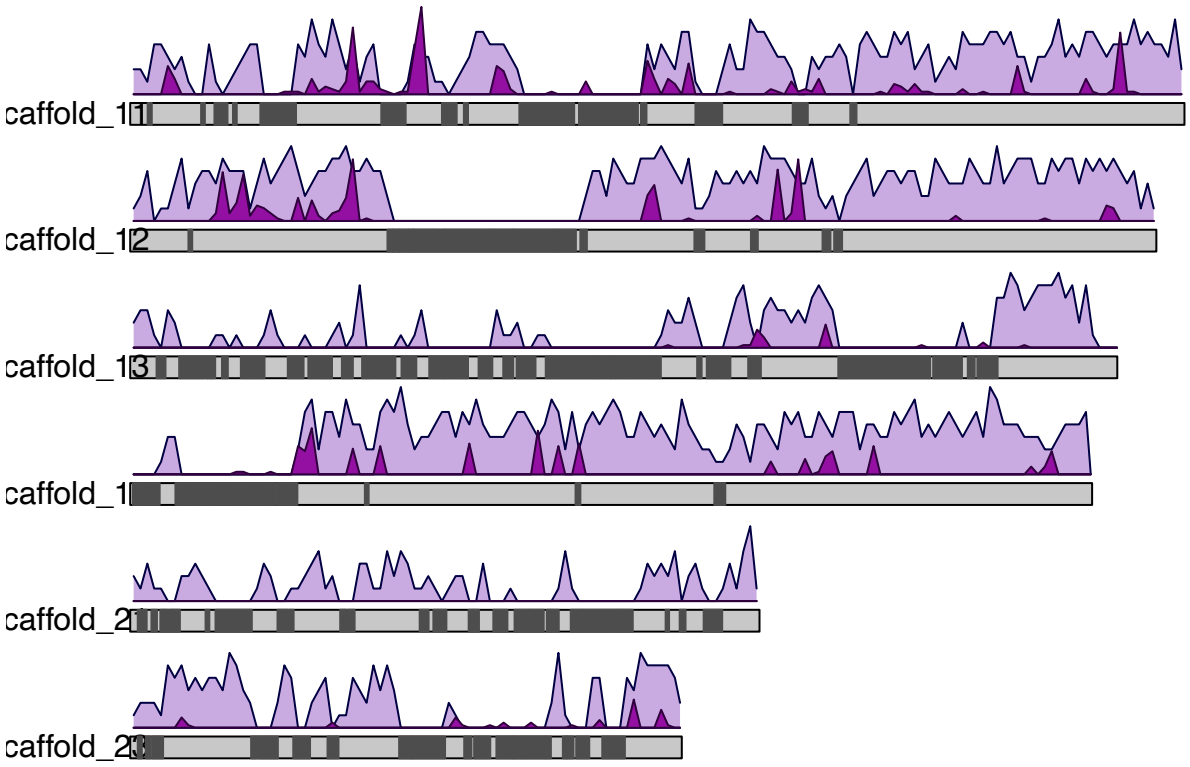
```
#lost_out<-lost_out[grep("Pyrenodesmiaerodens",lost_out$name),]
#save(lost_out,file="/Users/Fernando/Desktop/01_paper_sanger_drive/synteny.Rdata")
load("/Users/Fernando/Desktop/01_paper_sanger_drive/synteny.Rdata")
lost_genes<-melt(strsplit(ogs$V5,", "))
rownames(lost_genes)<-lost_genes[,1]
foo<-table(lost_genes$L1)
lost_genes<-lost_genes[lost_genes$L1%in%names(foo)[foo>=10],1]
lost_genes<-lost_out[lost_out$ref%in%lost_genes,]



lost_genes<-makeGRangesFromDataFrame(lost_genes,
                         keep.extra.columns=FALSE,
                         ignore.strand=FALSE,
                         seqinfo=NULL,
                         seqnames.field="scaffold",
                         start.field="start",
                         end.field="end",
                         strand.field="strand",
                         starts.in.df.are.0based=FALSE)
kp <- plotKaryotype(genome = pe.genome, plot.type= 2)
kpPlotRegions(kp, data=paste(lrar$Name,":",lrar$Start+1,"-",lrar$End+1,sep=""),data.panel = 2,
              col=grey(0.3), r0=-0.35, r1=-0.1, avoid.overlapping=FALSE)
fer<-kpPlotDensity(kp, ranges_genes2, col="#6001A655",data.panel = 1, r0=0, r1=1, window.size = 50000)
cobol<-kpPlotDensity(kp,lost_genes, col=spectrum[10],data.panel = 1, r0=0, r1=1.5, window.size = 50000)
trans<-kpPlotDensity(kp, data=mobile_elements,data.panel = 2,col=grey(0.3), r0=0, r1=1,window.size=5000
```


scaffold_1
scaffold_2
scaffold_3
scaffold_4
scaffold_5
scaffold_6
scaffold_7
scaffold_8
scaffold_9
scaffold_10
scaffold_11
scaffold_12
scaffold_13
scaffold_14
scaffold_15
scaffold_16
scaffold_17
scaffold_18
scaffold_19
scaffold_20
scaffold_21
scaffold_22
scaffold_23
scaffold_24
scaffold_25
scaffold_26
scaffold_27
scaffold_28
scaffold_29
scaffold_30
scaffold_31
scaffold_32
scaffold_33
scaffold_34
scaffold_35
scaffold_36

3.6 Detail of disrupted scaffolds

```
detail <- plotKaryotype(genome = pe.genome,plot.type=1,chromosomes=c("scaffold_11","scaffold_12","scaff
kpPlotRegions(detail, data=paste(lrar$Name,":",lrar$Start+1,"-",lrar$End+1,sep=""),data.panel = 2,
              col=grey(0.3), r0=-0.35, r1=-0.1, avoid.overlapping=FALSE)
#kpPlotDensity(kp,lost_genes, col=spectrum[10],data.panel = 1, r0=0, r1=1, window.size = 50000)
kpPlotDensity(detail, ranges_genes2, col="#6001A655",data.panel = 1, r0=0, r1=1, window.size = 10000)
kpPlotDensity(detail,lost_genes, col=spectrum[10],data.panel = 1, r0=0, r1=1, window.size = 10000)
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_21
##   - in 'y': scaffold_1, scaffold_10, scaffold_15, scaffold_16, scaffold_17, scaffold_18, scaffold_19
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```



3.6.1 Influence of the density of transposable elements and overall coding density on loss of synteny

```
anova(glm(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density*trans$lates
```

```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: cobol$latest.plot$computed.values$density
##
## Terms added sequentially (first to last)
##
##
##                                                                                    Df
```

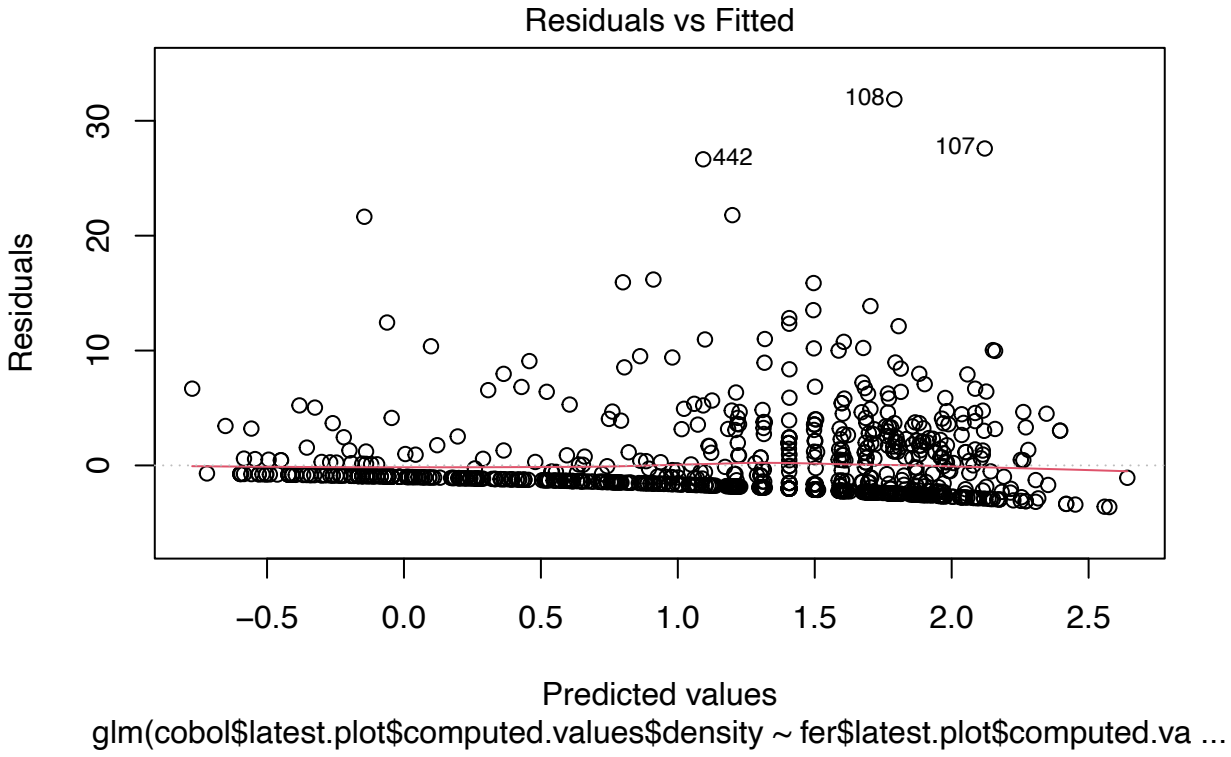
```
## NULL
## fer$latest.plot$computed.values$density                                          1
## trans$latest.plot$computed.values$density                                        1
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density  1
##                                                                                Deviance
## NULL
## fer$latest.plot$computed.values$density                                       1363.23
## trans$latest.plot$computed.values$density                                        5.65
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density   12.48
##                                                                              Resid. Df
## NULL                                                                               857
## fer$latest.plot$computed.values$density                                            856
## trans$latest.plot$computed.values$density                                          855
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density      854
##                                                                              Resid. Dev
## NULL                                                                            9003.6
## fer$latest.plot$computed.values$density                                         7640.4
## trans$latest.plot$computed.values$density                                       7634.8
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density    7622.3
```

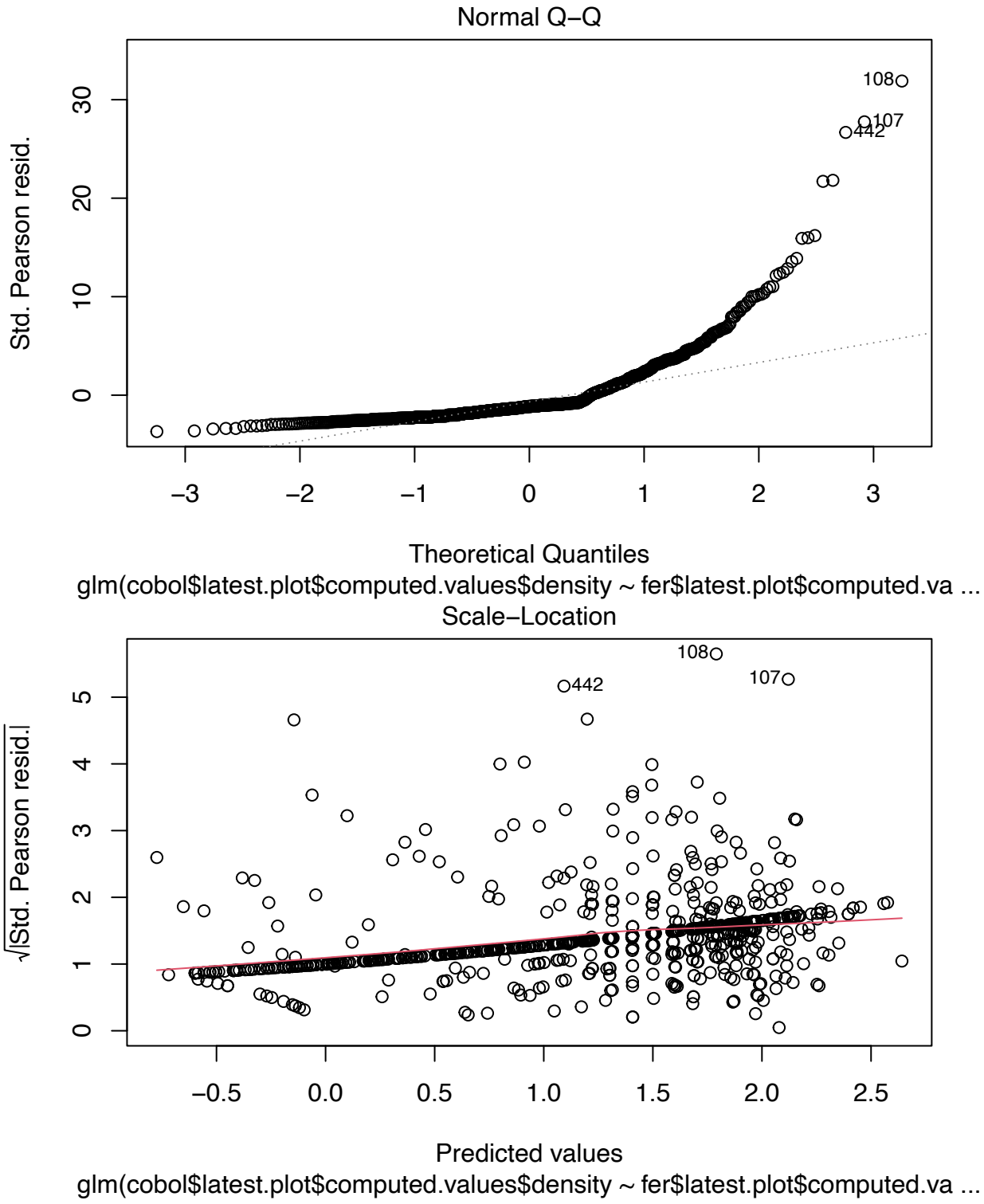
```
summary(glm(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density*trans$late
```

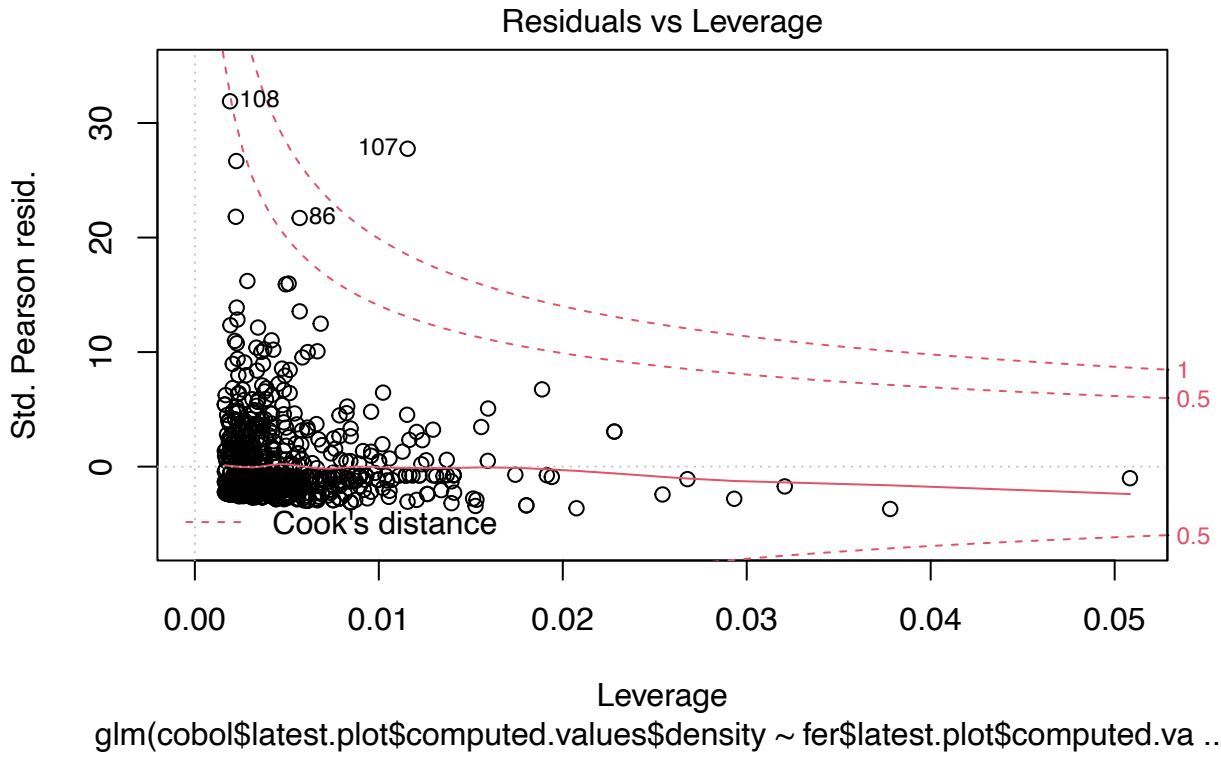
```
##
## Call:
## glm(formula = cobol$latest.plot$computed.values$density ~ fer$latest.plot$computed.values$density *
##     trans$latest.plot$computed.values$density, family = poisson)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.126  -2.854  -1.583   0.641  16.957
##
## Coefficients:
##                                                                                Estimate
## (Intercept)                                                                   0.2801911
## fer$latest.plot$computed.values$density                                       0.0867651
## trans$latest.plot$computed.values$density                                    -0.0135070
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density  0.0010261
##                                                                              Std. Error
## (Intercept)                                                                   0.0899817
## fer$latest.plot$computed.values$density                                       0.0053204
## trans$latest.plot$computed.values$density                                     0.0034140
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density  0.0002946
##                                                                              z value
## (Intercept)                                                                    3.114
## fer$latest.plot$computed.values$density                                       16.308
## trans$latest.plot$computed.values$density                                     -3.956
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density   3.483
##                                                                              Pr(>|z|)
## (Intercept)                                                                  0.001847
## fer$latest.plot$computed.values$density                                       < 2e-16
## trans$latest.plot$computed.values$density                                    7.61e-05
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density 0.000497
##
## (Intercept)                                                                  **
## fer$latest.plot$computed.values$density                                      ***
```

```
## trans$latest.plot$computed.values$density                                    ***
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 9003.6  on 857  degrees of freedom
## Residual deviance: 7622.3  on 854  degrees of freedom
## AIC: 8956.4
##
## Number of Fisher Scoring iterations: 6
```

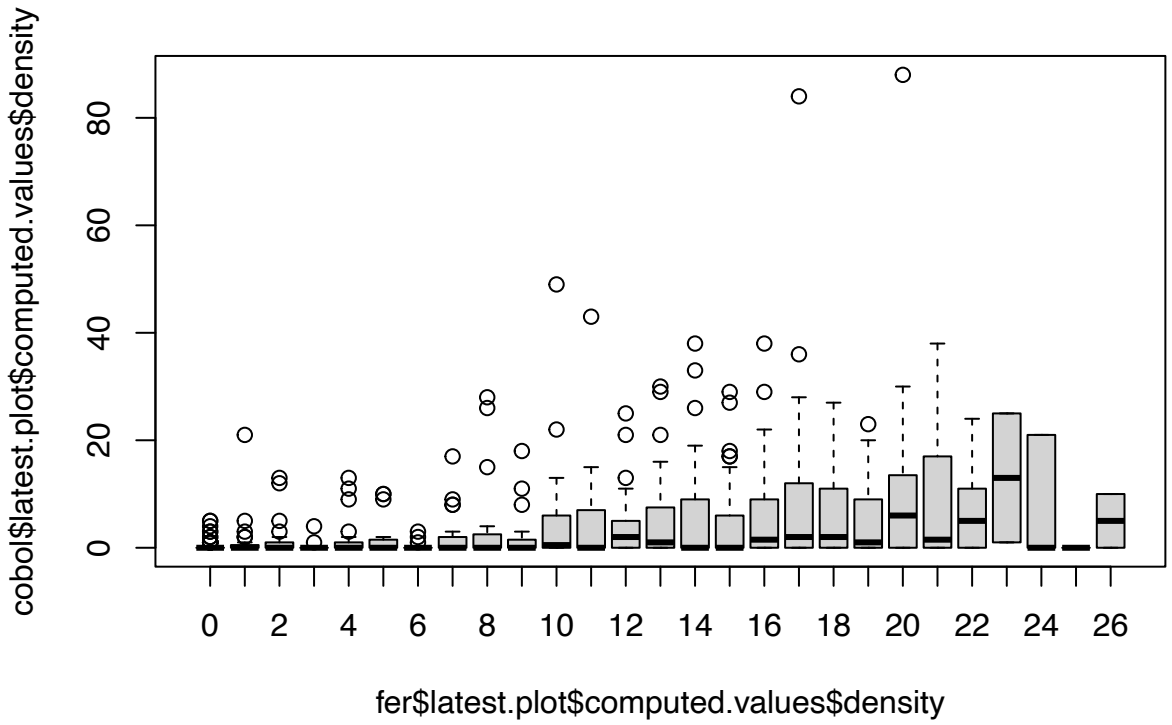
```
plot(glm(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density*trans$latest
```



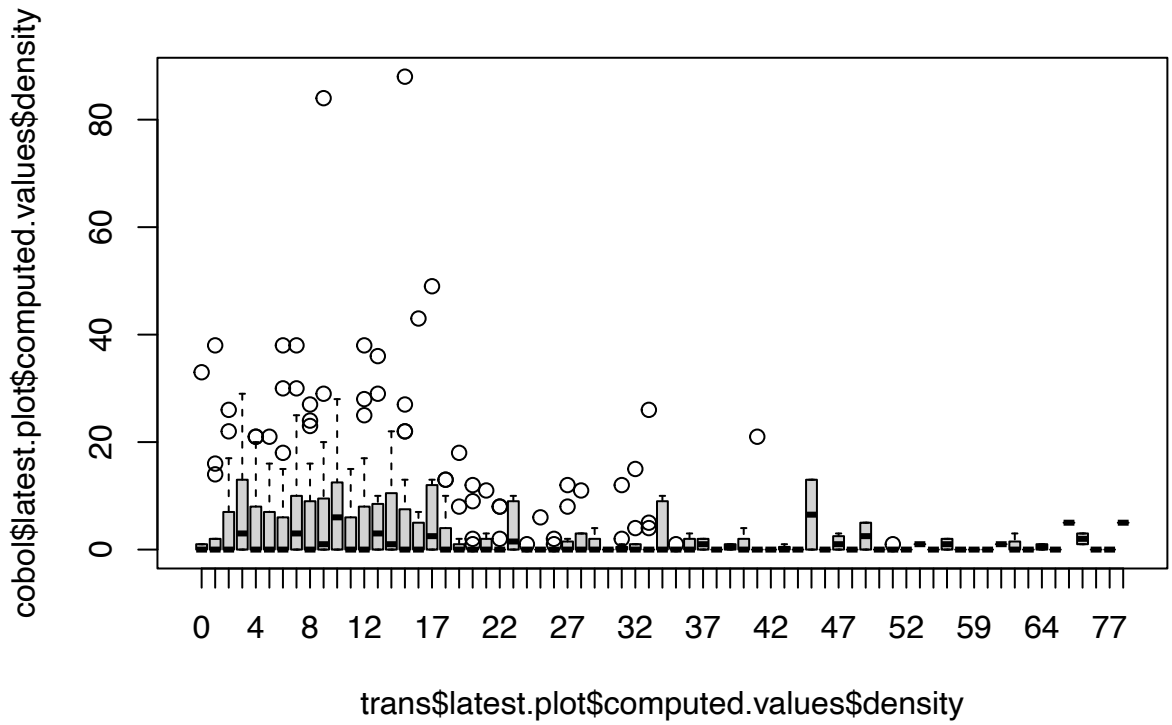

Normal Q–Q
Std. Pearson resid.
108
107
442
0
10
20
30
–3
–2
–1
0
1
2
3
Theoretical Quantiles
glm(cobol$latest.plot$computed.values$density ~ fer$latest.plot$computed.va ...
Scale–Location
√|Std. Pearson resid.|
108
107
442
0
1
2
3
4
5
–0.5
0.0
0.5
1.0
1.5
2.0
2.5
Predicted values
glm(cobol$latest.plot$computed.values$density ~ fer$latest.plot$computed.va ...




```
boxplot(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density)
```



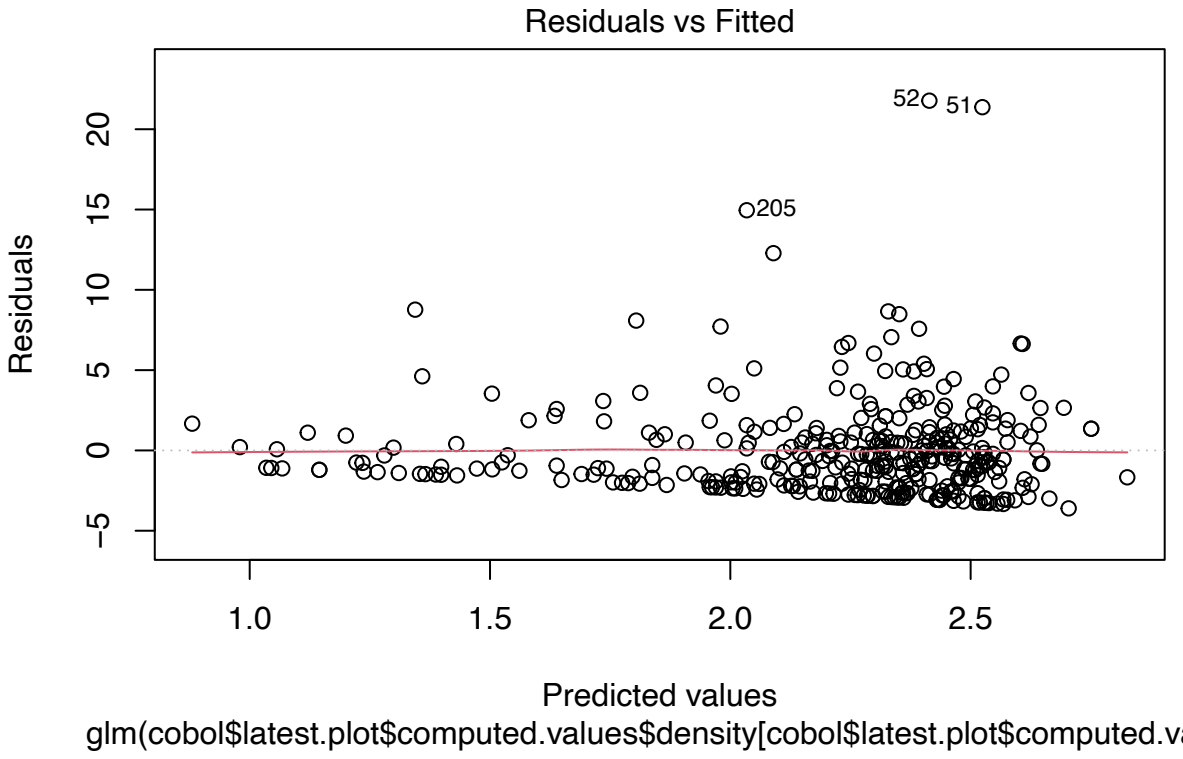

```
boxplot(cobol$latest.plot$computed.values$density~trans$latest.plot$computed.values$density)
```

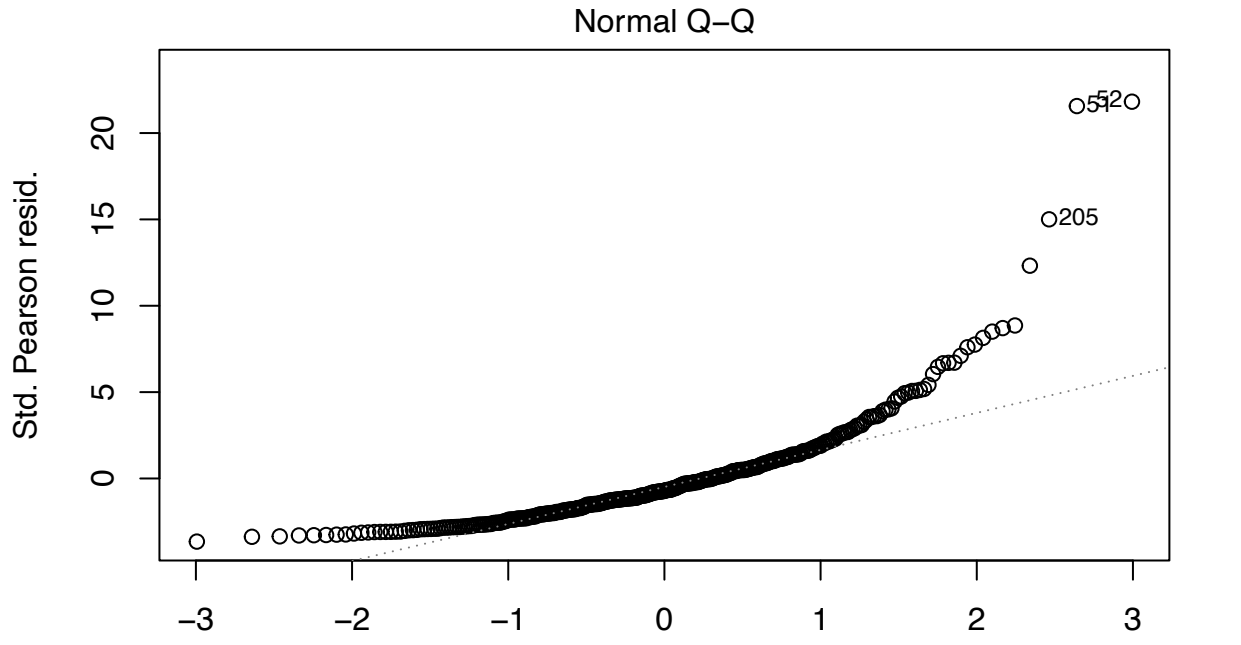


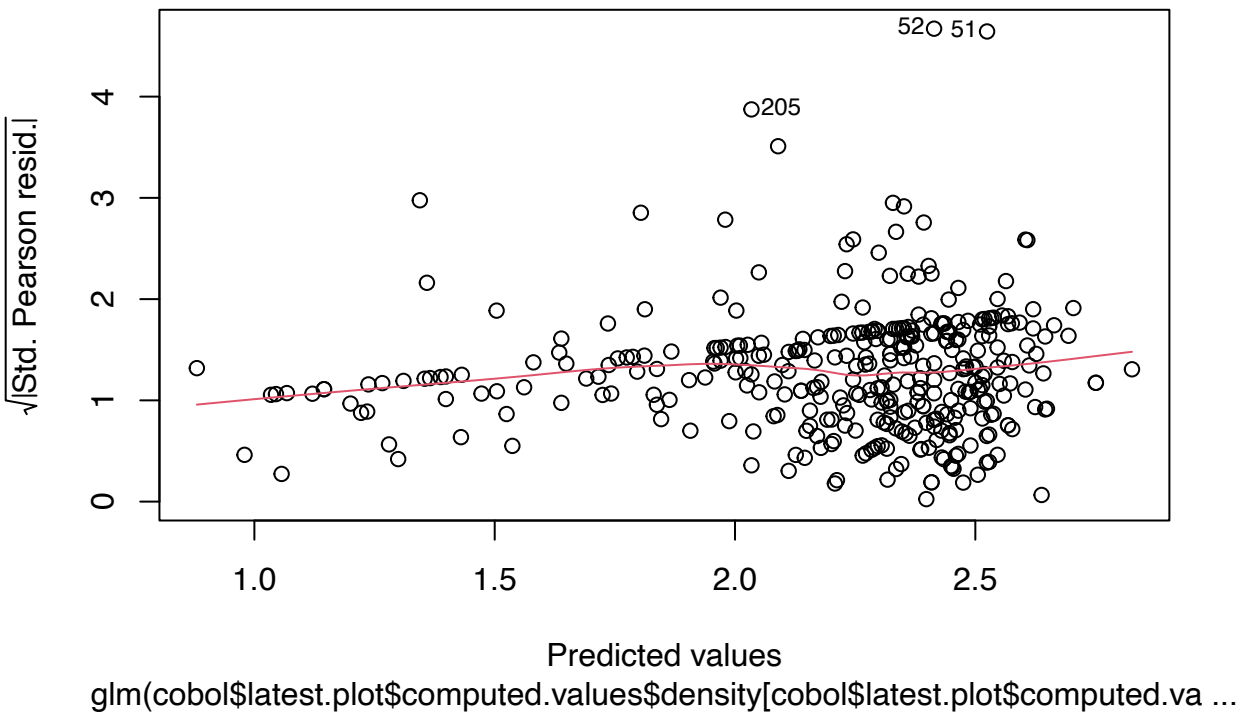
###

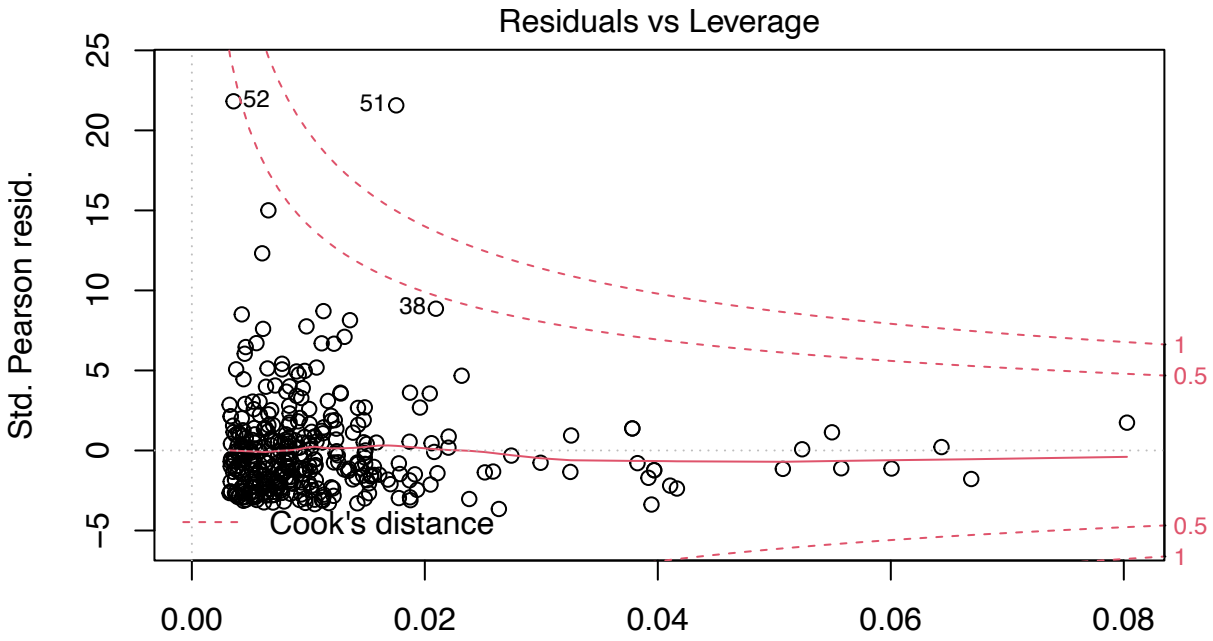
Taking regions with null gene density to implicitely eliminate the effect of centromeric regions and LRARs.

```
plot(glm(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~fer$la
```

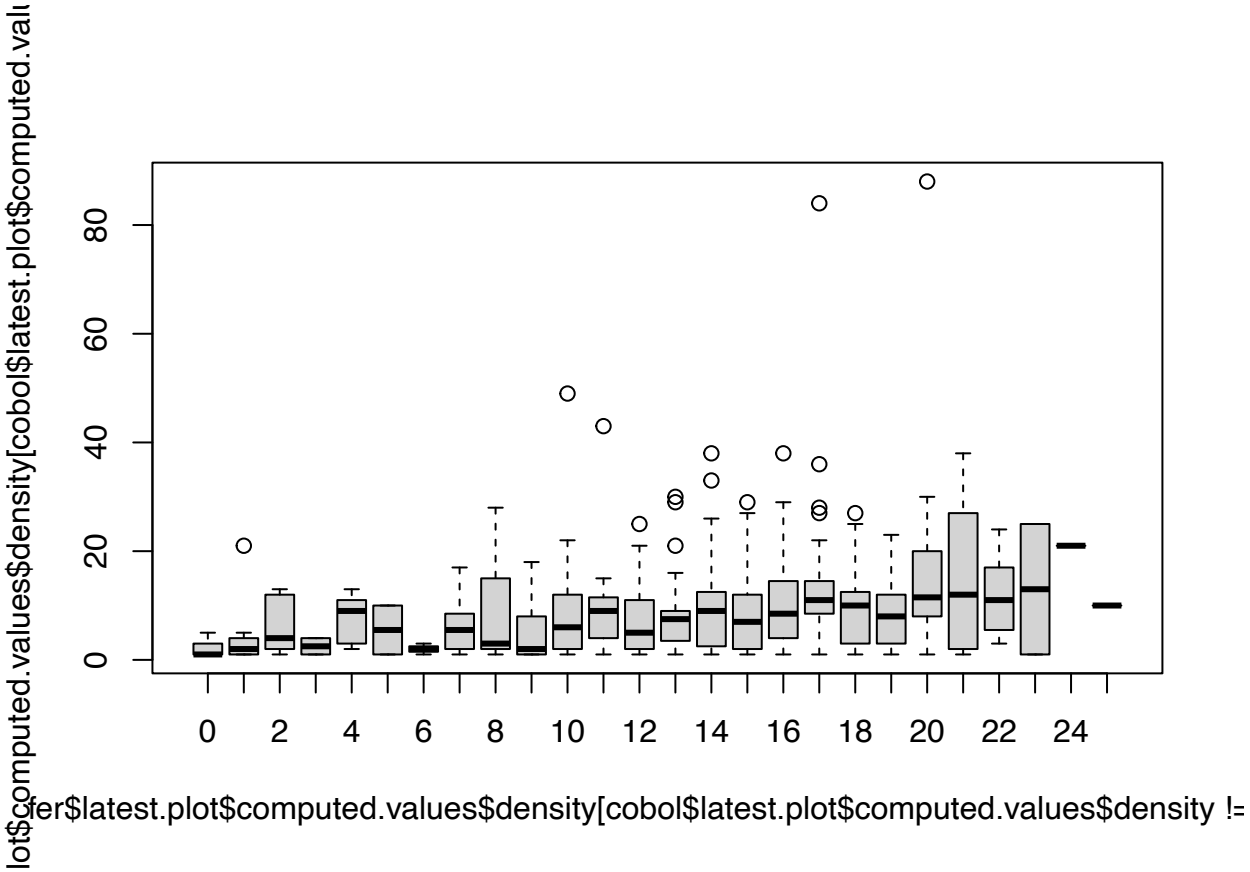



Normal Q–Q
Std. Pearson resid.
0
5
10
15
20
52
51
205
–3
–2
–1
0
1
2
3
Theoretical Quantiles
glm(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.va ...

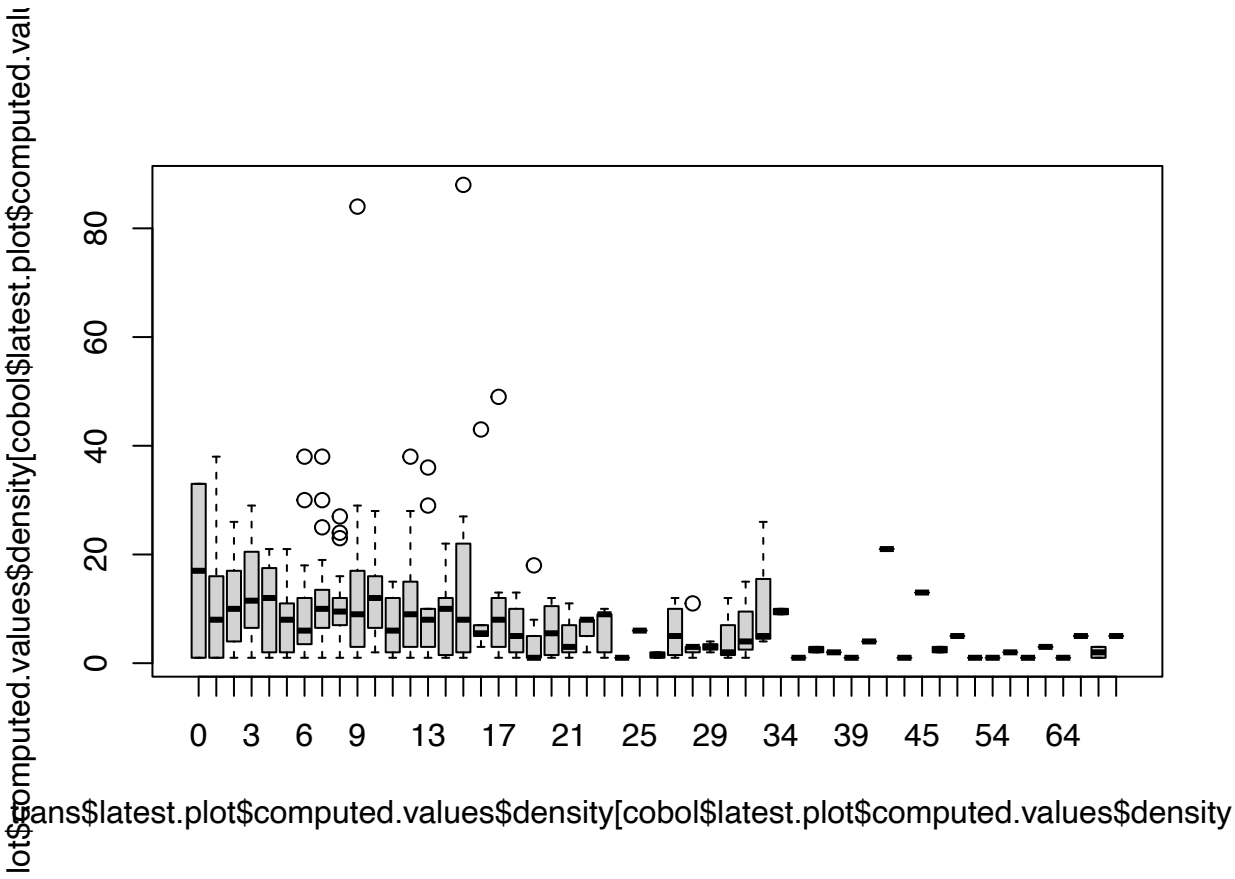

Scale–Location
√|Std. Pearson resid.|
0
1
2
3
4
52
51
205
1.0
1.5
2.0
2.5
Predicted values
glm(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.va ...




```
boxplot(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~fer$late
```



```
boxplot(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~trans$la
```



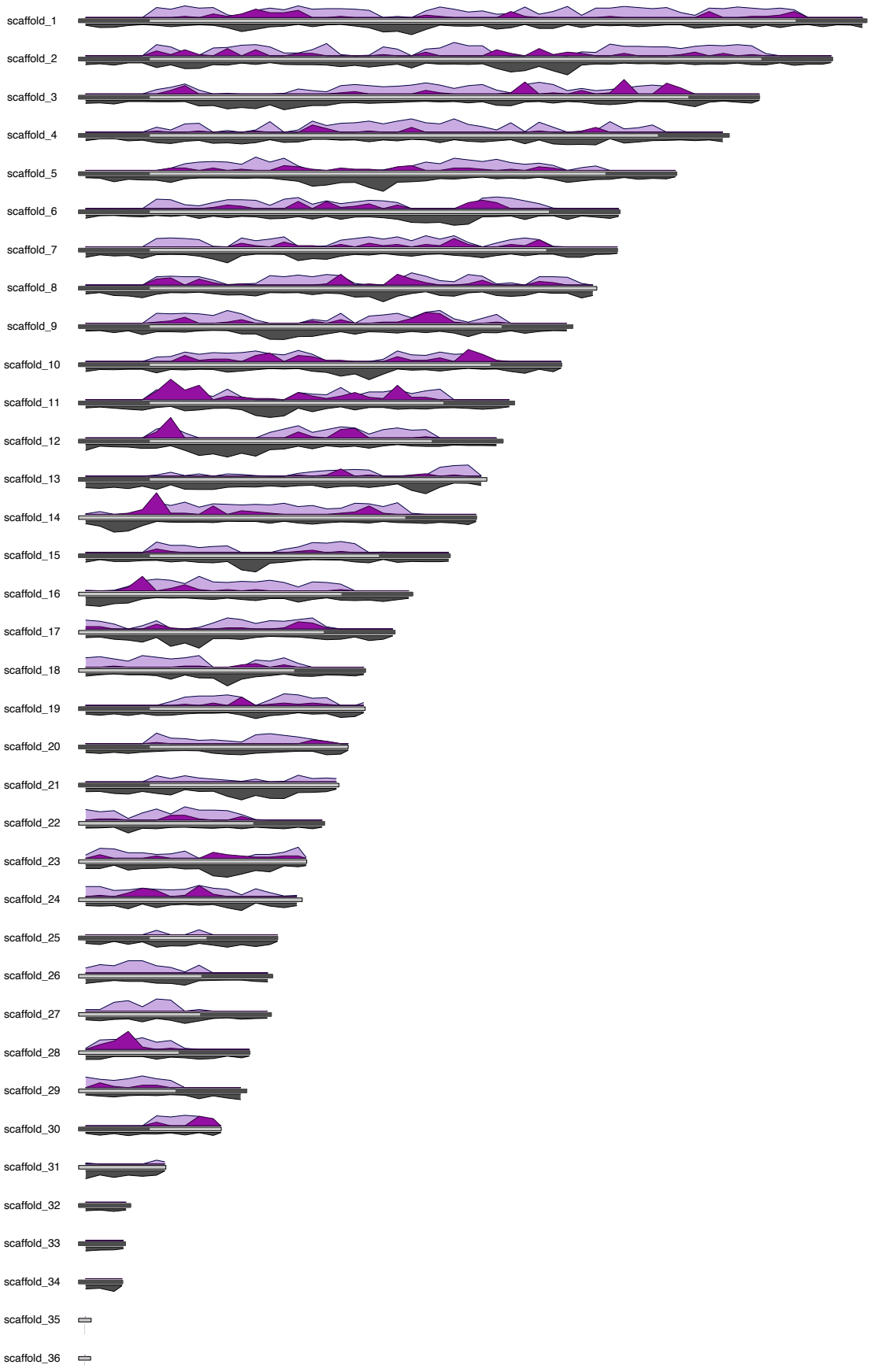
3.6.2 Explicitely eliminating Subtelomeric regions

To eliminate subtelomeric regions we substracted 0.30Mbp from each side of the scaffolds where Telomeric motif repeats have been identified.

```
telomere_locs<-rbind(cbind(paste("scaffold",c(1:13,15,19:21,25,30,34),sep="_"),0,250000),cbind(paste("s
colnames(telomere_locs)<-c("scaffold","start","end")
telomere_locs<-data.frame(scaffold=telomere_locs[,1],start=as.numeric(telomere_locs[,2]),end=as.numeric
telomere_locs$start[telomere_locs$start<0]<-0
telomere_locs$end[telomere_locs$end>sapply(pe_genome[telomere_locs$scaffold],length)]<-sapply(pe_genome
telomere_locs<-makeGRangesFromDataFrame(telomere_locs,
                         keep.extra.columns=FALSE,
                         ignore.strand=FALSE,
                         seqinfo=NULL,
                         seqnames.field="scaffold",
                         start.field="start",
                         end.field="end",
                         strand.field="strand",
                         starts.in.df.are.0based=FALSE)
kp <- plotKaryotype(genome = pe.genome, plot.type= 2)
kpPlotRegions(kp, telomere_locs,data.panel = 2,col=grey(0.3), r0=-0.35, r1=-0.1, avoid.overlapping=FALSE
fer<-kpPlotDensity(kp, GenomicRanges::setdiff(ranges_genes2,telomere_locs), col="#6001A655",data.panel =
cobol<-kpPlotDensity(kp,GenomicRanges::setdiff(lost_genes,telomere_locs), col=spectrum[10],data.panel =

## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
```

```
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_21, scaffold_27, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
trans<-kpPlotDensity(kp, GenomicRanges::setdiff(mobile_elements,telomere_locs),data.panel = 2,col=grey(
```


scaffold_1
scaffold_2
scaffold_3
scaffold_4
scaffold_5
scaffold_6
scaffold_7
scaffold_8
scaffold_9
scaffold_10
scaffold_11
scaffold_12
scaffold_13
scaffold_14
scaffold_15
scaffold_16
scaffold_17
scaffold_18
scaffold_19
scaffold_20
scaffold_21
scaffold_22
scaffold_23
scaffold_24
scaffold_25
scaffold_26
scaffold_27
scaffold_28
scaffold_29
scaffold_30
scaffold_31
scaffold_32
scaffold_33
scaffold_34
scaffold_35
scaffold_36

3.6.3 Influence of the density of transposable elements and overall coding density on loss of synteny

```
summary(glm(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density*trans$late
```

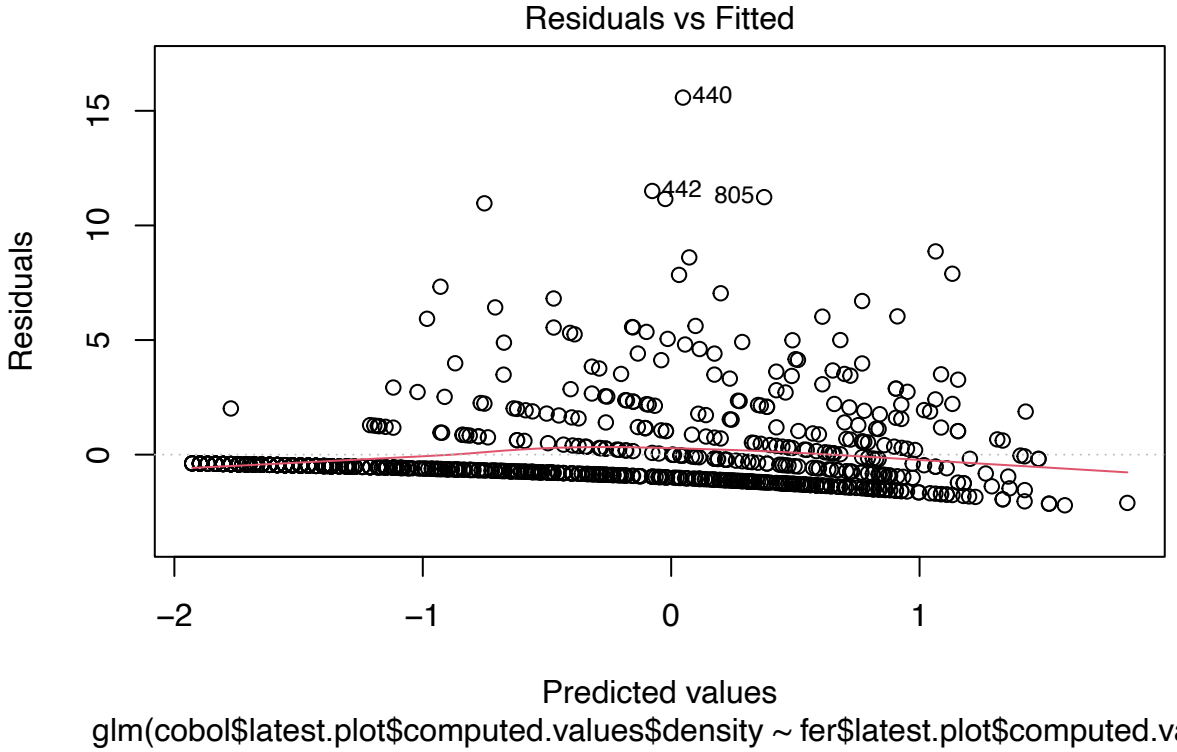
```
##
## Call:
## glm(formula = cobol$latest.plot$computed.values$density ~ fer$latest.plot$computed.values$density *
##     trans$latest.plot$computed.values$density, family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.1237  -1.1386  -0.6811  -0.1890   7.9254
##
## Coefficients:
##                                                                                 Estimate
## (Intercept)                                                                    -1.9283601
## fer$latest.plot$computed.values$density                                         0.1393210
## trans$latest.plot$computed.values$density                                       0.0311400
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density -0.0004145
##                                                                                Std. Error
## (Intercept)                                                                     0.1404064
## fer$latest.plot$computed.values$density                                         0.0085091
## trans$latest.plot$computed.values$density                                       0.0052655
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density  0.0005365
##                                                                                z value
## (Intercept)                                                                    -13.734
## fer$latest.plot$computed.values$density                                         16.373
## trans$latest.plot$computed.values$density                                        5.914
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density  -0.773
##                                                                                Pr(>|z|)
## (Intercept)                                                                     < 2e-16
## fer$latest.plot$computed.values$density                                         < 2e-16
## trans$latest.plot$computed.values$density                                      3.34e-09
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density     0.44
##
## (Intercept)                                                                    ***
## fer$latest.plot$computed.values$density                                        ***
## trans$latest.plot$computed.values$density                                      ***
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 2565.8  on 857  degrees of freedom
## Residual deviance: 1916.8  on 854  degrees of freedom
## AIC: 2685.2
##
## Number of Fisher Scoring iterations: 6
```

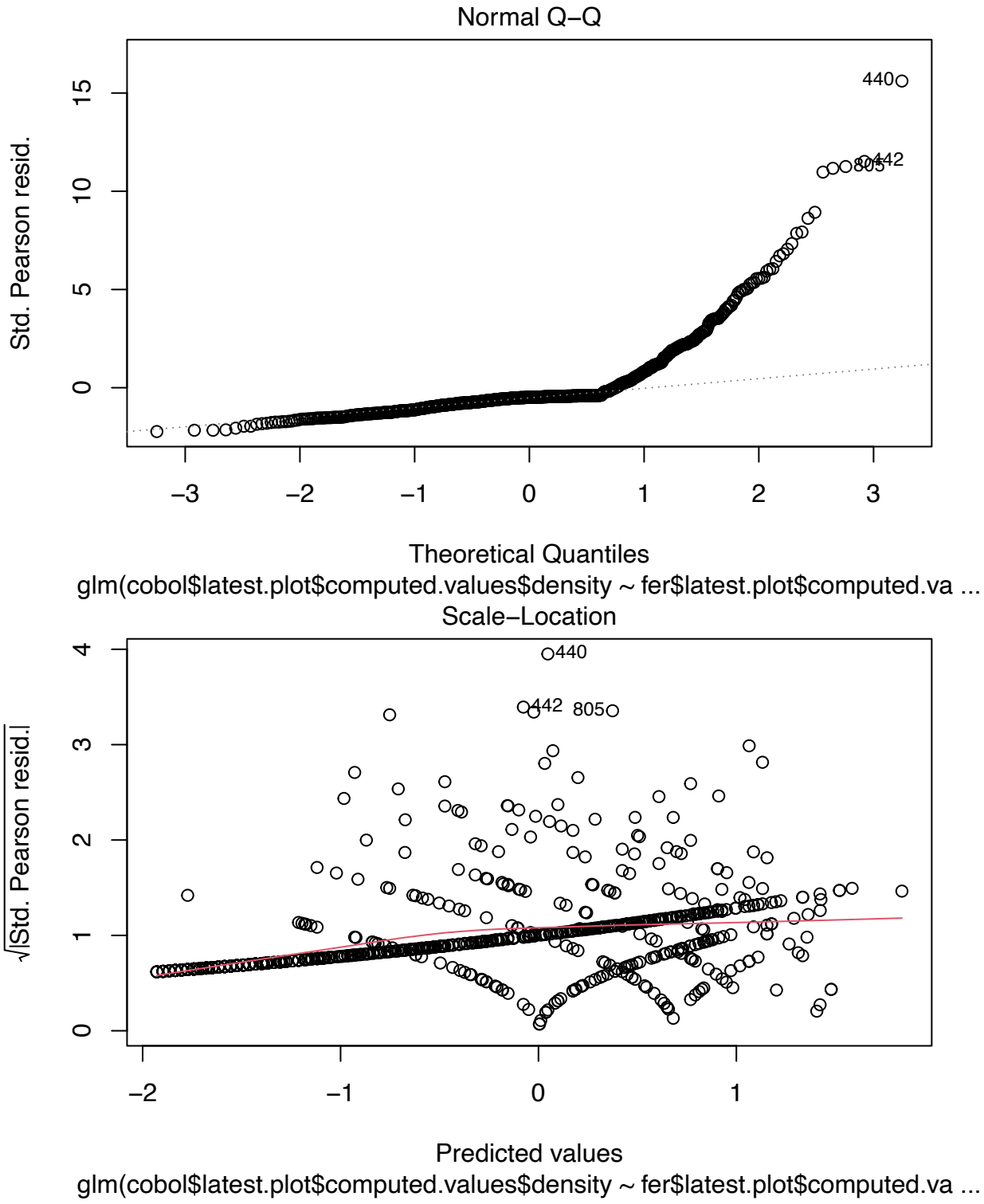
```
anova(glm(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density*trans$lates
```

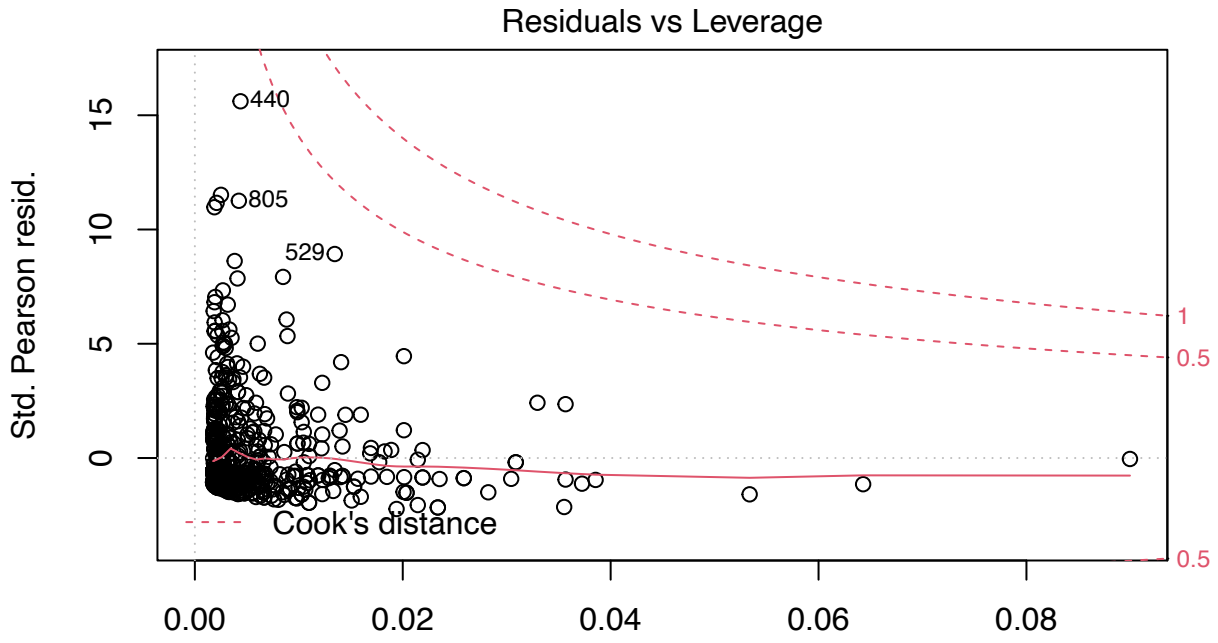
```
## Analysis of Deviance Table
##
```

```
## Model: poisson, link: log
##
## Response: cobol$latest.plot$computed.values$density
##
## Terms added sequentially (first to last)
##
##
##                                                                                 Df
## NULL
## fer$latest.plot$computed.values$density                                          1
## trans$latest.plot$computed.values$density                                        1
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density  1
##                                                                                 Deviance
## NULL
## fer$latest.plot$computed.values$density                                        608.96
## trans$latest.plot$computed.values$density                                       39.42
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density     0.60
##                                                                                 Resid. Df
## NULL                                                                                  857
## fer$latest.plot$computed.values$density                                               856
## trans$latest.plot$computed.values$density                                             855
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density       854
##                                                                                 Resid. Dev
## NULL                                                                             2565.8
## fer$latest.plot$computed.values$density                                          1956.8
## trans$latest.plot$computed.values$density                                        1917.4
## fer$latest.plot$computed.values$density:trans$latest.plot$computed.values$density     1916.8
```

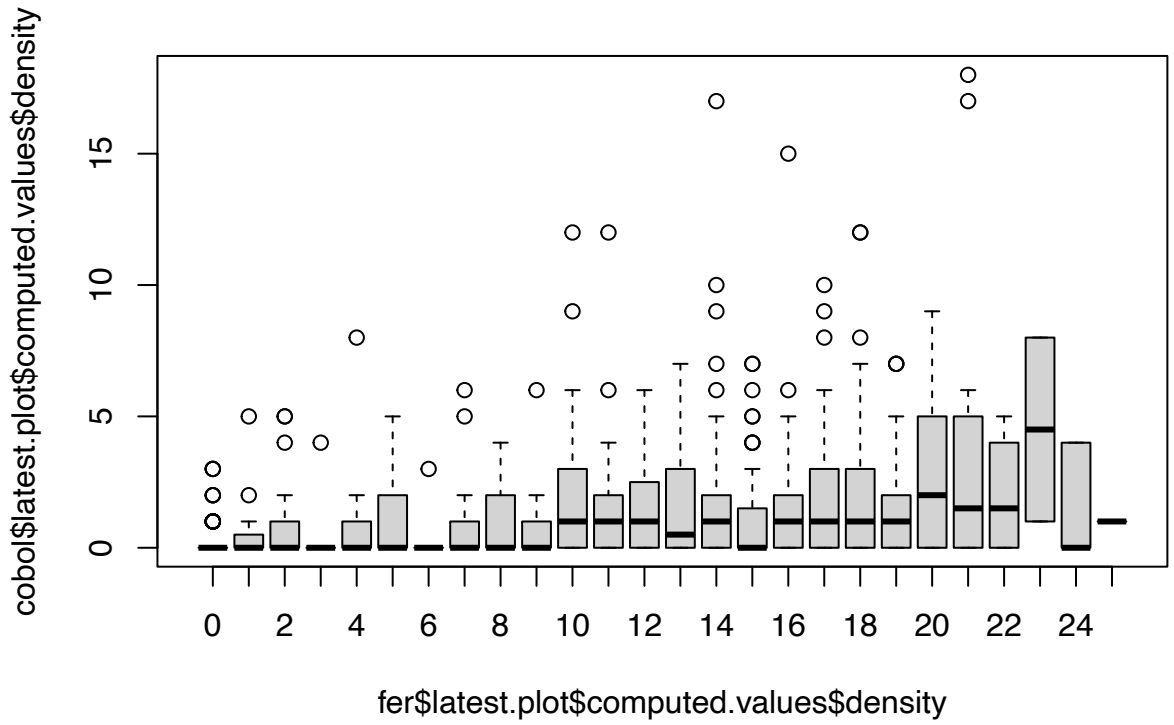
```
plot(glm(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density*trans$latest
```



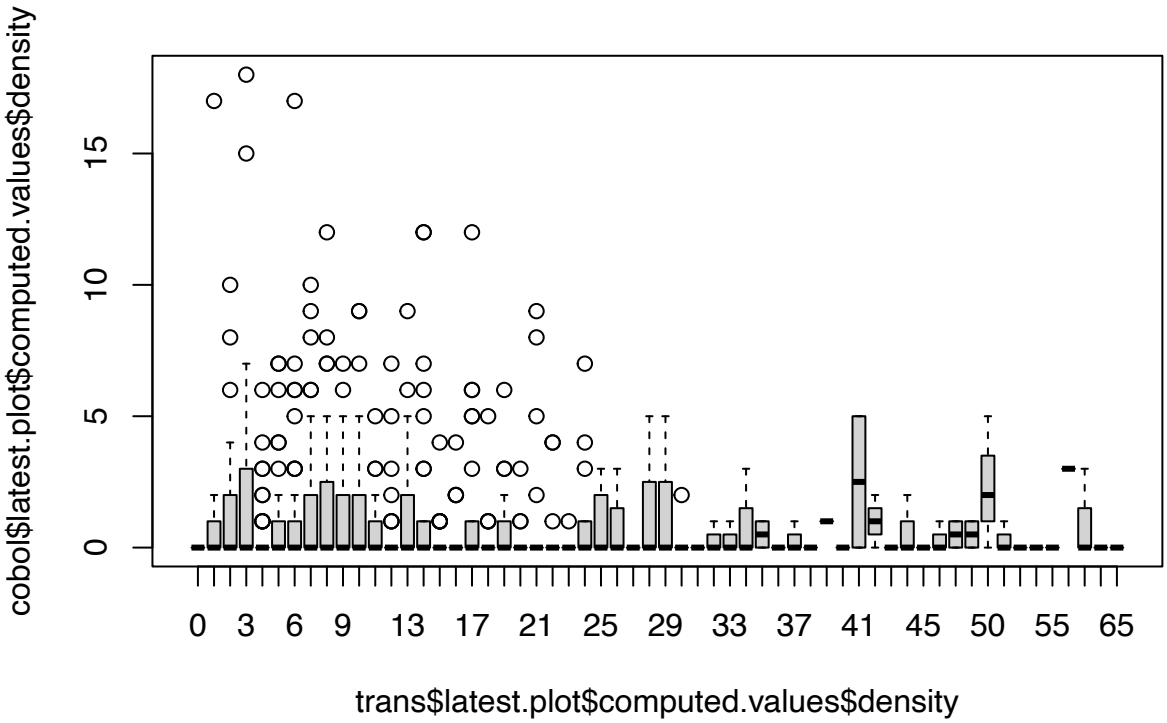

Normal Q–Q
Std. Pearson resid.
440
442
805
−3
−2
−1
0
1
2
3
0
5
10
15
Theoretical Quantiles
glm(cobol$latest.plot$computed.values$density ~ fer$latest.plot$computed.va ...
Scale–Location
√|Std. Pearson resid.|
440
442
805
−2
−1
0
1
0
1
2
3
4
Predicted values
glm(cobol$latest.plot$computed.values$density ~ fer$latest.plot$computed.va ...




```
boxplot(cobol$latest.plot$computed.values$density~fer$latest.plot$computed.values$density)
```




```
boxplot(cobol$latest.plot$computed.values$density~trans$latest.plot$computed.values$density)
```



3.6.4 Taking regions with null gene density to implicitely eliminate the effect of centromeric regions and LRARs.

```
summary(glm(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~fer$
```

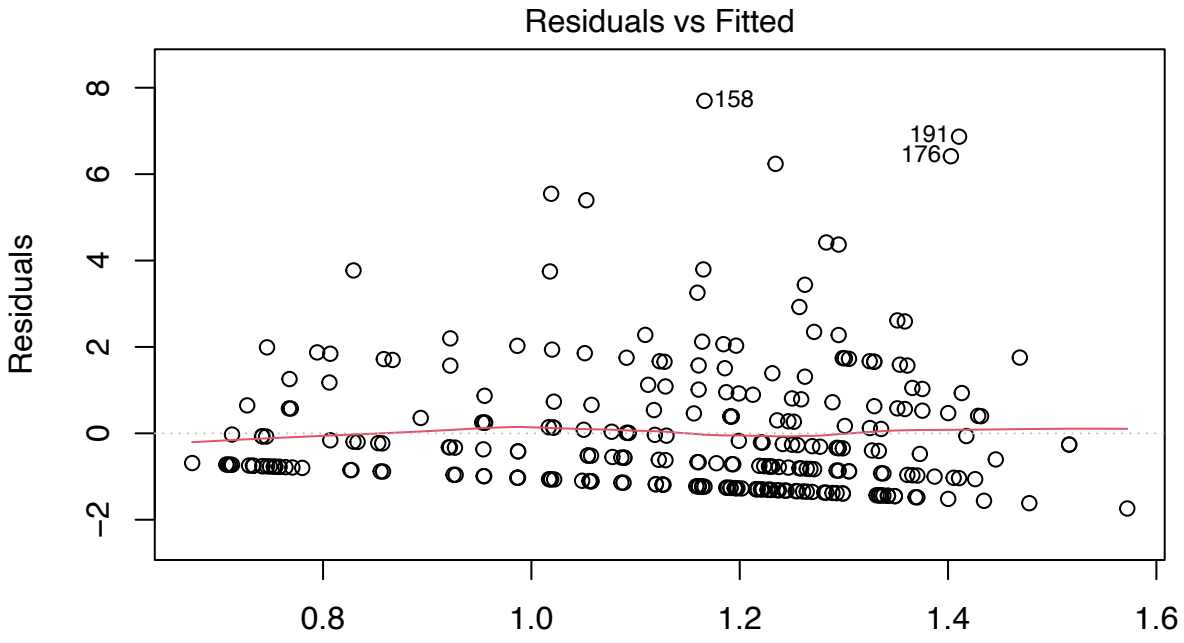
```
##
## Call:
## glm(formula = cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density !=
##     0] ~ fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density !=
##     0] * trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density !=
##     0], family = poisson)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.1187  -1.2252  -0.5951   0.5425   5.3942
##
## Coefficients:
##
## (Intercept)
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
##
## (Intercept)
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
##
## (Intercept)
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
```

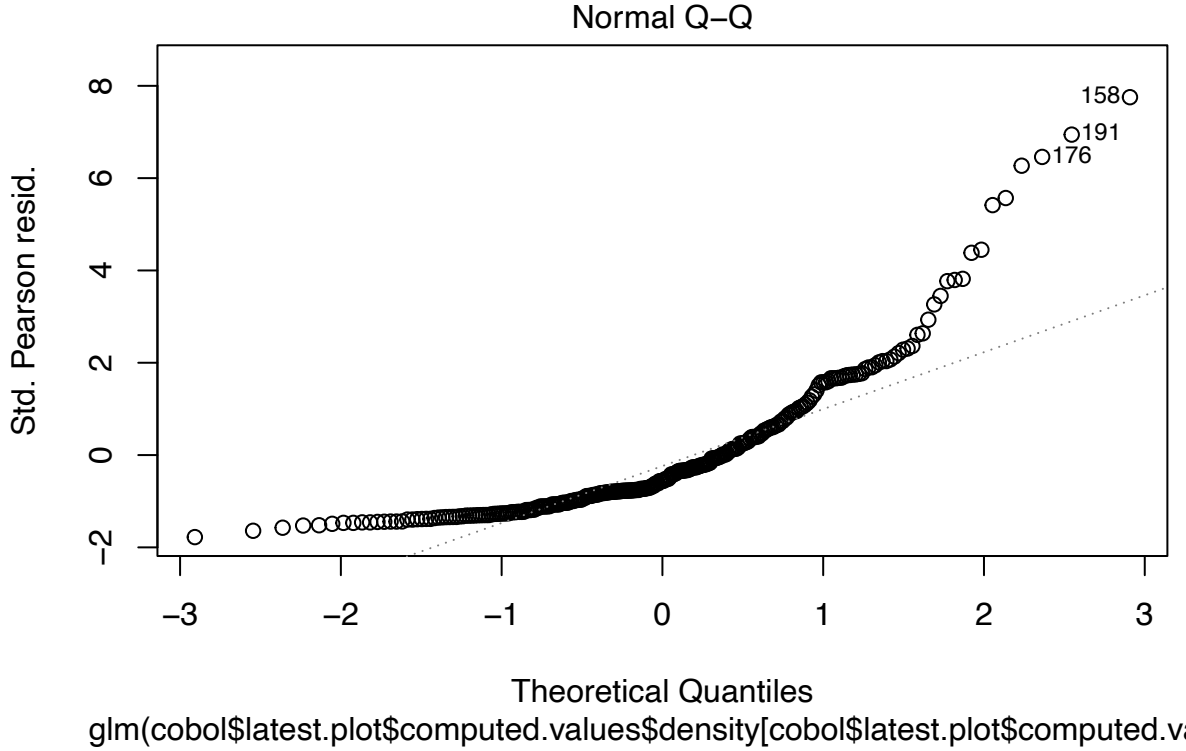
```
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
##
## (Intercept)
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
##
## (Intercept)
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 572.32  on 274  degrees of freedom
## Residual deviance: 539.30  on 271  degrees of freedom
## AIC: 1307.6
##
## Number of Fisher Scoring iterations: 5
```

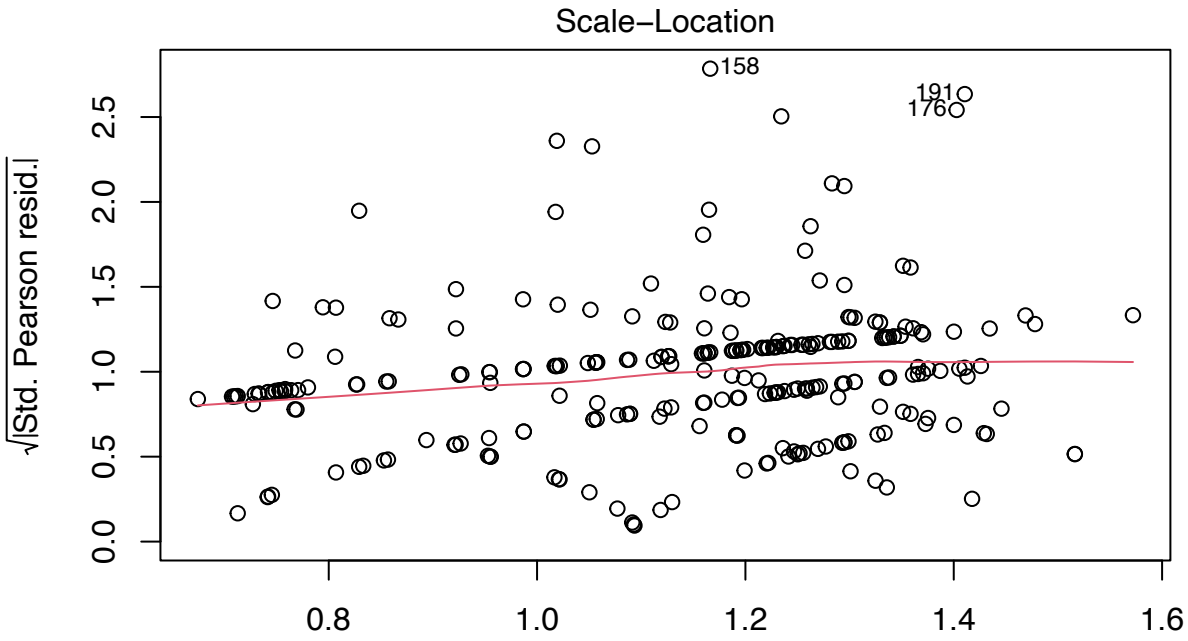
```
anova(glm(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~fer$la
```

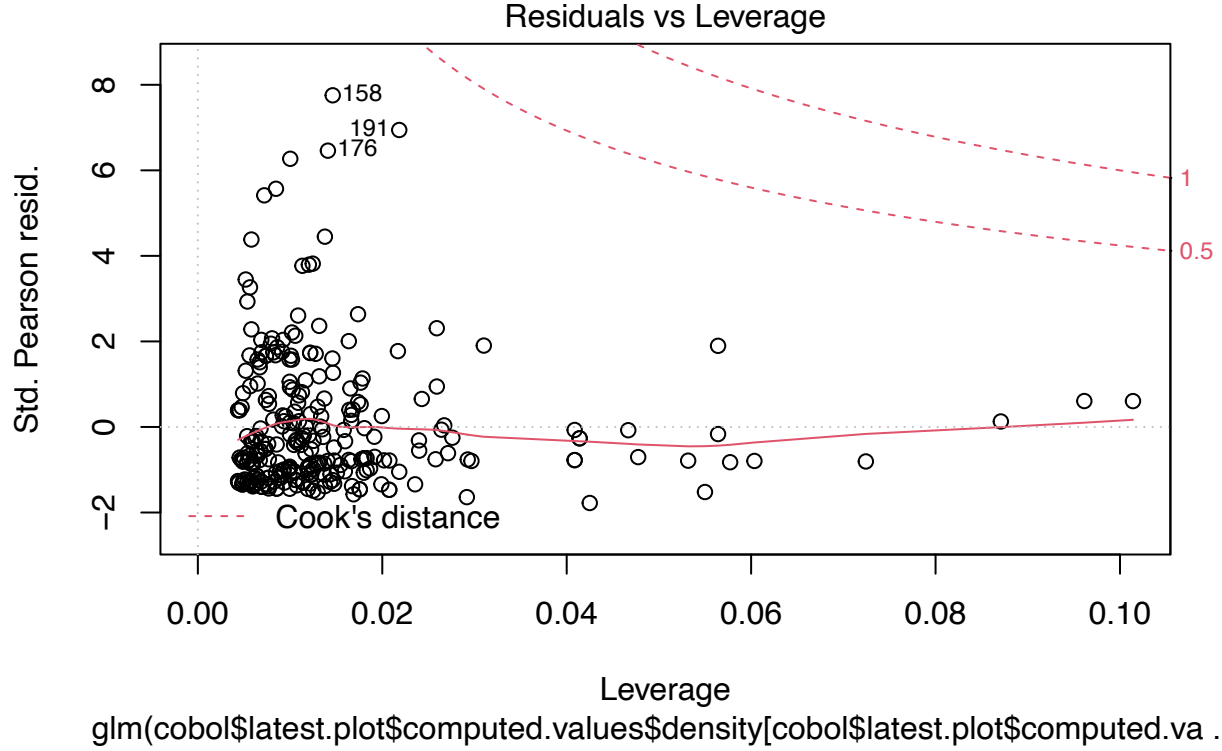
```
## Analysis of Deviance Table
##
## Model: poisson, link: log
##
## Response: cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
##
## Terms added sequentially (first to last)
##
##
##
## NULL
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
##
## NULL
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
##
## NULL
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
##
## NULL
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## trans$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]
## fer$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density != 0]:trans$latest
```

```
plot(glm(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~fer$la
```

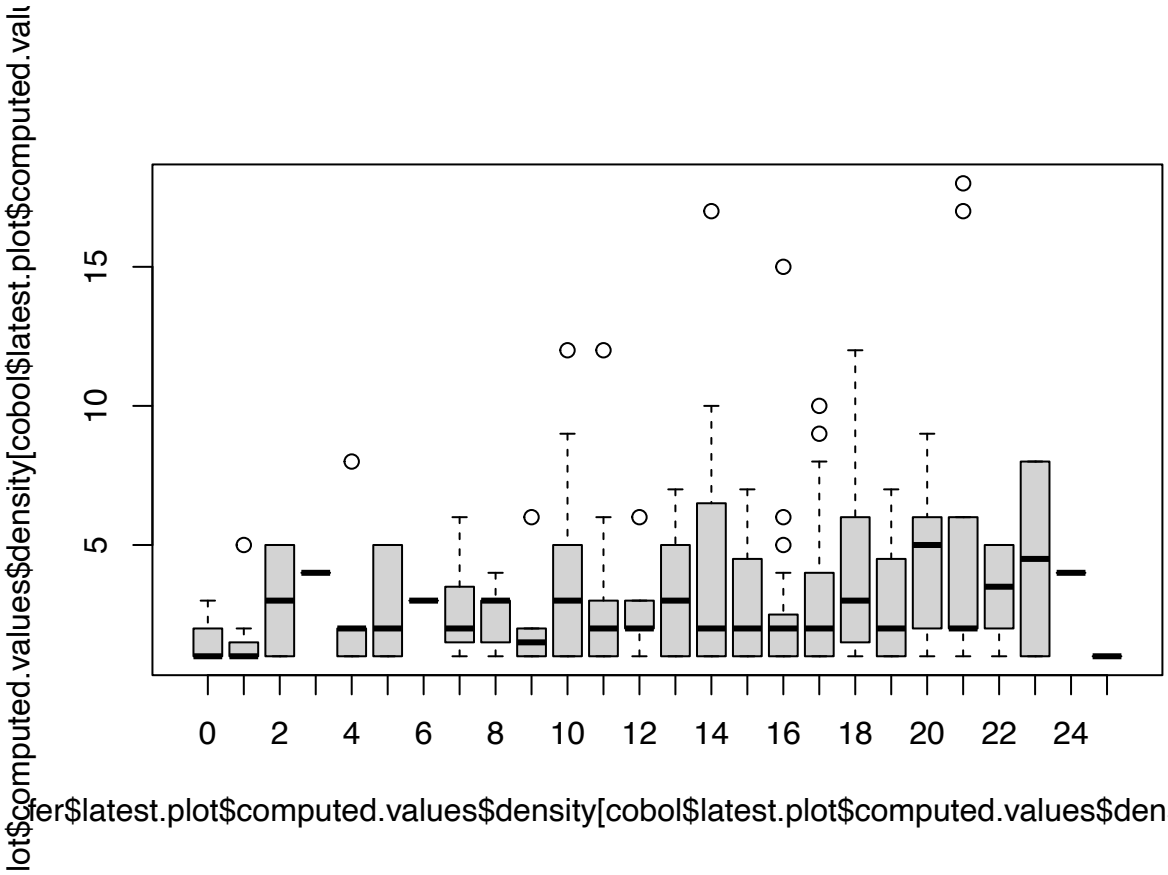




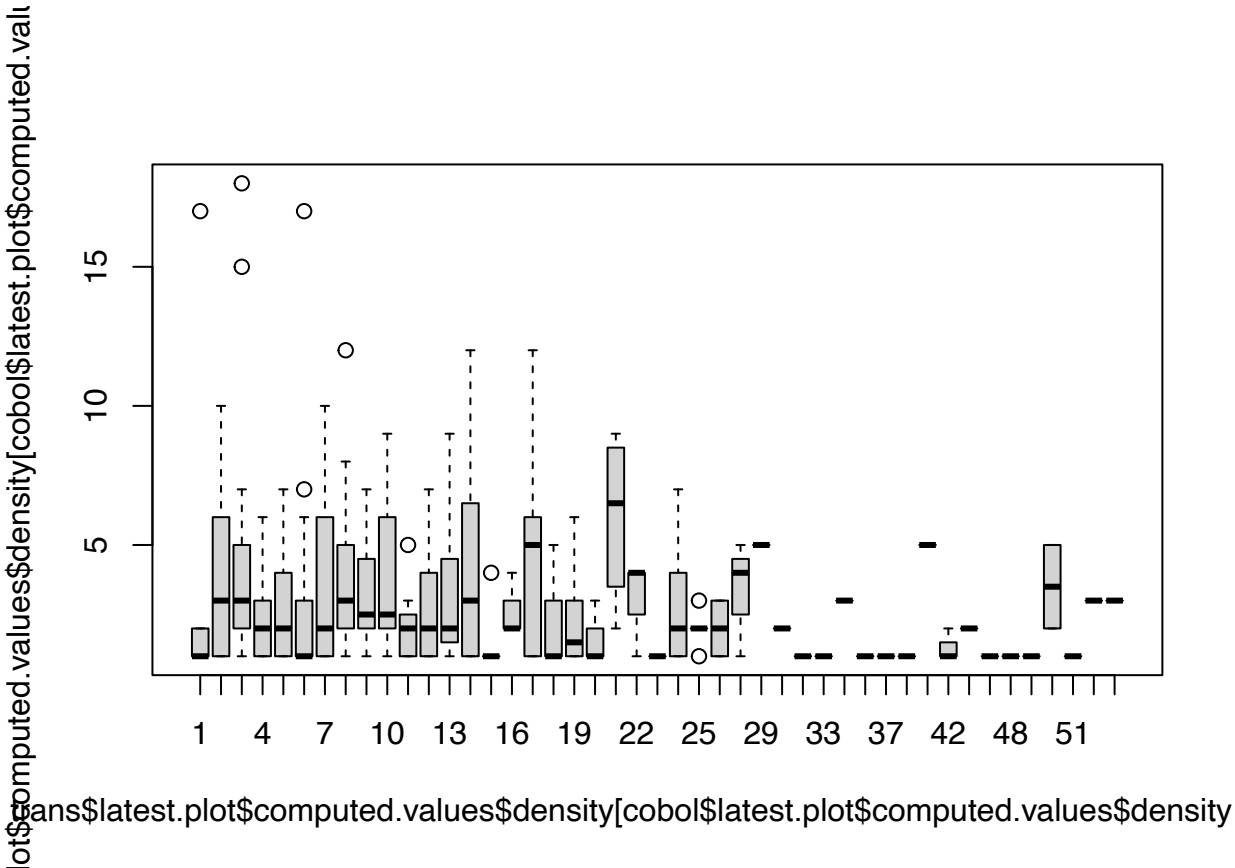





```
boxplot(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~fer$lat
```




```
boxplot(cobol$latest.plot$computed.values$density[cobol$latest.plot$computed.values$density!=0]~trans$la
```

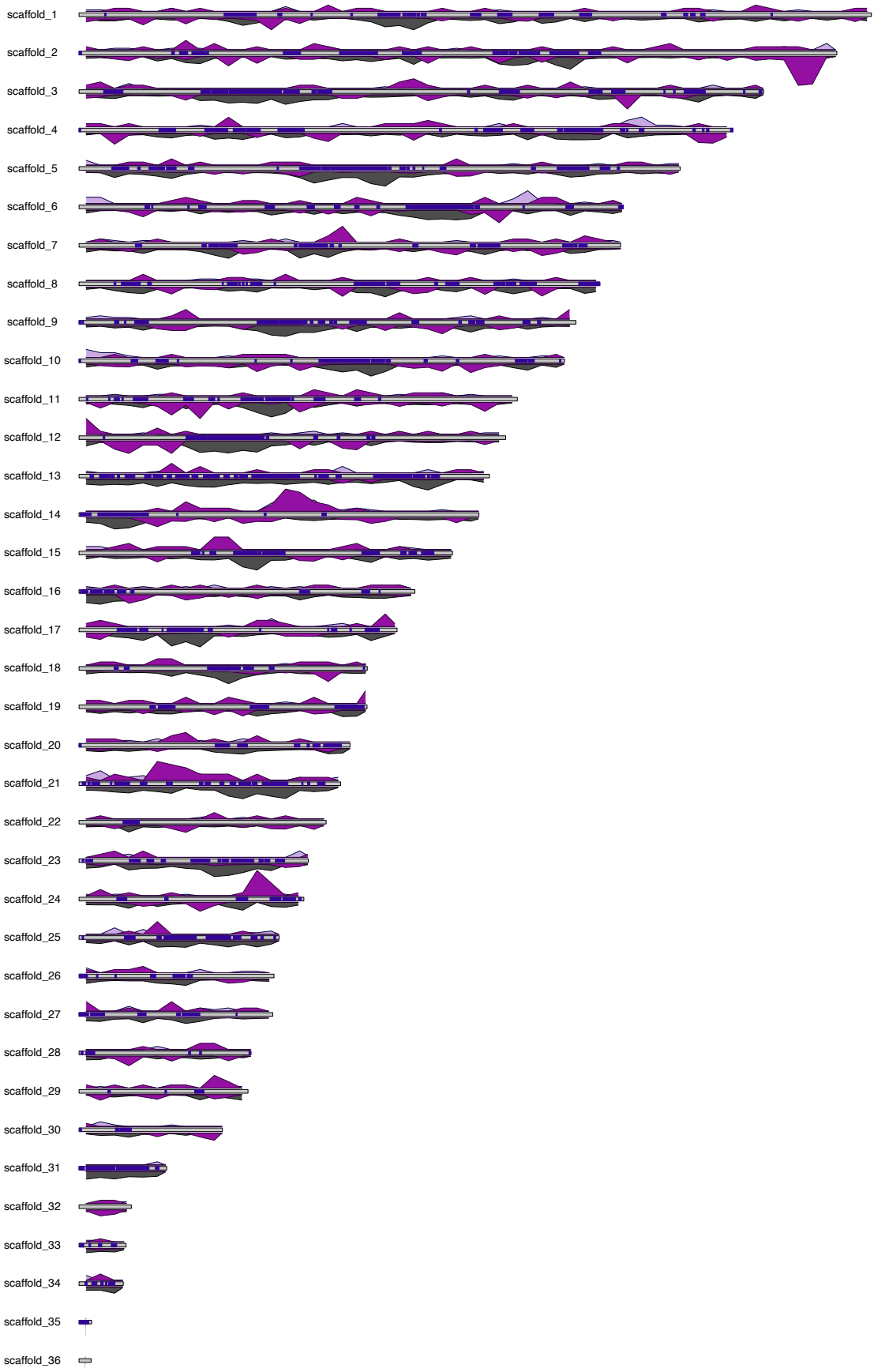


Location of Orphan and underrepresented genes

```
all.ogs<-all.ogs[grep("Pyrenodesmiaerodens",all.ogs$V5),]
foo<-strsplit(all.ogs$V5,", ")
# Get orphan genes
prots_not_ogs<-unlist(foo)
prots_not_ogs<-gsub("-T1","",prots_not_ogs)
prots_not_ogs<-genes2[!(rownames(genes2)%in%prots_not_ogs),]
# Get underrepresented genes
ogs_depth<-sapply(foo,length)
foo<-sapply(foo,FUN=function(x){x<-x[grep("Pyrenodesmiaerodens",x)]})
ogs_depth<-ogs_depth[sapply(foo,length)==1]
foo<-foo[sapply(foo,length)==1]
foo<-foo[ogs_depth<=4]
foo<-unlist(foo)
foo<-gsub("-T1","",foo)
foo<-genes2[(rownames(genes2)%in%foo),]

#Karioplot_both
prots_not_ogs<-makeGRangesFromDataFrame(prots_not_ogs,
                        keep.extra.columns=FALSE,
                        ignore.strand=FALSE,
                        seqinfo=NULL,
                        seqnames.field="scaffold",
                        start.field="start",
                        end.field="end",
                        strand.field="strand",
```

```
                          starts.in.df.are.0based=FALSE)
prots_og_4s<-makeGRangesFromDataFrame(foo,
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="scaffold",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
```

```
kp <- plotKaryotype(genome = pe.genome, plot.type= 2)
kpPlotRegions(kp, data=paste(lrar$Name,":",lrar$Start+1,"-",lrar$End+1,sep=""),data.panel = 2,
              col=spectrum[3], r0=-0.35, r1=-0.1, avoid.overlapping=FALSE)
#kpPlotDensity(kp, ranges_genes2, col="#6001A655",data.panel = 1, r0=0, r1=1, window.size = 50000)
kpPlotDensity(kp, data=mobile_elements,data.panel = 2,col=grey(0.3), r0=0, r1=1,window.size=50000)
kpPlotDensity(kp,lost_genes, col=spectrum[10],data.panel = 2, r0=0, r1=2, window.size = 50000)
kpPlotDensity(kp, data=prots_og_4s,data.panel = 1,col="#6001A655", r0=0, r1=1,window.size=50000)
kpPlotDensity(kp, data=prots_not_ogs,data.panel = 1,col=spectrum[10], r0=0, r1=1.75,window.size=50000)
```

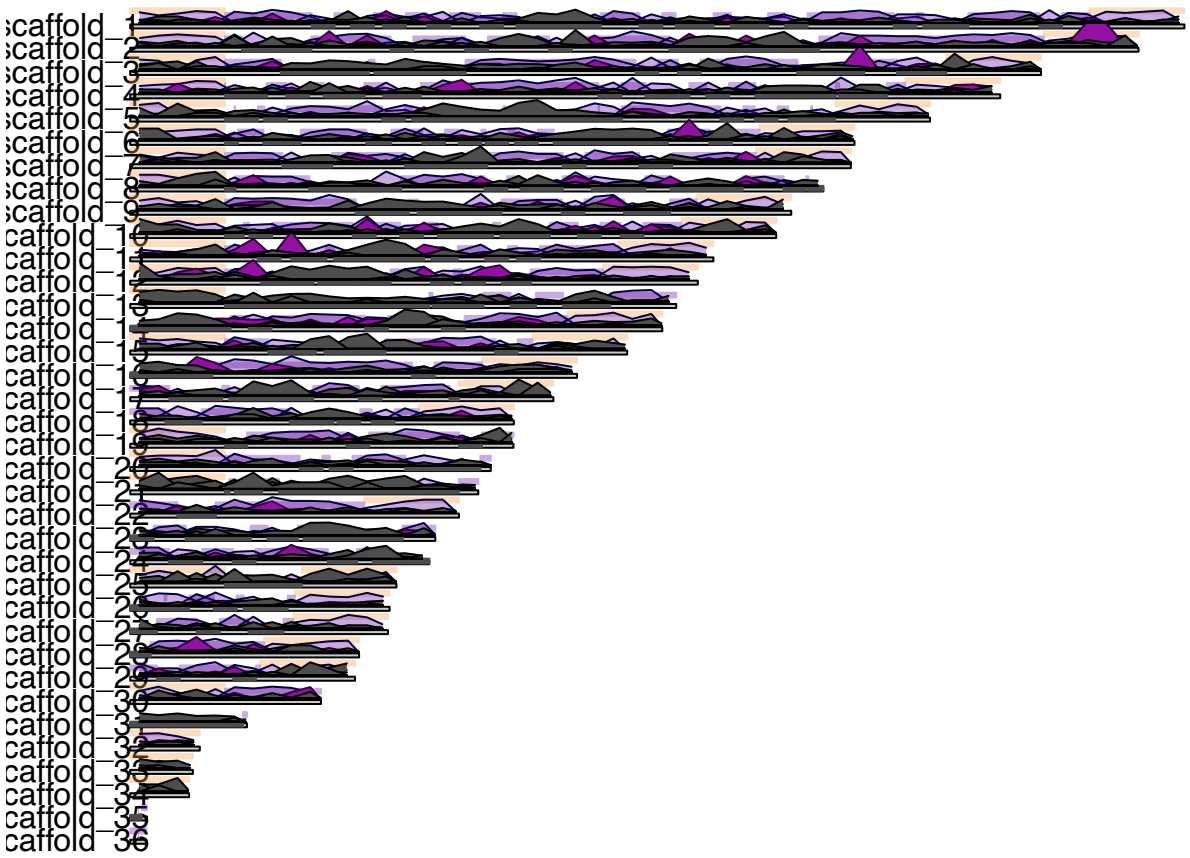

scaffold_1
scaffold_2
scaffold_3
scaffold_4
scaffold_5
scaffold_6
scaffold_7
scaffold_8
scaffold_9
scaffold_10
scaffold_11
scaffold_12
scaffold_13
scaffold_14
scaffold_15
scaffold_16
scaffold_17
scaffold_18
scaffold_19
scaffold_20
scaffold_21
scaffold_22
scaffold_23
scaffold_24
scaffold_25
scaffold_26
scaffold_27
scaffold_28
scaffold_29
scaffold_30
scaffold_31
scaffold_32
scaffold_33
scaffold_34
scaffold_35
scaffold_36

3.7 Compare subtelomeric regions and regions proximal to LRARs with the rest of the genome

```
lrar_flanks<-makeGRangesFromDataFrame(lrar,
                         keep.extra.columns=FALSE,
                         ignore.strand=FALSE,
                         seqinfo=NULL,
                         seqnames.field="Name",
                         start.field="Start",
                         end.field="End",
                         strand.field="strand",
                         starts.in.df.are.0based=FALSE)
lrar_flanks<-flank(lrar_flanks,30000,both=TRUE)
lrar_flanks<-reduce(lrar_flanks)
lrar_flanks<-GenomicRanges::intersect(lrar_flanks,pe.genome)
lrar_flanks<-GenomicRanges::setdiff(lrar_flanks,telomere_locs)

non_telomeric_regions<-suppressWarnings(c(telomere_locs,lrar_flanks))
non_telomeric_regions<-reduce(non_telomeric_regions)
non_telomeric_regions<-GenomicRanges::setdiff(pe.genome,non_telomeric_regions)

kp <- plotKaryotype(genome = pe.genome, plot.type= 1)
kpPlotRegions(kp, lrar_flanks,data.panel = 2,col=grey(0.3), r0=0, r1=-0.3, avoid.overlapping=FALSE)
kpPlotRegions(kp, non_telomeric_regions,data.panel = 2,col="#6001A655", r0=0.3, r1=0.6, avoid.overlappi
kpPlotRegions(kp, telomere_locs,data.panel = 2,col="#F9973E55", r0=0.6, r1=0.9, avoid.overlapping=FALSE
# gene density
genes_density_nt<-kpPlotDensity(kp, subsetByOverlaps(ranges_genes2,non_telomeric_regions), col="#6001A6
genes_density_t<-kpPlotDensity(kp, subsetByOverlaps(ranges_genes2,telomere_locs), col="#6001A655",data.
genes_density_r<-kpPlotDensity(kp, subsetByOverlaps(ranges_genes2,lrar_flanks), col="#6001A655",data.pa
# Lost gene density
lost_density_nt<-kpPlotDensity(kp, subsetByOverlaps(lost_genes,non_telomeric_regions), col=spectrum[10]
lost_density_t<-kpPlotDensity(kp, subsetByOverlaps(lost_genes,telomere_locs), col=spectrum[10],data.pan
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the oth
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_21, scaffold_27, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
lost_density_r<-kpPlotDensity(kp, subsetByOverlaps(lost_genes,lrar_flanks), col=spectrum[10],data.panel
# Mobile element density
trans_density_nt<-kpPlotDensity(kp, subsetByOverlaps(mobile_elements,non_telomeric_regions), data.panel
trans_density_t<-kpPlotDensity(kp, subsetByOverlaps(mobile_elements,telomere_locs),data.panel = 2, col=
trans_density_r<-kpPlotDensity(kp, subsetByOverlaps(mobile_elements,lrar_flanks),data.panel = 2,col=grey
# Underrepresented gene density (orthologs in 1 or two more genomes)
under_density_nt<-kpPlotDensity(kp, subsetByOverlaps(prots_og_4s,non_telomeric_regions), data.panel = 2
under_density_t<-kpPlotDensity(kp, subsetByOverlaps(prots_og_4s,telomere_locs),data.panel = 2, col=grey
under_density_r<-kpPlotDensity(kp, subsetByOverlaps(prots_og_4s,lrar_flanks),data.panel = 2,col=grey(0.3
# Orfan gene density
orfan_density_nt<-kpPlotDensity(kp, subsetByOverlaps(prots_not_ogs,non_telomeric_regions), data.panel =
orfan_density_t<-kpPlotDensity(kp, subsetByOverlaps(prots_not_ogs,telomere_locs),data.panel = 2, col=gr
orfan_density_r<-kpPlotDensity(kp, subsetByOverlaps(prots_not_ogs,lrar_flanks),data.panel = 2,col=grey(
```



```
#
```

```
base.for.circos<-function(GENOME)
{
require(circlize)
#------------------
# Initialize
#------------------
circos.genomicInitialize(data.frame(names=names(GENOME),start=0,end=sapply(GENOME,length)),sector.names
set_track_gap(gap = 0.01)
circos.genomicTrack(ylim=c(0,1), track.height=0.015,bg.col=NA,bg.border=TRUE,cell.padding=c(0,0,0,0))
for (i in names(GENOME))
  {
  foo<-lrar_flanks[lrar_flanks@seqnames==i,]
    if (length(foo)>0)
  {
    circos.genomicRect(cbind(foo@ranges@start,foo@ranges@start+foo@ranges@width-1), sector.index = i,yt
  }
    foo<-telomere_locs[telomere_locs@seqnames==i,]
    if (length(foo)>0)
  {
    circos.genomicRect(cbind(foo@ranges@start,foo@ranges@start+foo@ranges@width-1), sector.index = i,yt
    }
}
circos.genomicTrack(ylim=c(0,1), track.height=0.015,bg.col=NA,bg.border=TRUE,cell.padding=c(0,0,0,0))
# First track LRAR
set_track_gap(gap = 0.02)
```

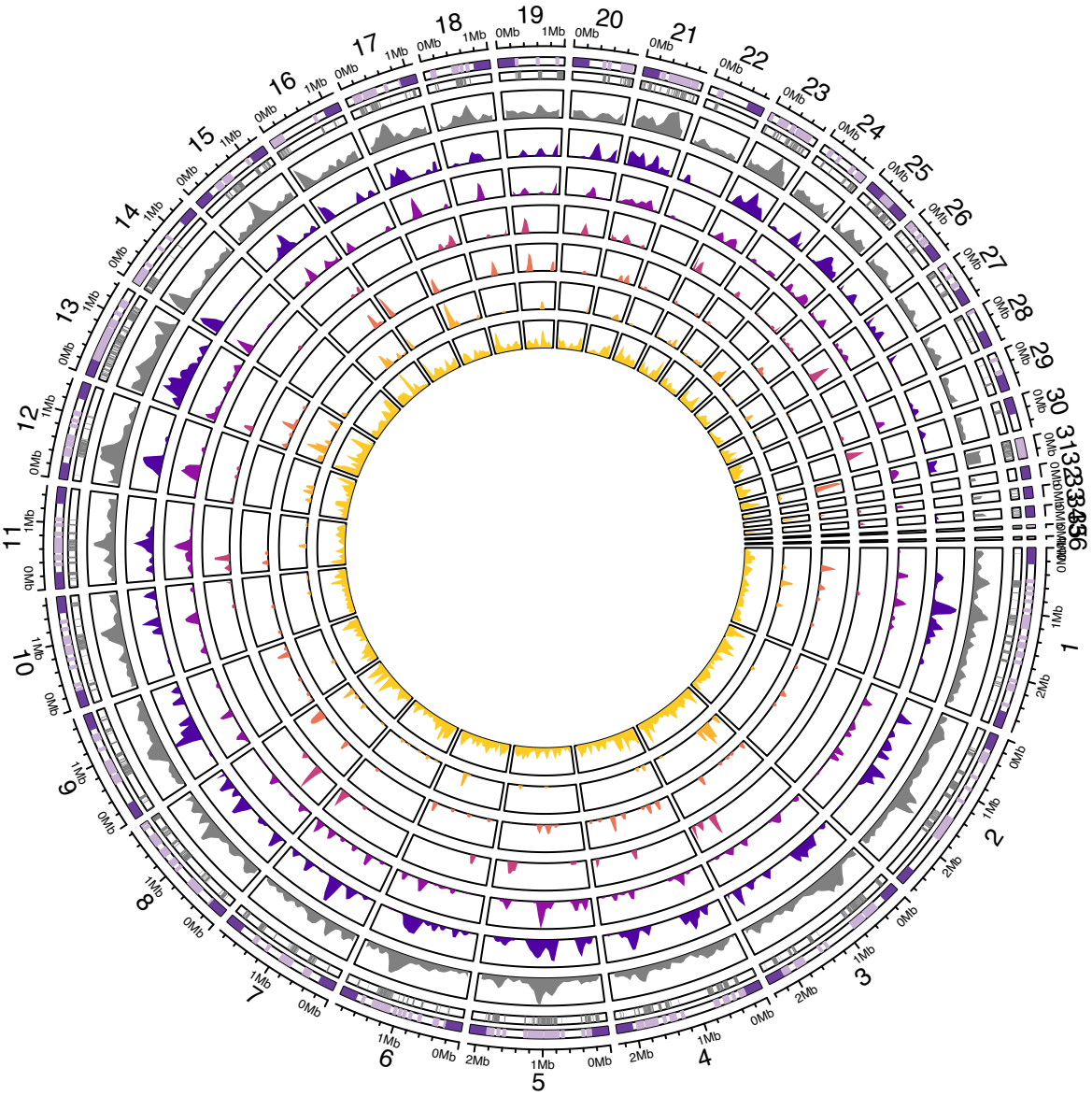
```
for (i in names(GENOME))
{
  foo<-lrar[lrar$Name==i,]
  if (dim(foo)[1]>0)
  {
    circos.genomicRect(foo[,c(2,3)], sector.index = i,ytop = 1, ybottom = 0, col=grey(.5), border = NA)
  }
}
# Second track Density mobile elements
circos.genomicDensity(as.data.frame(mobile_elements),col = grey(0.5), count_by = "number", track.height
# Third Gipsy
circos.genomicDensity(as.data.frame(ltrgypsy),col = spectrum[5], count_by = "number", track.height = 0.
#
circos.genomicDensity(as.data.frame(ltrcopia),col = spectrum[10], count_by = "number", track.height = 0
#
circos.genomicDensity(as.data.frame(ltrNgaro),col = spectrum[15], count_by = "number", track.height = 0
#
circos.genomicDensity(as.data.frame(dna1),col = spectrum[20], count_by = "number", track.height = 0.05,
#
circos.genomicDensity(as.data.frame(helitron),col = spectrum[25], count_by = "number", track.height = 0

circos.genomicDensity(as.data.frame(repetitive_sequences),col = spectrum[27], count_by = "number", trac
#
}

fii<-base.for.circos(pe_genome)
```

```
## Loading required package: circlize
```

```
## ========================================
## circlize version 0.4.15
## CRAN page: https://cran.r-project.org/package=circlize
## Github page: https://github.com/jokergoo/circlize
## Documentation: https://jokergoo.github.io/circlize_book/book/
##
## If you use it in published research, please cite:
## Gu, Z. circlize implements and enhances circular visualization
##   in R. Bioinformatics 2014.
##
## This message can be suppressed by:
##   suppressPackageStartupMessages(library(circlize))
## ========================================
```

```
##
## Attaching package: 'circlize'
```

```
## The following object is masked from 'package:ape':
##
##     degree
```



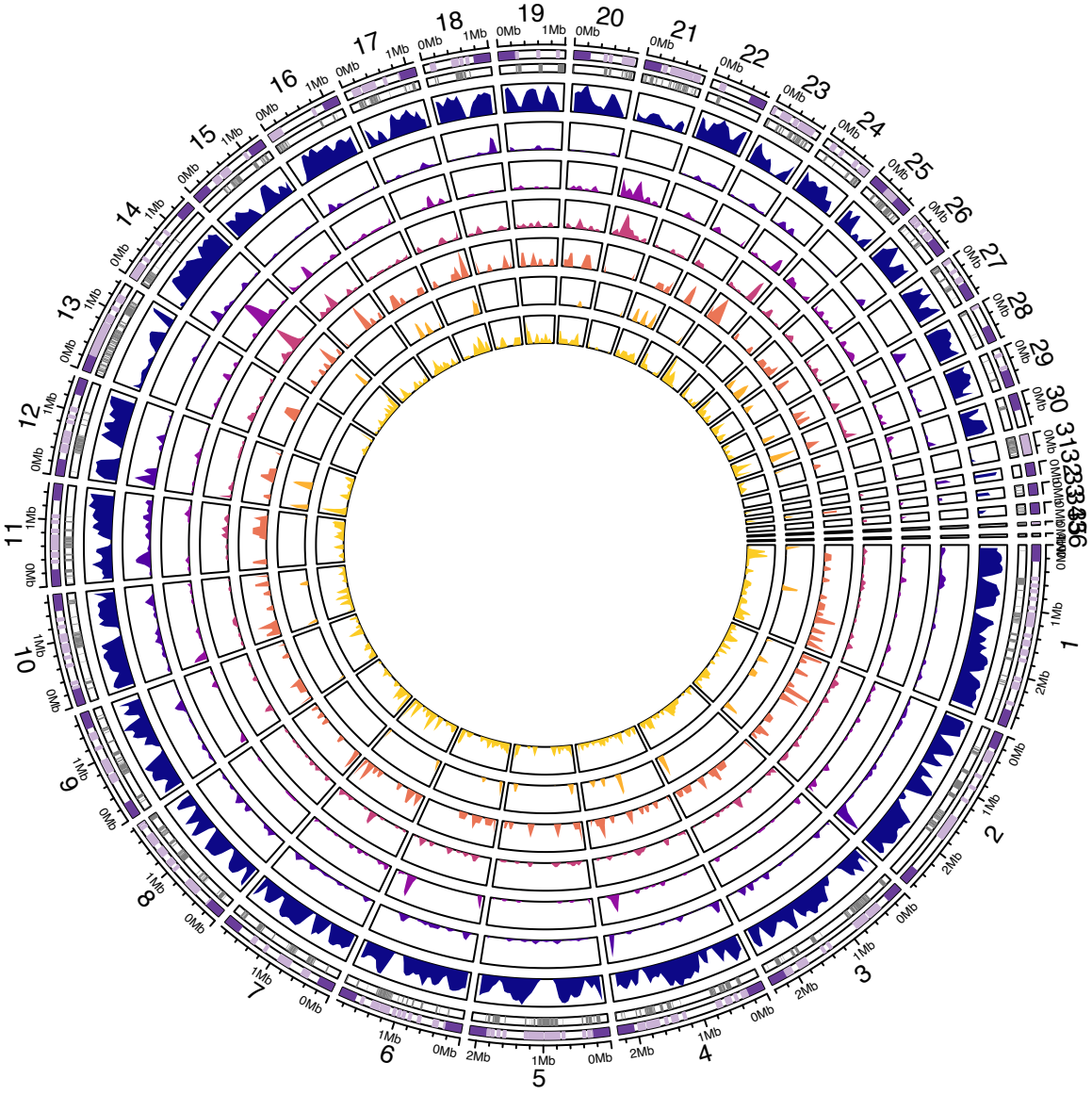
```
print(fii)
```

```
## NULL
```

```
annotations_complete<-read.delim("/Users/Fernando/Desktop/01_paper_sanger_drive/Pyrenodesmia_erodens.an
sec_clust<-makeGRangesFromDataFrame(genes2[genes.all$SecMet.Cluster!="",],
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="scaffold",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)

sec_met<-makeGRangesFromDataFrame(genes2[rownames(annotations_complete)[annotations_complete$Secreted!=
                          keep.extra.columns=FALSE,
```

```
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="scaffold",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
cazymes<-makeGRangesFromDataFrame(genes2[rownames(annotations_complete)[annotations_complete$CAZyme!=""]
                          keep.extra.columns=FALSE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="scaffold",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)

base.for.circos<-function(GENOME)
{
require(circlize)
#------------------
# Initialize
#------------------
circos.genomicInitialize(data.frame(names=names(GENOME),start=0,end=sapply(GENOME,length)),sector.names
set_track_gap(gap = 0.01)
circos.genomicTrack(ylim=c(0,1), track.height=0.015,bg.col=NA,bg.border=TRUE,cell.padding=c(0,0,0,0))
for (i in names(GENOME))
  {
  foo<-lrar_flanks[lrar_flanks@seqnames==i,]
    if (length(foo)>0)
  {
    circos.genomicRect(cbind(foo@ranges@start,foo@ranges@start+foo@ranges@width-1), sector.index = i,yt
  }
    foo<-telomere_locs[telomere_locs@seqnames==i,]
    if (length(foo)>0)
  {
    circos.genomicRect(cbind(foo@ranges@start,foo@ranges@start+foo@ranges@width-1), sector.index = i,yt
    }
}
circos.genomicTrack(ylim=c(0,1), track.height=0.015,bg.col=NA,bg.border=TRUE,cell.padding=c(0,0,0,0))
# First track LRAR
set_track_gap(gap = 0.02)
for (i in names(GENOME))
{
  foo<-lrar[lrar$Name==i,]
  if (dim(foo)[1]>0)
  {
    circos.genomicRect(foo[,c(2,3)], sector.index = i,ytop = 1, ybottom = 0, col=grey(.5), border = NA)
  }
}

# Second track Density genes
circos.genomicDensity(as.data.frame(ranges_genes2),col = spectrum[1], count_by = "number", track.height
```

```
# Missing genes
circos.genomicDensity(as.data.frame(lost_genes),col = spectrum[5], count_by = "number", track.height = 0
# Small OGs
circos.genomicDensity(as.data.frame(prots_og_4s),col = spectrum[10], count_by = "number", track.height =
# Orphan genes
circos.genomicDensity(as.data.frame(prots_not_ogs),col = spectrum[15], count_by = "number", track.height
#
circos.genomicDensity(as.data.frame(cazymes),col = spectrum[20], count_by = "number", track.height = 0.0
#
circos.genomicDensity(as.data.frame(sec_clust),col = spectrum[25], count_by = "number", track.height = 0
#
circos.genomicDensity(as.data.frame(sec_met),col = spectrum[27], count_by = "number", track.height = 0.0
}

fii<-base.for.circos(pe_genome)
```



```
print(fii)
```

```
## NULL
```

```
kp <- plotKaryotype(genome = pe.genome, plot.type= 1)

#ltrgypsy
ltrgypsy_density_nt<-kpPlotDensity(kp, subsetByOverlaps(ltrgypsy,non_telomeric_regions), data.panel = 1
ltrgypsy_density_t<-kpPlotDensity(kp, subsetByOverlaps(ltrgypsy,telomere_locs),data.panel = 1, col=grey
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24, scaffold_31, scaffold_35, scaffold_36
##   - in 'y': scaffold_33
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
ltrgypsy_density_r<-kpPlotDensity(kp, subsetByOverlaps(ltrgypsy,lrar_flanks),data.panel = 1,col=grey(0.3
#ltrcopia
ltrcopia_density_nt<-kpPlotDensity(kp, subsetByOverlaps(ltrcopia,non_telomeric_regions), data.panel = 1
ltrcopia_density_t<-kpPlotDensity(kp, subsetByOverlaps(ltrcopia,telomere_locs),data.panel = 1, col=grey
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24, scaffold_31, scaffold_35, scaffold_36
##   - in 'y': scaffold_30
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
ltrcopia_density_r<-kpPlotDensity(kp, subsetByOverlaps(ltrcopia,lrar_flanks),data.panel = 1,col=grey(0.3
#ltrNgaro
ltrNgaro_density_nt<-kpPlotDensity(kp, subsetByOverlaps(ltrNgaro,non_telomeric_regions), data.panel = 1
ltrNgaro_density_t<-kpPlotDensity(kp, subsetByOverlaps(ltrNgaro,telomere_locs),data.panel = 1, col=grey
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_16, scaffold_2, scaffold_22, scaffold_32, scaffold_9
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
ltrNgaro_density_r<-kpPlotDensity(kp, subsetByOverlaps(ltrNgaro,lrar_flanks),data.panel = 1,col=grey(0.3
#dna1
dna1_density_nt<-kpPlotDensity(kp, subsetByOverlaps(dna1,non_telomeric_regions), data.panel = 1, col=gre
dna1_density_t<-kpPlotDensity(kp, subsetByOverlaps(dna1,telomere_locs),data.panel = 1, col=grey(0.3), r0
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24, scaffold_31, scaffold_36
##   - in 'y': scaffold_12, scaffold_14, scaffold_25, scaffold_30, scaffold_32, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
dna1_density_r<-kpPlotDensity(kp, subsetByOverlaps(dna1,lrar_flanks),data.panel = 1,col=grey(0.3), r0=0
#helitron
helitron_density_nt<-kpPlotDensity(kp, subsetByOverlaps(helitron,non_telomeric_regions), data.panel = 1
helitron_density_t<-kpPlotDensity(kp, subsetByOverlaps(helitron,telomere_locs),data.panel = 1, col=grey
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24, scaffold_31, scaffold_36
##   - in 'y': scaffold_29, scaffold_30, scaffold_32
```

```
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
helitron_density_r<-kpPlotDensity(kp, subsetByOverlaps(helitron,lrar_flanks),data.panel = 1,col=grey(0.3
#repetitive_sequences
repetitive_sequences_density_nt<-kpPlotDensity(kp, subsetByOverlaps(repetitive_sequences,non_telomeric_
repetitive_sequences_density_t<-kpPlotDensity(kp, subsetByOverlaps(repetitive_sequences,telomere_locs),
repetitive_sequences_density_r<-kpPlotDensity(kp, subsetByOverlaps(repetitive_sequences,lrar_flanks),da

# cazyme density
cazymes_density_nt<-kpPlotDensity(kp, subsetByOverlaps(cazymes,non_telomeric_regions), data.panel = 1,
cazymes_density_t<-kpPlotDensity(kp, subsetByOverlaps(cazymes,telomere_locs),data.panel = 1, col=grey(0
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_32, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

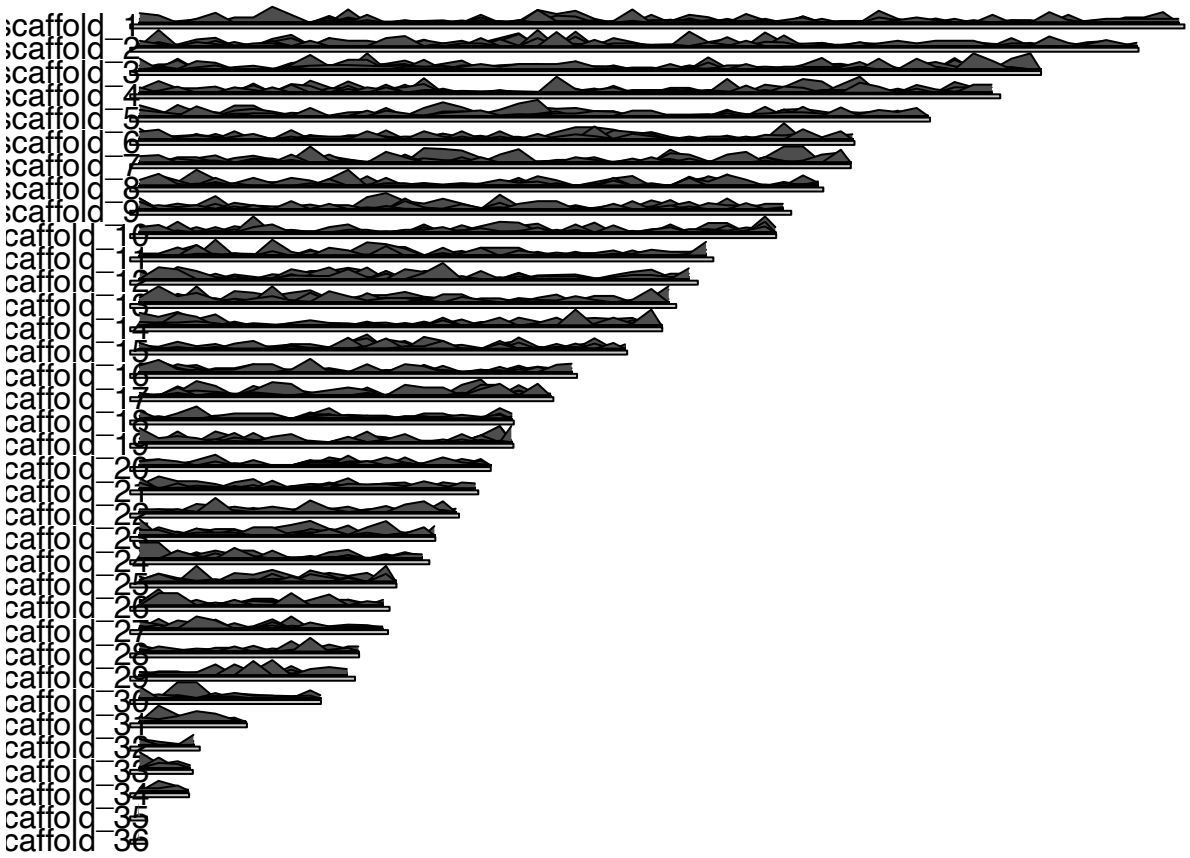
```
cazymes_density_r<-kpPlotDensity(kp, subsetByOverlaps(cazymes,lrar_flanks),data.panel = 1,col=grey(0.3)
# secmet density
sec_met_density_nt<-kpPlotDensity(kp, subsetByOverlaps(sec_met,non_telomeric_regions), data.panel = 2,
sec_met_density_t<-kpPlotDensity(kp, subsetByOverlaps(sec_met,telomere_locs),data.panel = 2, col=grey(0
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
sec_met_density_r<-kpPlotDensity(kp, subsetByOverlaps(sec_met,lrar_flanks),data.panel = 2,col=grey(0.3)
# clust density
sec_clust_density_nt<-kpPlotDensity(kp, subsetByOverlaps(sec_clust,non_telomeric_regions), data.panel =
sec_clust_density_t<-kpPlotDensity(kp, subsetByOverlaps(sec_clust,telomere_locs),data.panel = 2, col=gr
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23
##   - in 'y': scaffold_13, scaffold_15, scaffold_18, scaffold_19, scaffold_25, scaffold_32, scaffold_3
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
sec_clust_density_r<-kpPlotDensity(kp, subsetByOverlaps(sec_clust,lrar_flanks),data.panel = 2,col=grey(
```



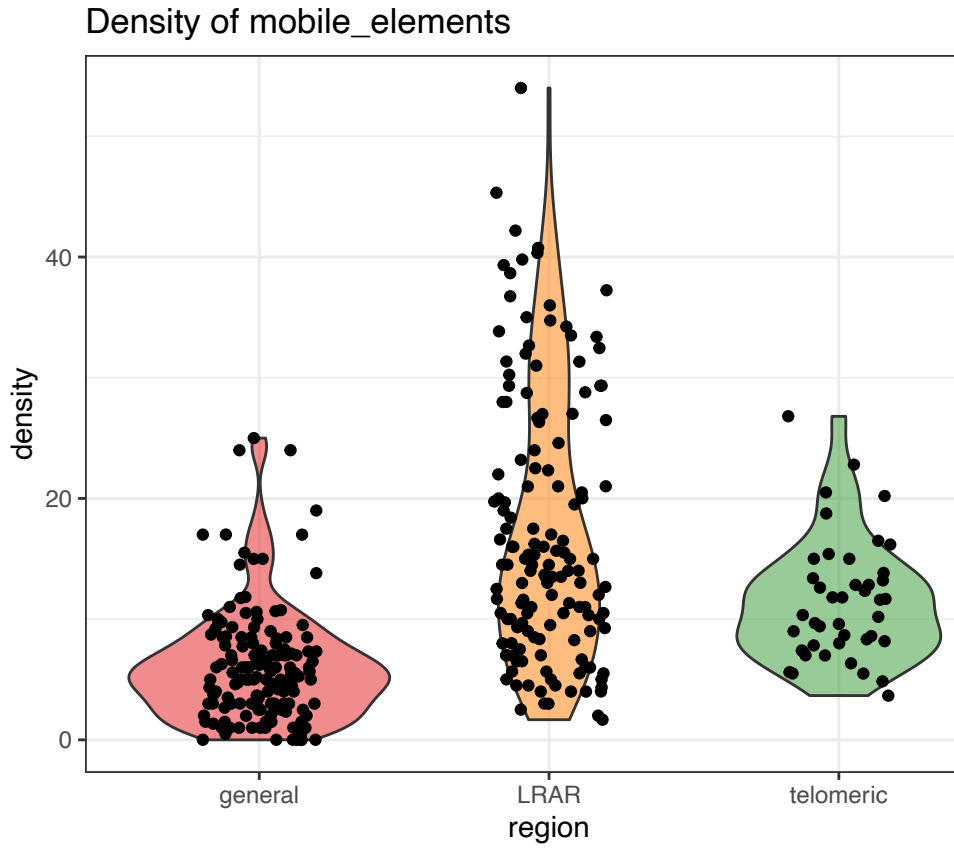
Differences in gene density between Telomeric, Centromeric+LRAR flanking regions and the rest of the genome

```
ranges_results<-NULL
for (REGION in c("general","telomeric","LRAR"))
  {
  if (REGION=="general")
    {ranges2use<-non_telomeric_regions
  }else if (REGION=="telomeric")
    {ranges2use<-telomere_locs
  }else if (REGION=="LRAR")
  {ranges2use<-lrar_flanks
    }
  for (TYPE in c("gene_density","lost_genes","mobile_elements","ltrgypsy","ltrcopia","ltrNgaro","dna1",
  {
   if (REGION=="general"&TYPE=="gene_density")
    {data2use<-genes_density_nt
  }else if (REGION=="telomeric"&TYPE=="gene_density")
    {data2use<-genes_density_t
  }else if (REGION=="LRAR"&TYPE=="gene_density")
  {data2use<-genes_density_r
    } else if (REGION=="general"&TYPE=="lost_genes")
    {data2use<-lost_density_nt
  }else if (REGION=="telomeric"&TYPE=="lost_genes")
    {data2use<-lost_density_t
  }else if (REGION=="LRAR"&TYPE=="lost_genes")
  {data2use<-lost_density_r
    } else    if (REGION=="general"&TYPE=="mobile_elements")
```

```
  {data2use<-trans_density_nt
}else if (REGION=="telomeric"&TYPE=="mobile_elements")
  {data2use<-trans_density_t
}else if (REGION=="LRAR"&TYPE=="mobile_elements")
{data2use<-trans_density_r
  } else    if (REGION=="general"&TYPE=="small_ogs")
  {data2use<-under_density_nt
}else if (REGION=="telomeric"&TYPE=="small_ogs")
  {data2use<-under_density_t
}else if (REGION=="LRAR"&TYPE=="small_ogs")
{data2use<-under_density_r
  } else    if (REGION=="general"&TYPE=="orphan")
  {data2use<-orfan_density_nt
}else if (REGION=="telomeric"&TYPE=="orphan")
  {data2use<-orfan_density_t
}else if (REGION=="LRAR"&TYPE=="orphan")
{data2use<-orfan_density_r
} else    if (REGION=="general"&TYPE=="cazymes")
  {data2use<-cazymes_density_nt
}else if (REGION=="telomeric"&TYPE=="cazymes")
  {data2use<-cazymes_density_t
}else if (REGION=="LRAR"&TYPE=="cazymes")
{data2use<-cazymes_density_r
}else    if (REGION=="general"&TYPE=="sec_met")
  {data2use<-sec_met_density_nt
}else if (REGION=="telomeric"&TYPE=="sec_met")
  {data2use<-sec_met_density_t
}else if (REGION=="LRAR"&TYPE=="sec_met")
{data2use<-sec_met_density_r
}else    if (REGION=="general"&TYPE=="sec_clust")
  {data2use<-sec_clust_density_nt
}else if (REGION=="telomeric"&TYPE=="sec_clust")
  {data2use<-sec_clust_density_t
}else if (REGION=="LRAR"&TYPE=="sec_clust")
{data2use<-sec_clust_density_r
}else    if (REGION=="general"&TYPE=="ltrgypsy")
  {data2use<-ltrgypsy_density_nt
}else if (REGION=="telomeric"&TYPE=="ltrgypsy")
  {data2use<-ltrgypsy_density_t
}else if (REGION=="LRAR"&TYPE=="ltrgypsy")
{data2use<-ltrgypsy_density_r
}else if (REGION=="general"&TYPE=="ltrcopia")
  {data2use<-ltrcopia_density_nt
}else if (REGION=="telomeric"&TYPE=="ltrcopia")
  {data2use<-ltrcopia_density_t
}else if (REGION=="LRAR"&TYPE=="ltrcopia")
{data2use<-ltrcopia_density_r
}else if (REGION=="general"&TYPE=="ltrNgaro")
  {data2use<-ltrNgaro_density_nt
}else if (REGION=="telomeric"&TYPE=="ltrNgaro")
  {data2use<-ltrNgaro_density_t
}else if (REGION=="LRAR"&TYPE=="ltrNgaro")
{data2use<-ltrNgaro_density_r
```

```
  }else    if (REGION=="general"&TYPE=="dna1")
    {data2use<-dna1_density_nt
  }else if (REGION=="telomeric"&TYPE=="dna1")
    {data2use<-dna1_density_t
  }else if (REGION=="LRAR"&TYPE=="dna1")
  {data2use<-dna1_density_r
  }else    if (REGION=="general"&TYPE=="helitron")
    {data2use<-helitron_density_nt
  }else if (REGION=="telomeric"&TYPE=="helitron")
    {data2use<-helitron_density_t
  }else if (REGION=="LRAR"&TYPE=="helitron")
  {data2use<-helitron_density_r
  }else    if (REGION=="general"&TYPE=="repetitive_sequences")
    {data2use<-repetitive_sequences_density_nt
  }else if (REGION=="telomeric"&TYPE=="repetitive_sequences")
    {data2use<-repetitive_sequences_density_t
  }else if (REGION=="LRAR"&TYPE=="repetitive_sequences")
  {data2use<-repetitive_sequences_density_r
  }
  for (I in 1:length(ranges2use))
{
ranges_results<-rbind(ranges_results,cbind(REGION,TYPE,I,mean(  data2use$latest.plot$computed.values$den
}
    }
}
ranges_results<-data.frame(region=factor(ranges_results[,1]),feature=factor(ranges_results[,2]),local=ra
```

```
ggplot(ranges_results[ranges_results$feature=="mobile_elements",],aes(x=region,y=density,fill=region,z=
```

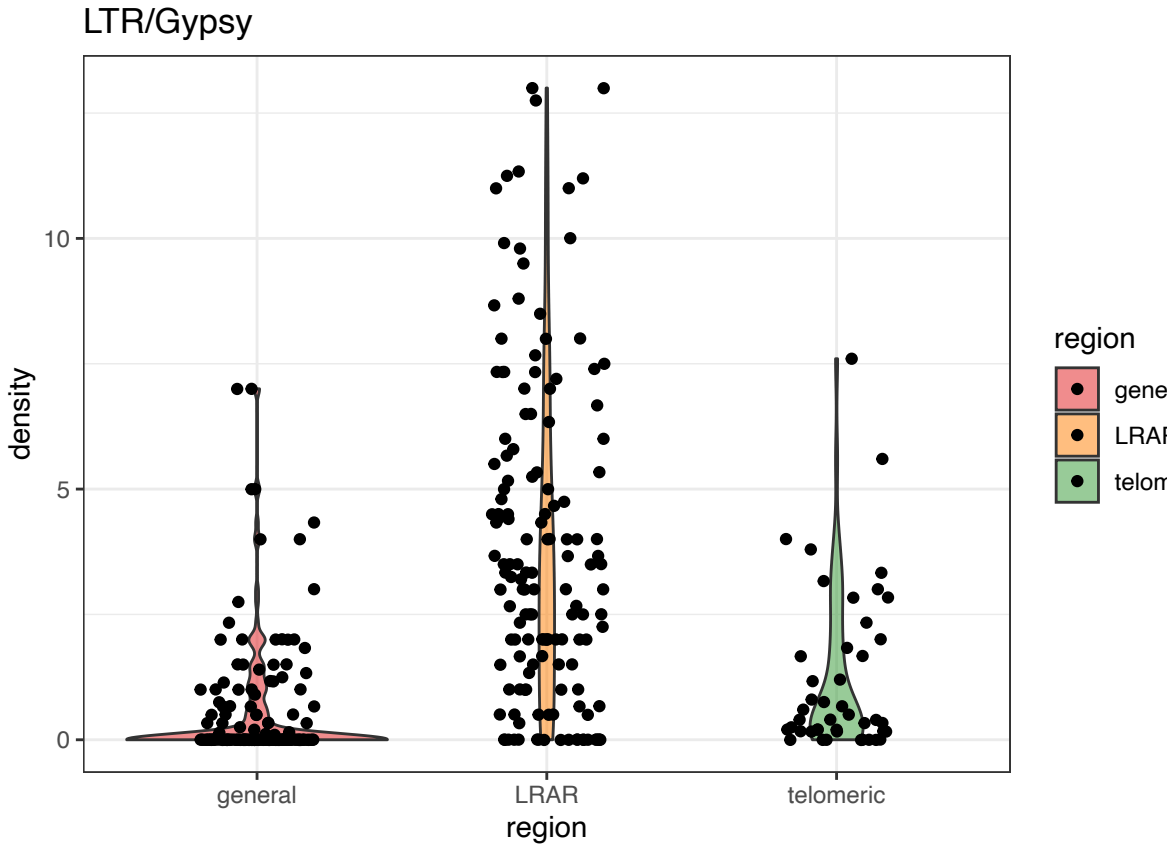


3.7.0.1 Density of mobile elements

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="mobile_elements","density"],ranges_results
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "mobile_elements", "density"] and ranges_results[rang
##
##           general LRAR
## LRAR      < 2e-16 -
## telomeric 1.2e-10 0.0045
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="ltrgypsy",],aes(x=region,y=density,fill=region,z=feature
```

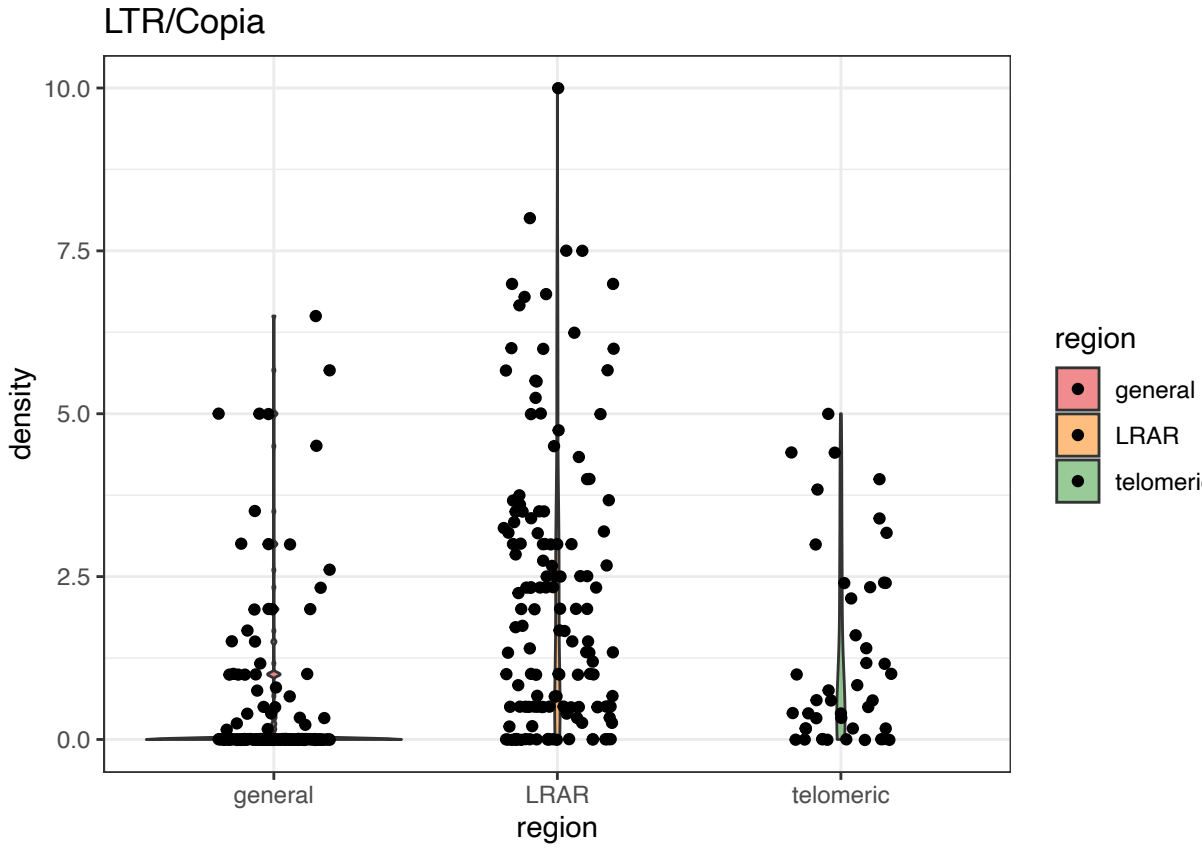


3.7.0.2 LTRgypsy

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="ltrgypsy","density"],ranges_results[ranges_
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "ltrgypsy", "density"] and ranges_results[ranges_resu
##
##           general LRAR
## LRAR      < 2e-16 -
## telomeric 1.3e-06 6.0e-07
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="ltrcopia",],aes(x=region,y=density,fill=region,z=feature
```

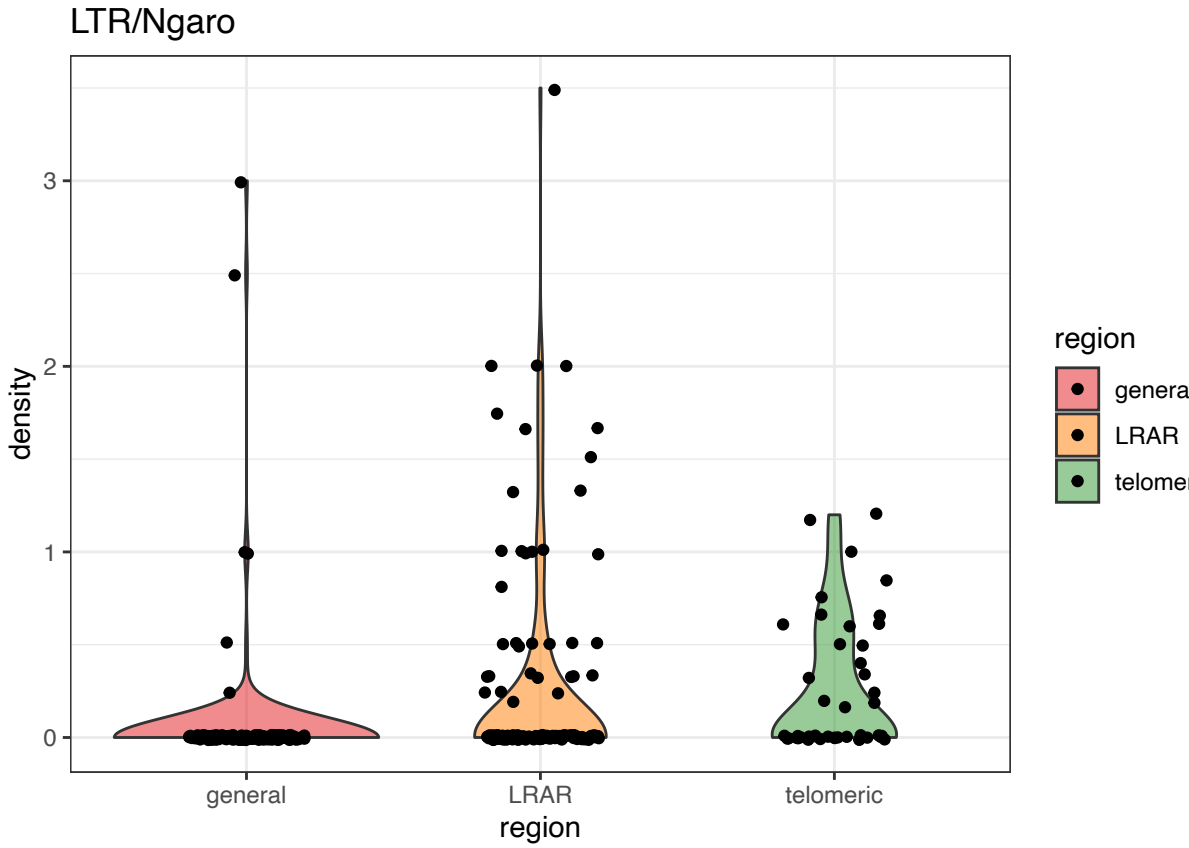


3.7.0.3 ltrcopia

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="ltrcopia","density"],ranges_results[ranges_
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "ltrcopia", "density"] and ranges_results[ranges_resu
##
##           general LRAR
## LRAR      < 2e-16 -
## telomeric 5.5e-09 0.013
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="ltrNgaro",],aes(x=region,y=density,fill=region,z=feature
```

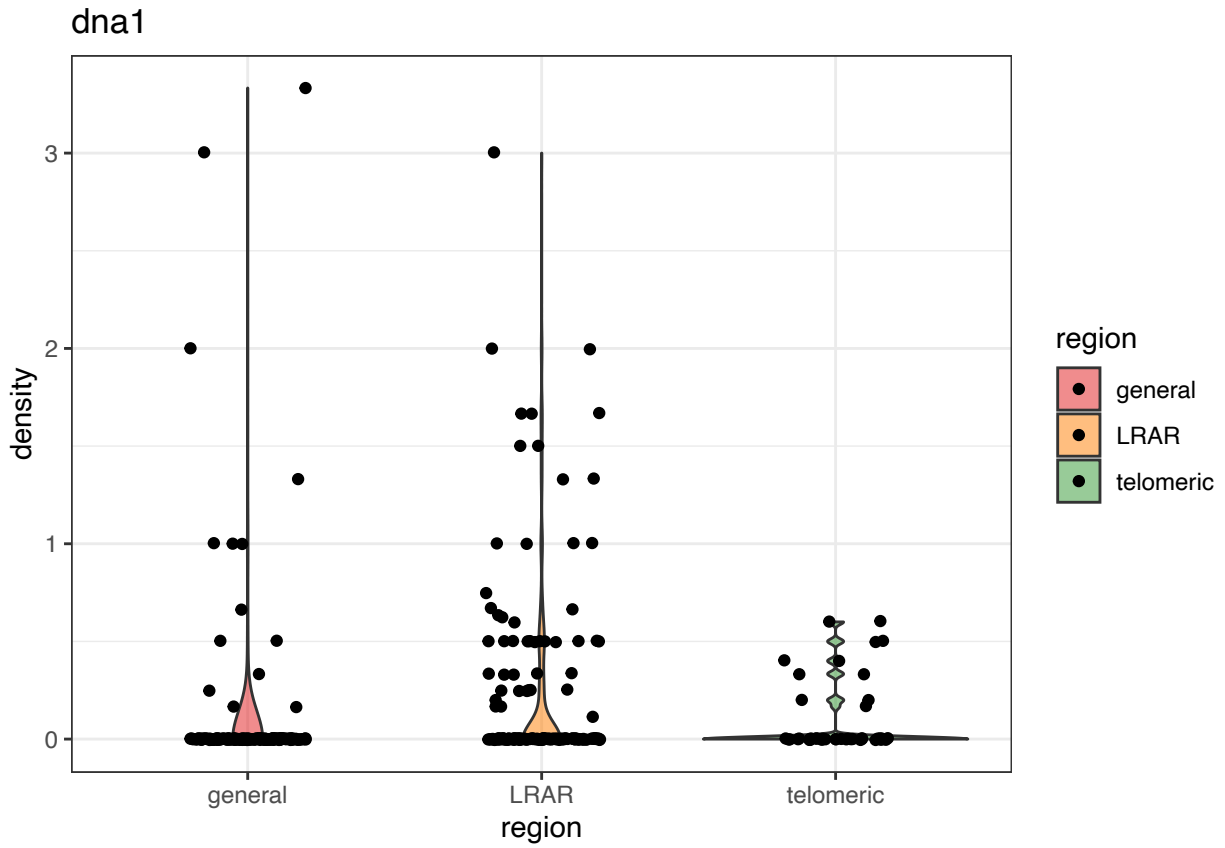


3.7.0.4 ltrNgaro

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="ltrNgaro","density"],ranges_results[ranges
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "ltrNgaro", "density"] and ranges_results[ranges_resu
##
##           general LRAR
## LRAR      6.5e-07 -
## telomeric 4.5e-11 0.041
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="dna1",],aes(x=region,y=density,fill=region,z=feature))+g
```

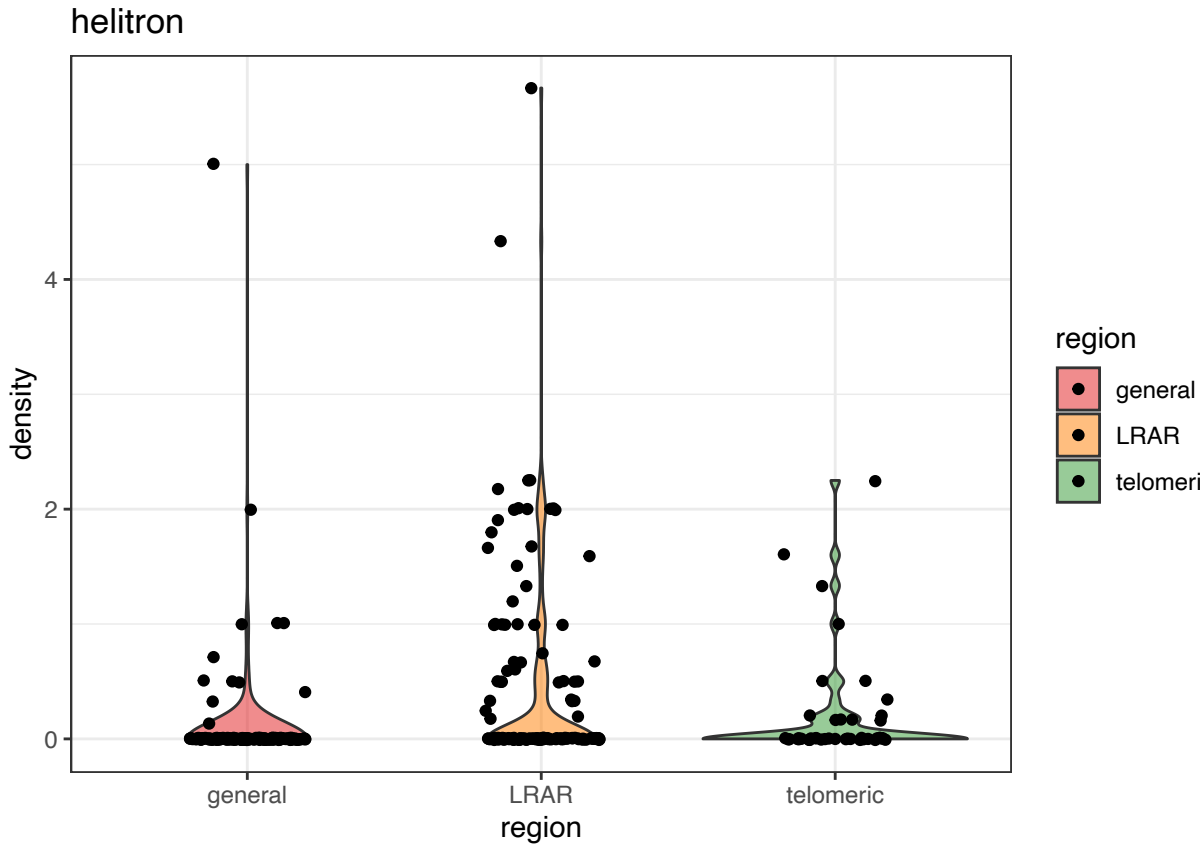


3.7.0.5 dna1

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="dna1","density"],ranges_results[ranges_resu
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "dna1", "density"] and ranges_results[ranges_results$
##
##           general LRAR
## LRAR      1.5e-06 -
## telomeric 0.014   0.162
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="helitron",],aes(x=region,y=density,fill=region,z=feature)
```

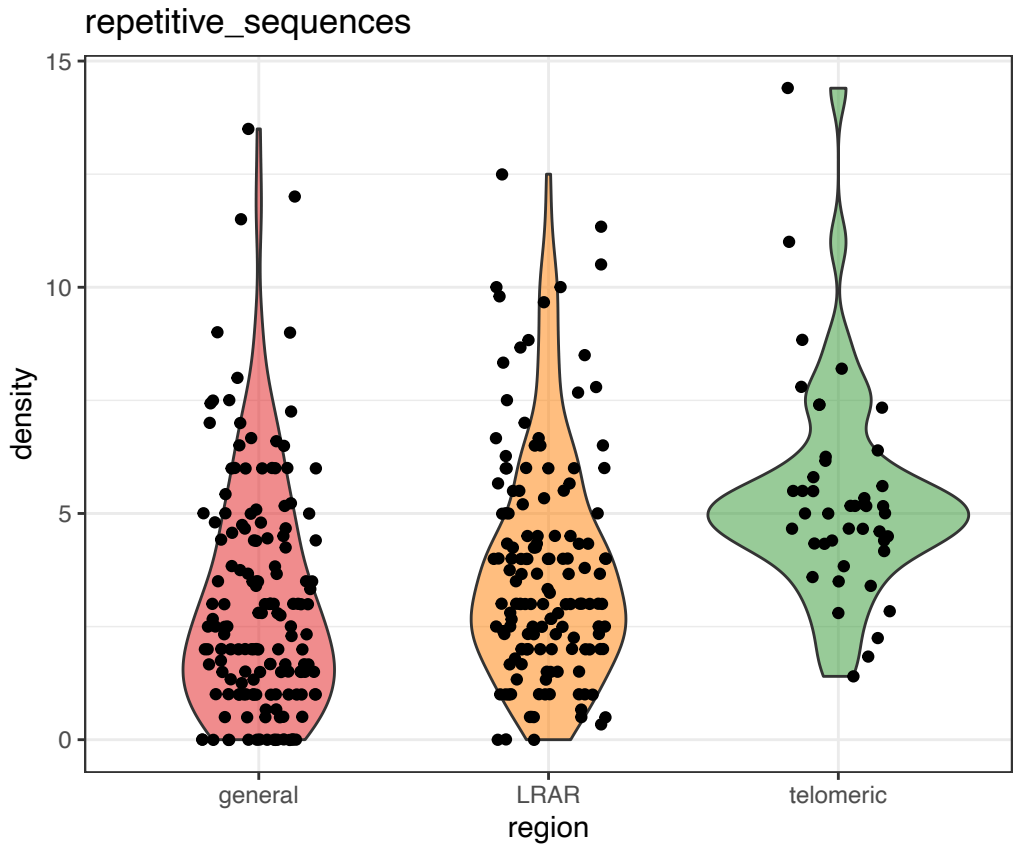


3.7.0.6 helitron

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="helitron","density"],ranges_results[ranges
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "helitron", "density"] and ranges_results[ranges_resu
##
##           general LRAR
## LRAR      2.5e-07 -
## telomeric 0.00033 0.38554
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="repetitive_sequences",],aes(x=region,y=density,fill=regi
```



3.7.0.7 repetitive_sequences

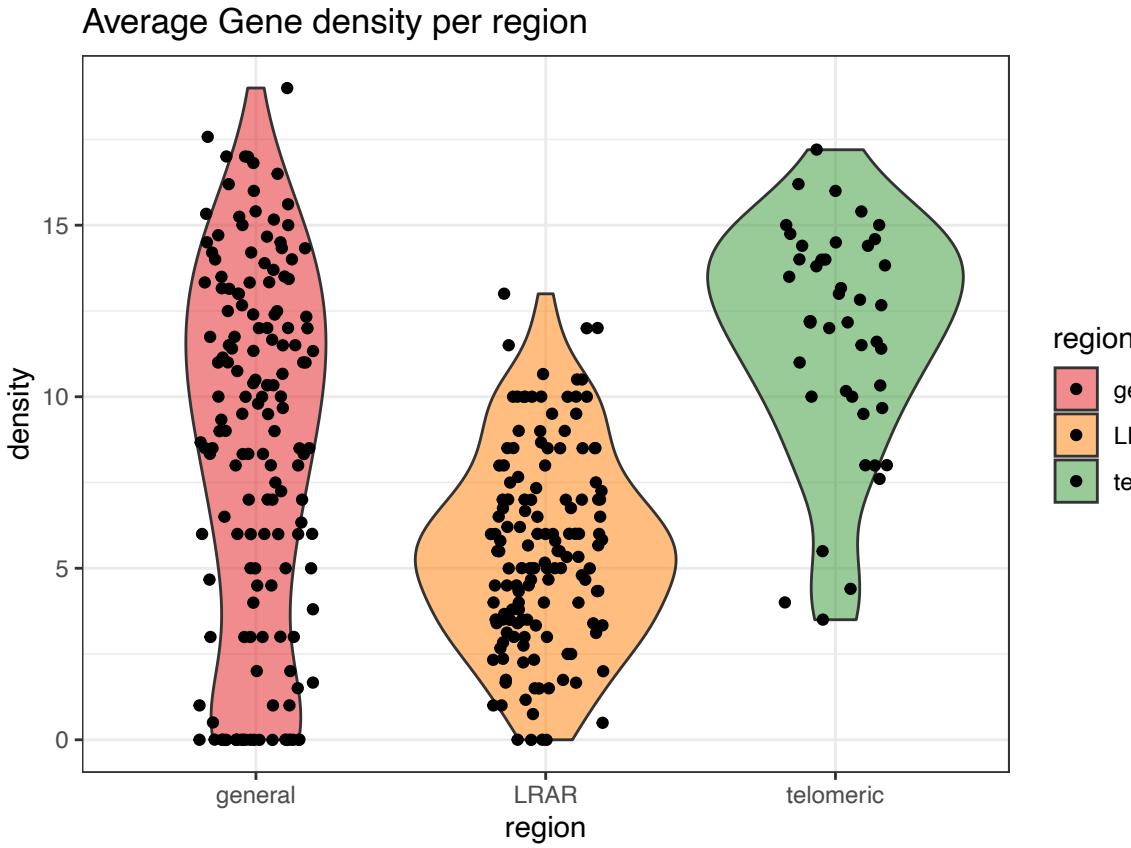
```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="repetitive_sequences","density"],ranges_res
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "repetitive_sequences", "density"] and ranges_results
##
##           general LRAR
## LRAR      0.0036  -
## telomeric 4.7e-08 1.9e-05
##
## P value adjustment method: BH
```

4

5

```
ggplot(ranges_results[ranges_results$feature=="gene_density",],aes(x=region,y=density,fill=region,z=feat
```

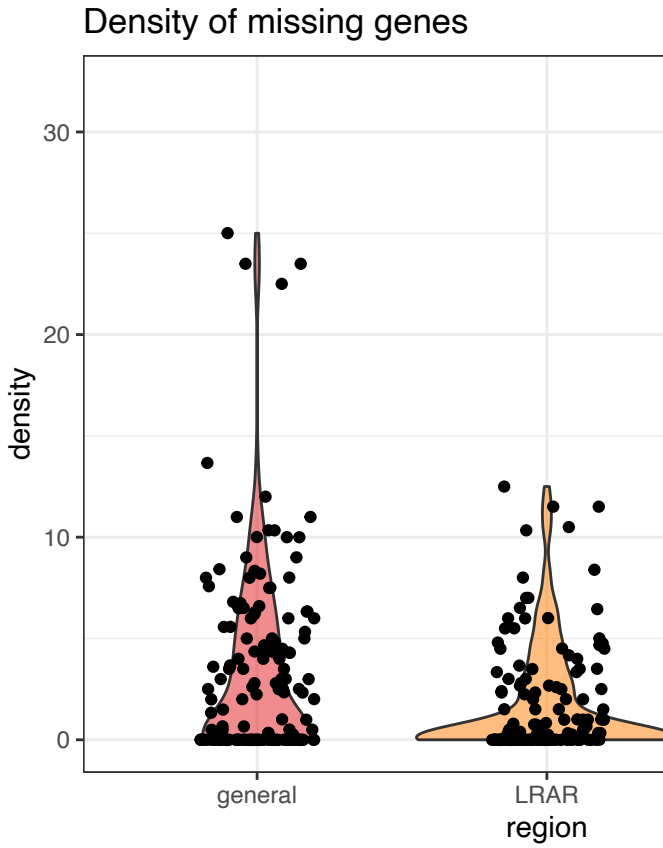


5.0.0.1 Gene density

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="gene_density","density"],ranges_results[ra
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "gene_density", "density"] and ranges_results[ranges_
##
##           general LRAR
## LRAR      1.1e-07 -
## telomeric 0.00024 8.9e-16
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="lost_genes",],aes(x=region,y=density,fill=region,z=featu
```

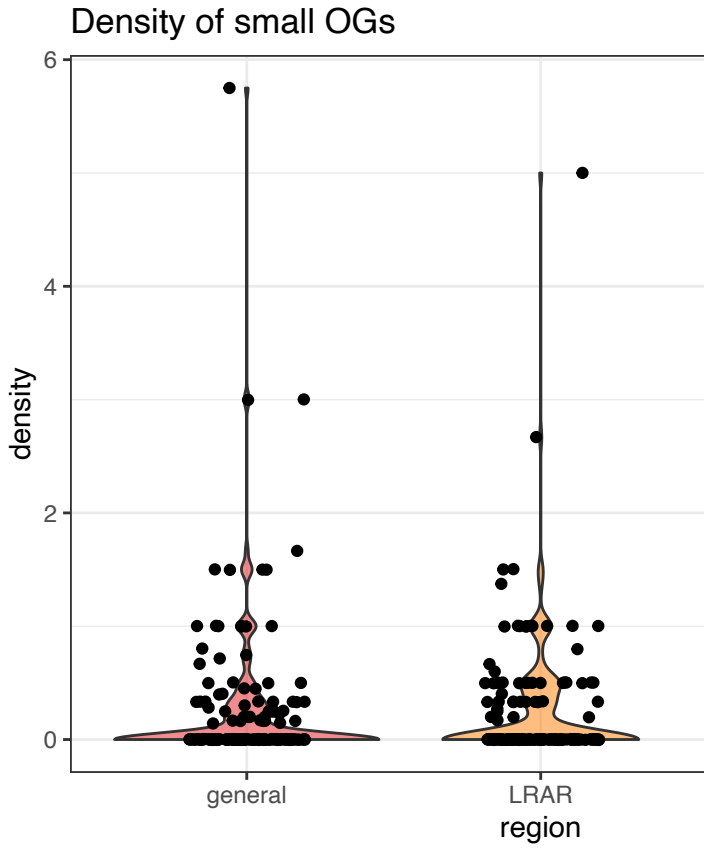


5.0.0.2 Loss of synteny, density of putative missing genes

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="lost_genes","density"],ranges_results[rang
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "lost_genes", "density"] and ranges_results[ranges_r
##
##           general LRAR
## LRAR      0.00764 -
## telomeric 0.24775 0.00099
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="small_ogs",],aes(x=region,y=density,fill=region,z=featur
```

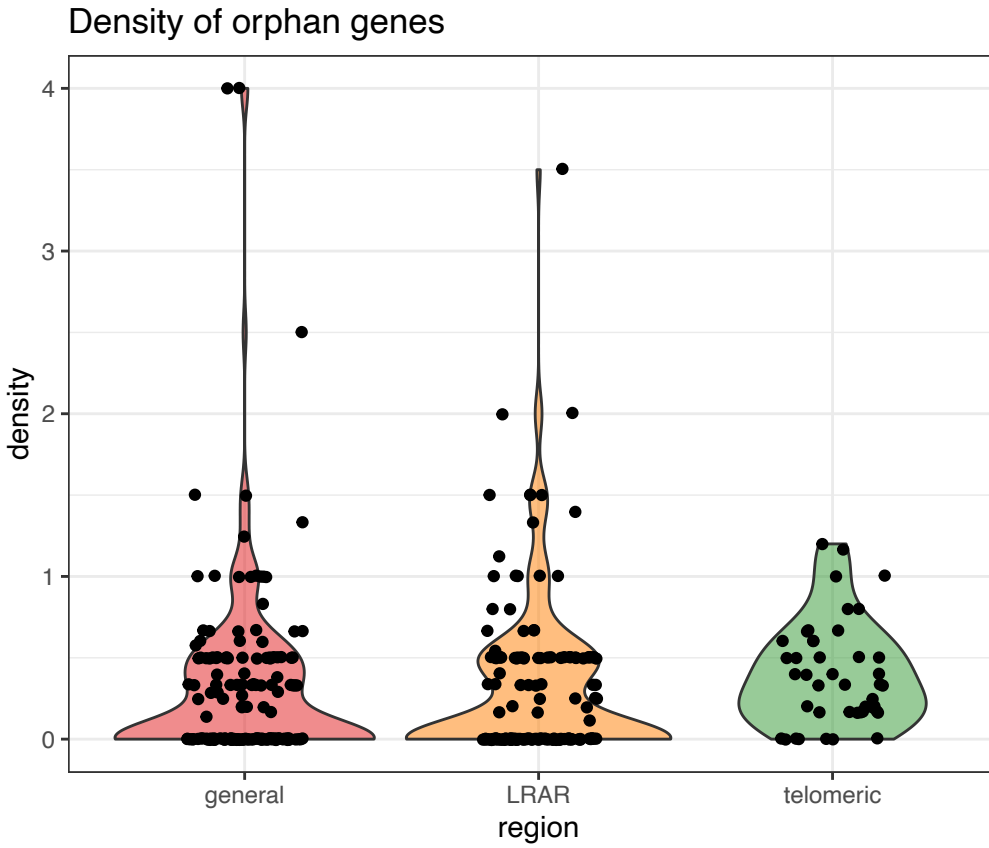


5.0.0.3 Small Orthogroups, with three samples or less

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="small_ogs","density"],ranges_results[ranges
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "small_ogs", "density"] and ranges_results[ranges_res
##
##           general LRAR
## LRAR      0.67    -
## telomeric 2.5e-09 6.3e-08
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="orphan",],aes(x=region,y=density,fill=region,z=feature))+
```

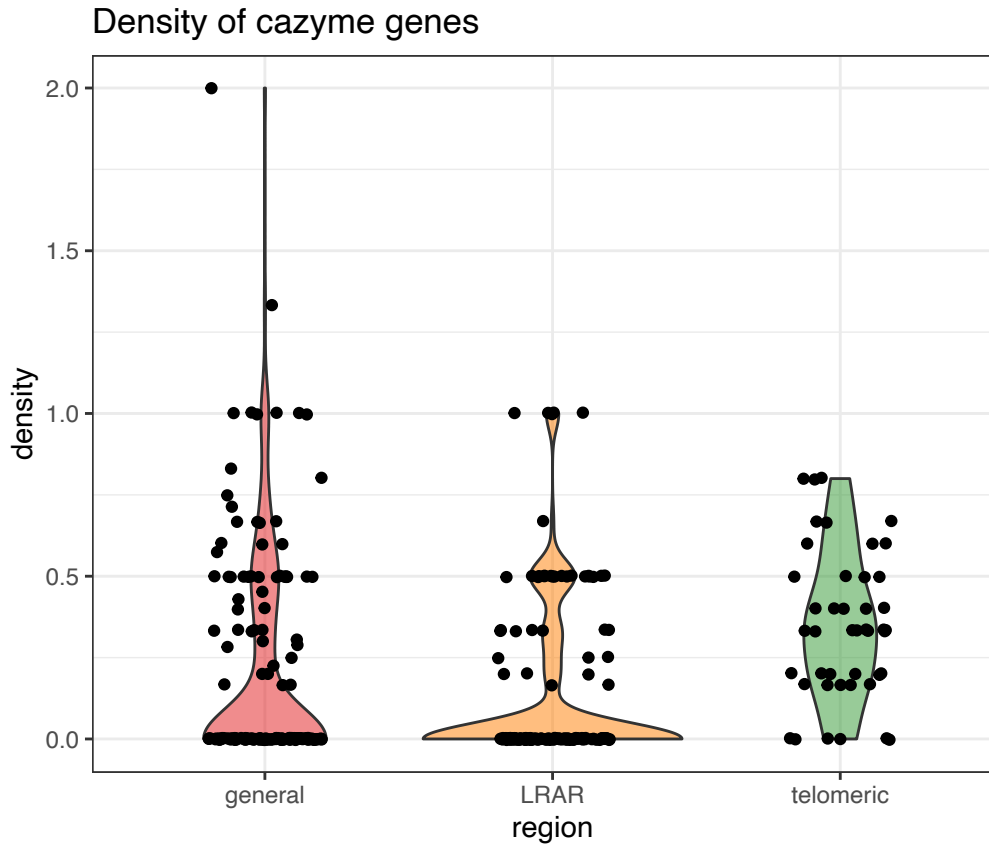


5.0.0.4 Density of orphan genes

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="orphan","density"],ranges_results[ranges_r
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "orphan", "density"] and ranges_results[ranges_resul
##
##           general LRAR
## LRAR      0.9965  -
## telomeric 0.0083  0.0083
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="cazymes",],aes(x=region,y=density,fill=region,z=feature)
```

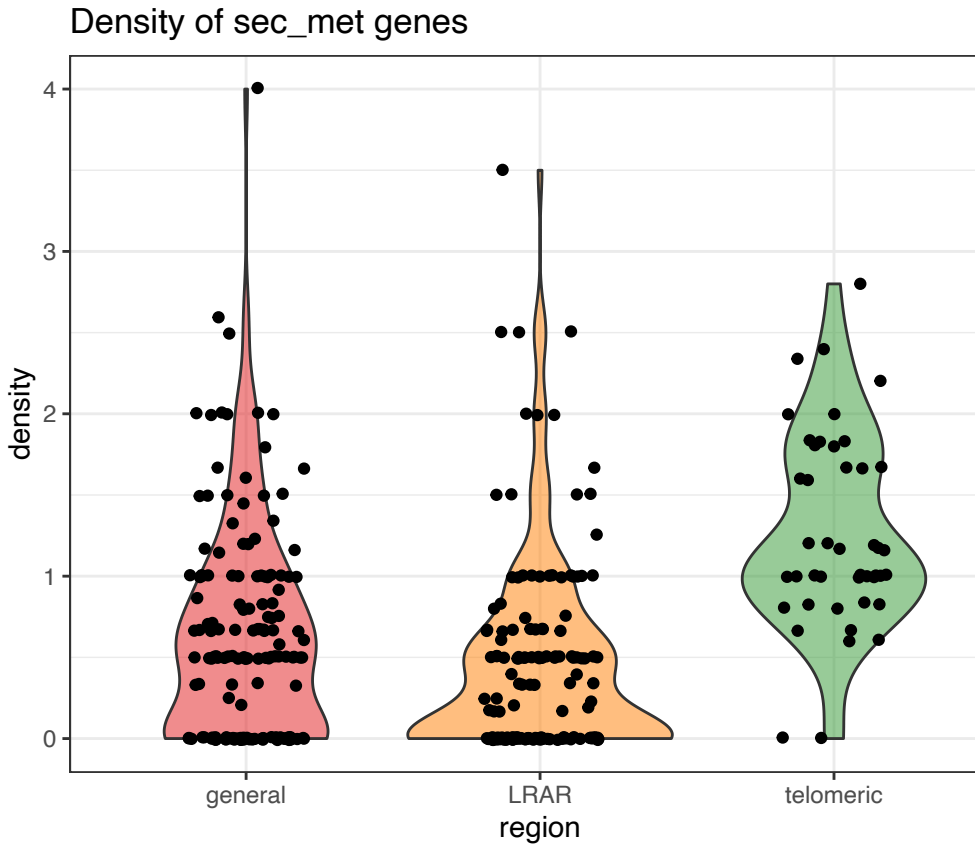


5.0.0.5 Density of cazyme genes

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="cazyme","density"],ranges_results[ranges_r
```

```
##
##  Pairwise comparisons using
##
## data:  ranges_results[ranges_results$feature == "cazyme", "density"] and ranges_results[ranges_resul
##
## <0 x 0 matrix>
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="sec_met",],aes(x=region,y=density,fill=region,z=feature)
```

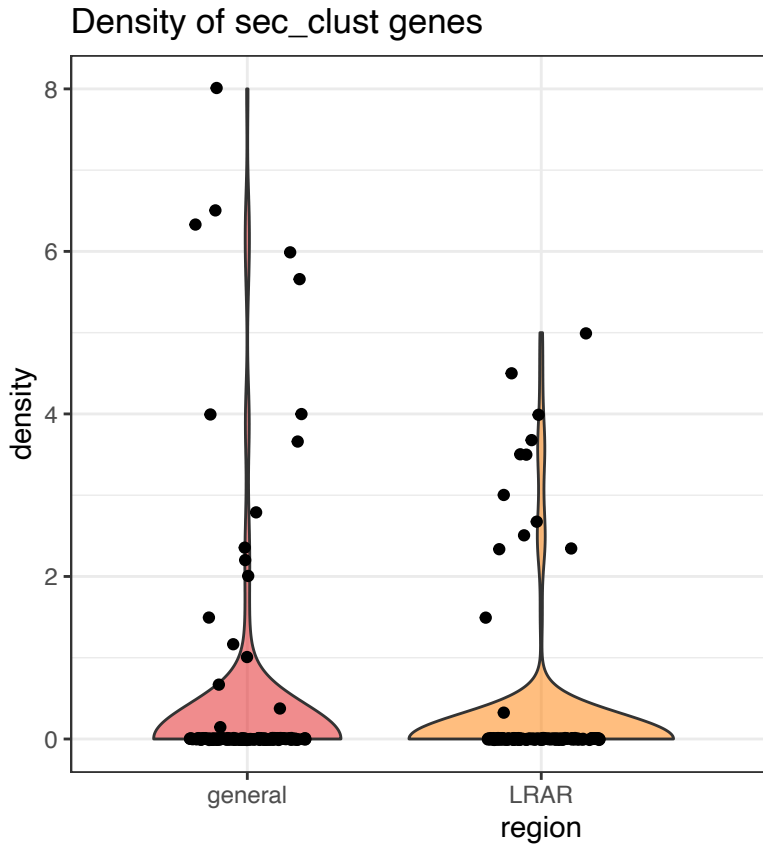


5.0.0.6 Density of secreted genes

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="sec_met","density"],ranges_results[ranges_
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "sec_met", "density"] and ranges_results[ranges_resul
##
##           general LRAR
## LRAR      0.011   -
## telomeric 2.2e-09 7.0e-13
##
## P value adjustment method: BH
```

```
ggplot(ranges_results[ranges_results$feature=="sec_clust",],aes(x=region,y=density,fill=region,z=feature
```



5.0.0.7 Density of secondary metabolism clusters

```
pairwise.wilcox.test(ranges_results[ranges_results$feature=="sec_clust","density"],ranges_results[ranges
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results[ranges_results$feature == "sec_clust", "density"] and ranges_results[ranges_res
##
##           general LRAR
## LRAR      0.47    -
## telomeric 1.4e-08 3.6e-09
##
## P value adjustment method: BH
```

5.1 Phylogenetic signal across the genome

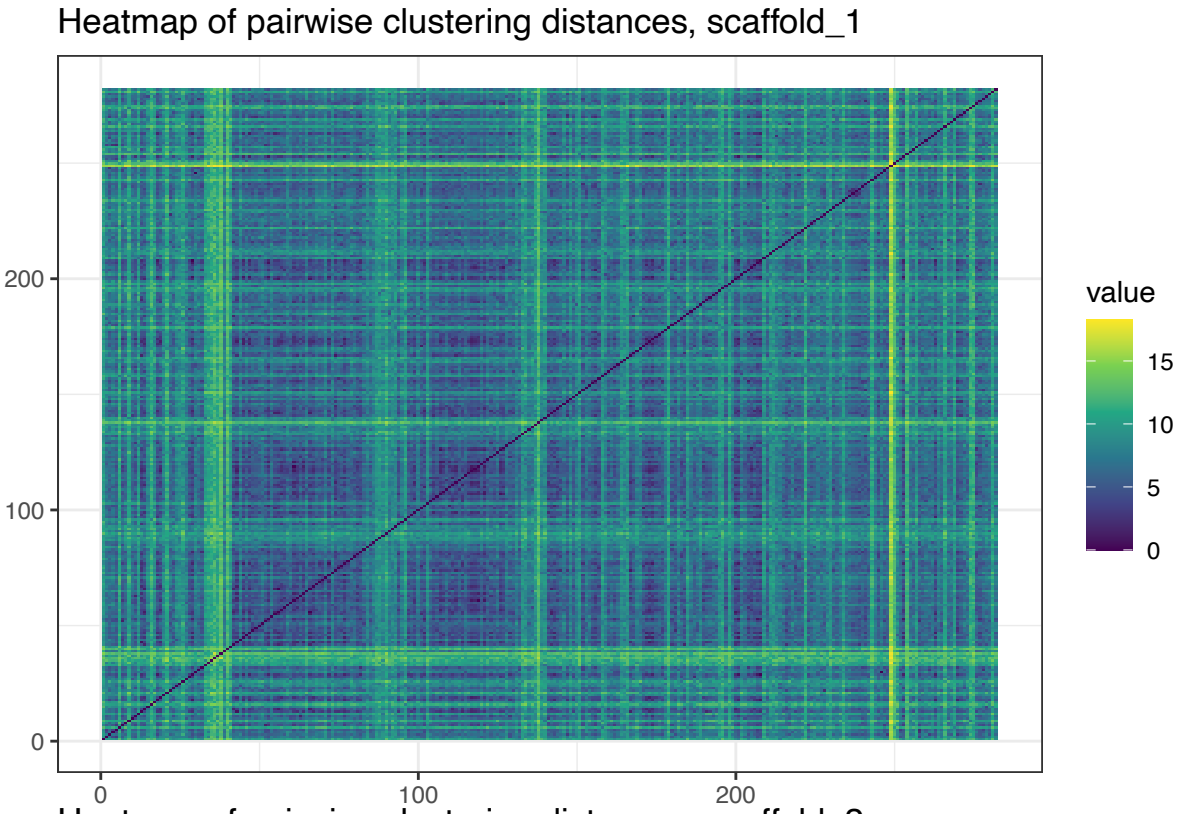
%### Plotting all %{r} %culo<-loci_trees[!sapply(loci_trees,FUN=function(x){is.monophyletic(unroot(x),c("
%for (i in 1:length(culo)) %{ % p<-ggtree(culo[[i]]) + geom_tippoint(color=as.numeric(factor(mat[culo[
+ %geom_tiplab(align=FALSE, linetype="dotted", linesize=.3, offset=0.1) %print(p) %} %

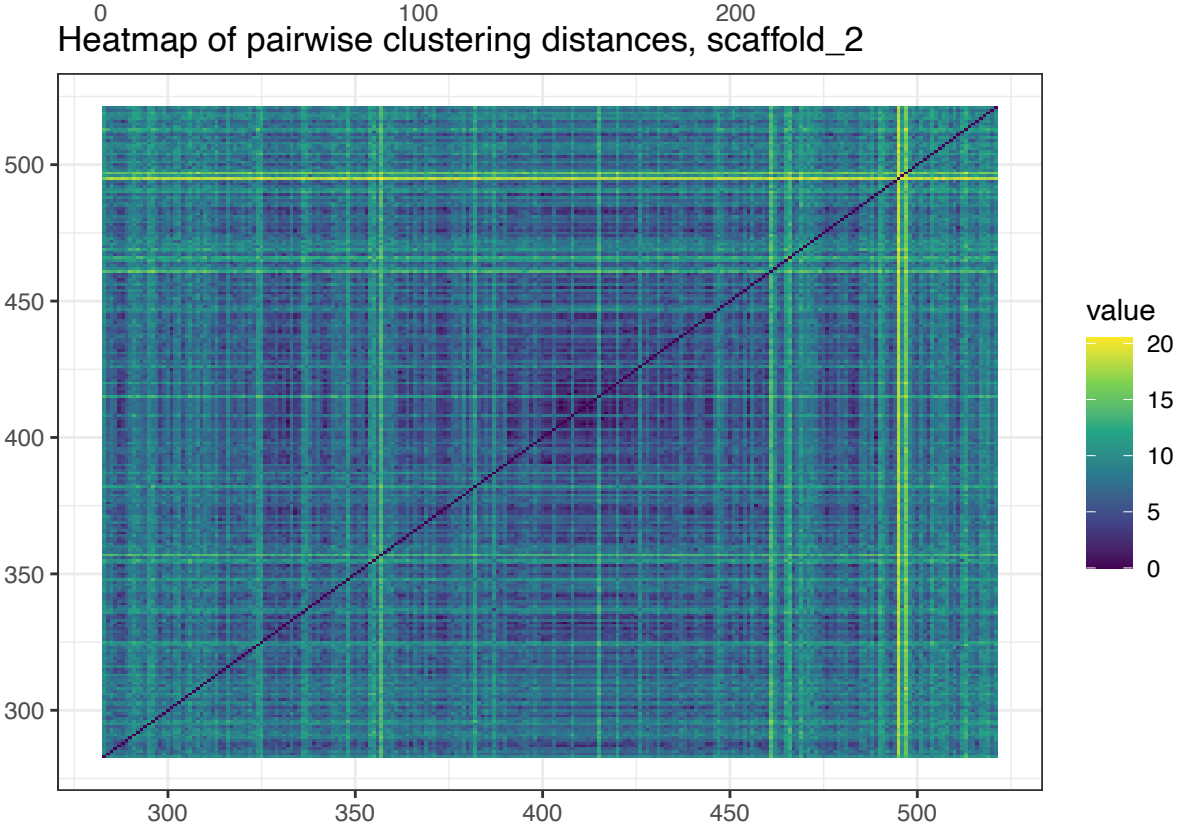
```
het<-c("Pyrenodesmiaerodens_003026-T1",
"Pyrenodesmiaerodens_008956-T1",
"Pyrenodesmiaerodens_002843-T1",
"Pyrenodesmiaerodens_008195-T1",
"Pyrenodesmiaerodens_006652-T1",
"Pyrenodesmiaerodens_007209-T1",
"Pyrenodesmiaerodens_007178-T1",
"Pyrenodesmiaerodens_001939-T1",
"Pyrenodesmiaerodens_006912-T1",
```

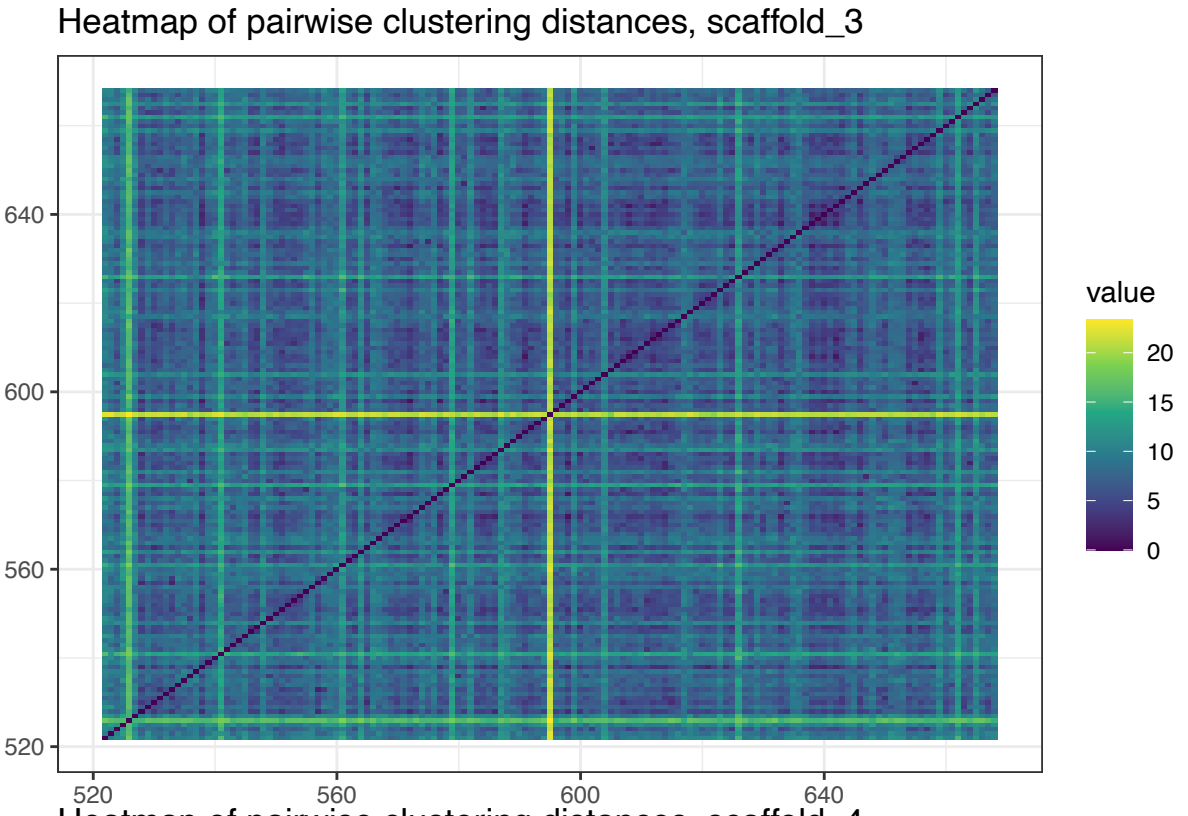
```
"Pyrenodesmiaerodens_001303-T1",
"Pyrenodesmiaerodens_008514-T1",
"Pyrenodesmiaerodens_007614-T1",
"Pyrenodesmiaerodens_005425-T1",
"Pyrenodesmiaerodens_003710-T1",
"Pyrenodesmiaerodens_006911-T1",
"Pyrenodesmiaerodens_003701-T1",
"Pyrenodesmiaerodens_009360-T1",
"Pyrenodesmiaerodens_001394-T1",
"Pyrenodesmiaerodens_006881-T1",
"Pyrenodesmiaerodens_004495-T1",
"Pyrenodesmiaerodens_004888-T1",
"Pyrenodesmiaerodens_006904-T1",
"Pyrenodesmiaerodens_006707-T1",
"Pyrenodesmiaerodens_003667-T1",
"Pyrenodesmiaerodens_007504-T1",
"Pyrenodesmiaerodens_007081-T1",
"Pyrenodesmiaerodens_003230-T1",
"Pyrenodesmiaerodens_000215-T1",
"Pyrenodesmiaerodens_004826-T1",
"Pyrenodesmiaerodens_008820-T1",
"Pyrenodesmiaerodens_000990-T1",
"Pyrenodesmiaerodens_004497-T1",
"Pyrenodesmiaerodens_001877-T1",
"Pyrenodesmiaerodens_008462-T1",
"Pyrenodesmiaerodens_004488-T1",
"Pyrenodesmiaerodens_008467-T1",
"Pyrenodesmiaerodens_004430-T1",
"Pyrenodesmiaerodens_001833-T1",
"Pyrenodesmiaerodens_007926-T1",
"Pyrenodesmiaerodens_008989-T1","Pyrenodesmiaerodens_008241-T1")
het<-genes2[gsub("-T1","",het),]
```

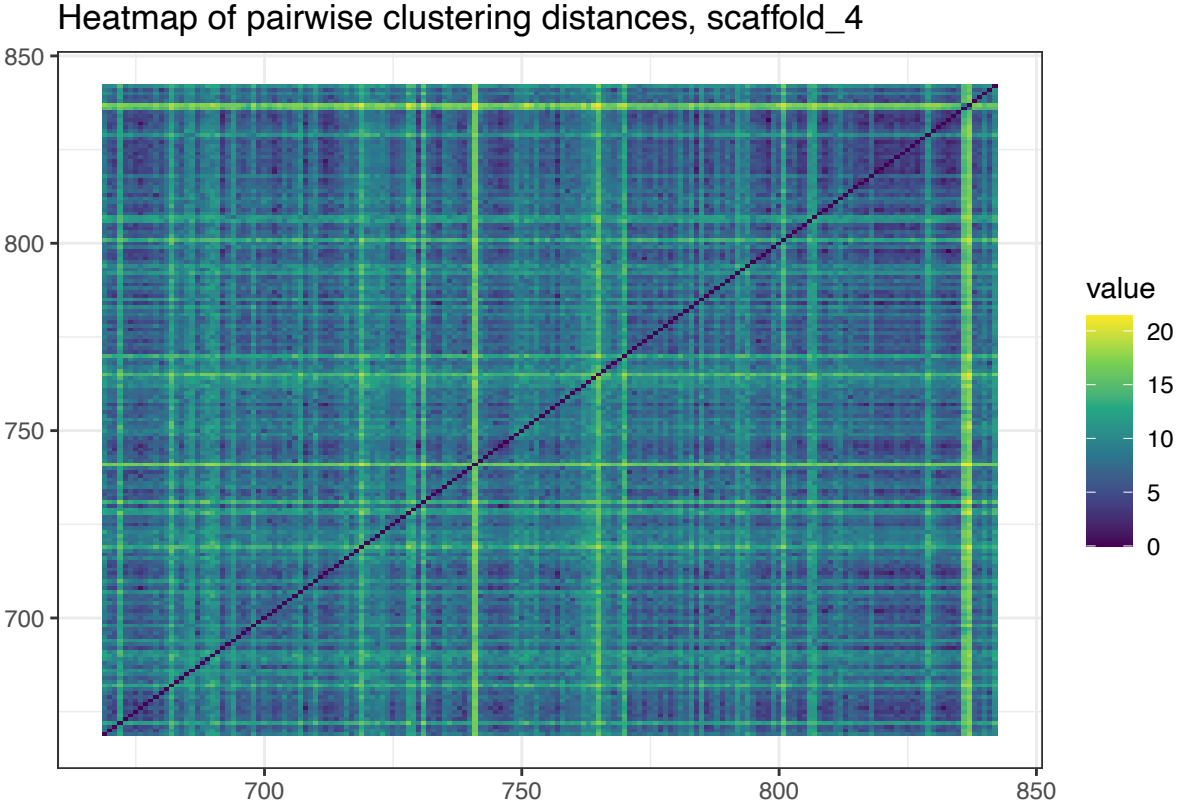
5.2 Plot pairwise heatmaps

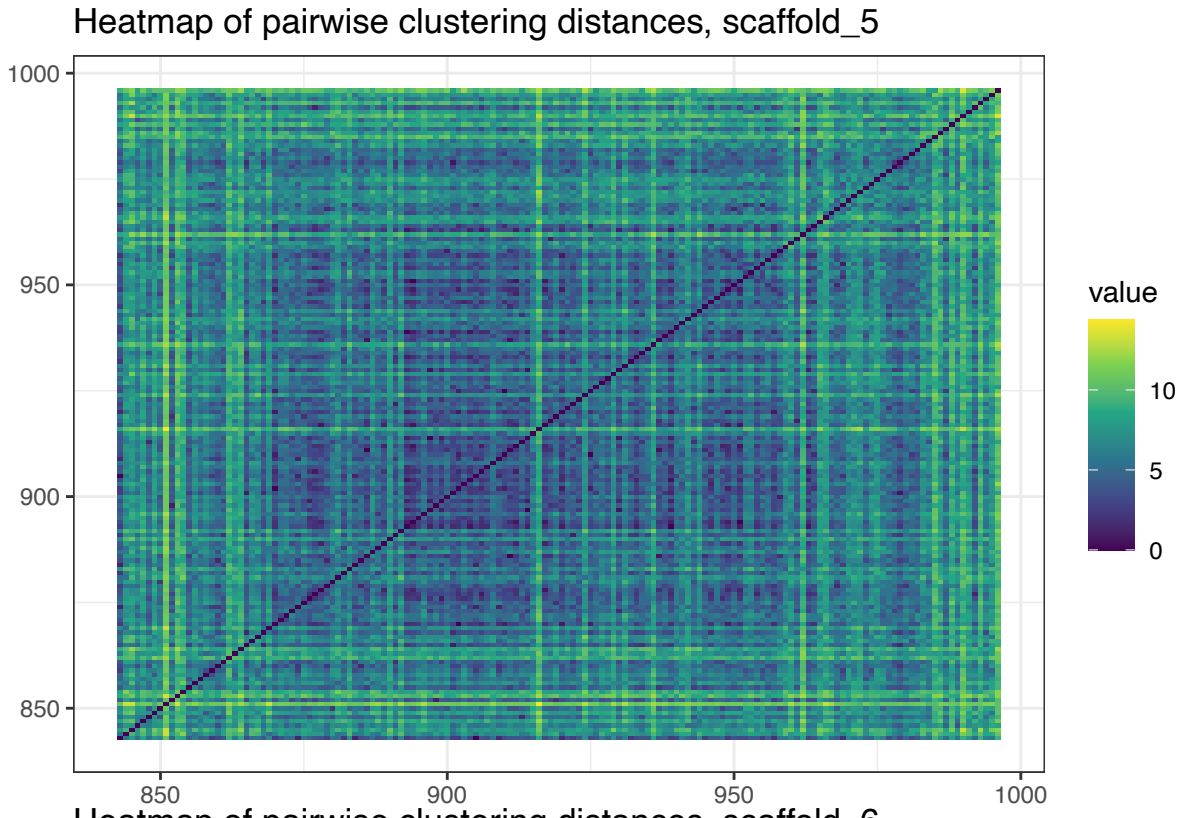
```
genes$scaffold<-factor(genes$scaffold)
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2))
{
    distances_foo<-data.frame(melt(as.matrix(distances)[genes$scaffold==SCAF,genes$scaffold==SCAF]))
    p<-ggplot(distances_foo, aes(x=Var1, y=Var2, fill=value) ) + geom_tile() + scale_fill_continuous(typ
    print(p)
}
```

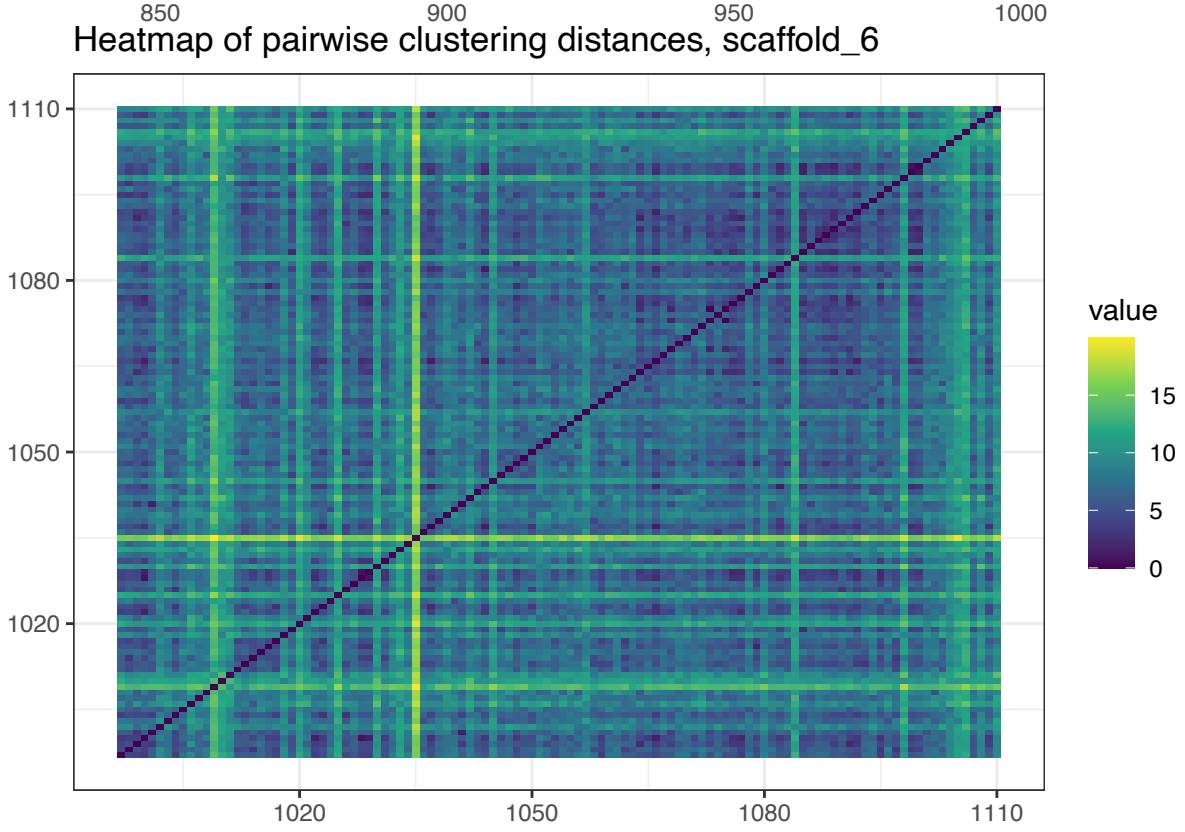

Heatmap of pairwise clustering distances, scaffold_1
200
100
0
0
100
200
value
15
10
5
0

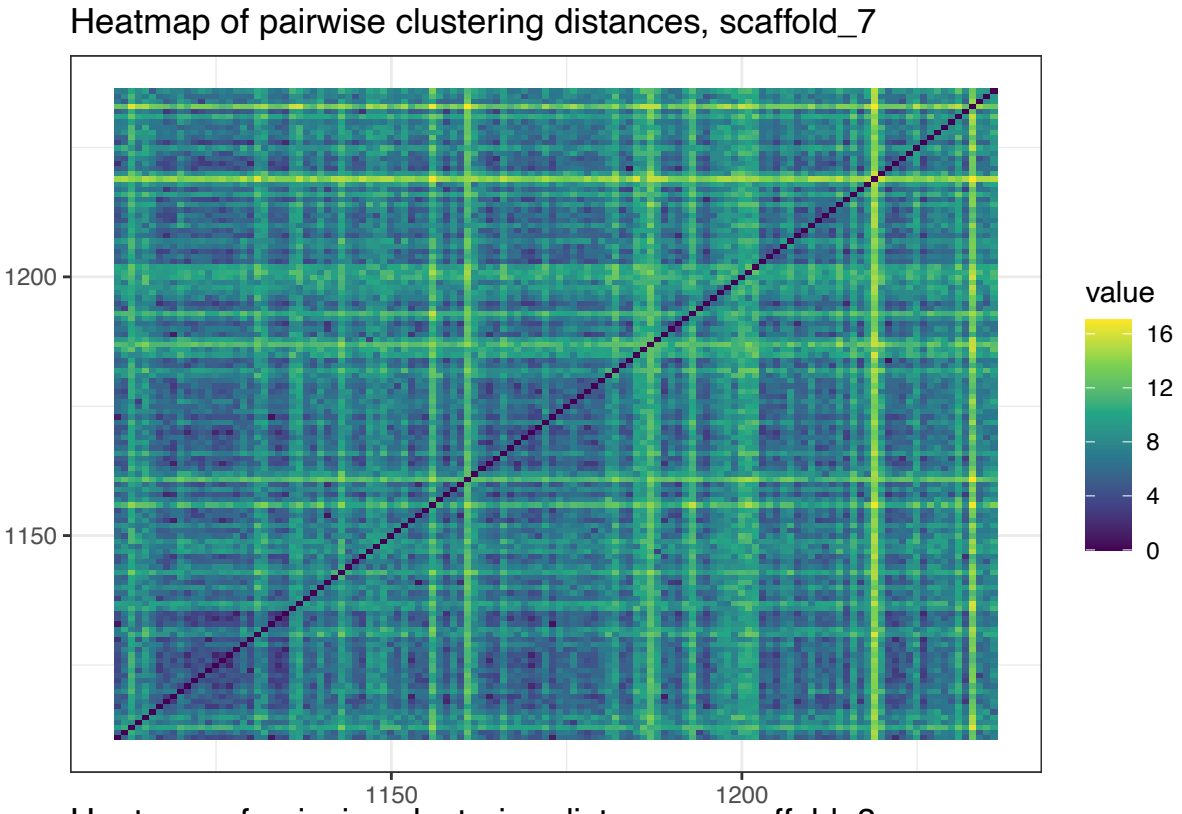

Heatmap of pairwise clustering distances, scaffold_2
500
450
400
350
300
300
350
400
450
500
value
20
15
10
5
0

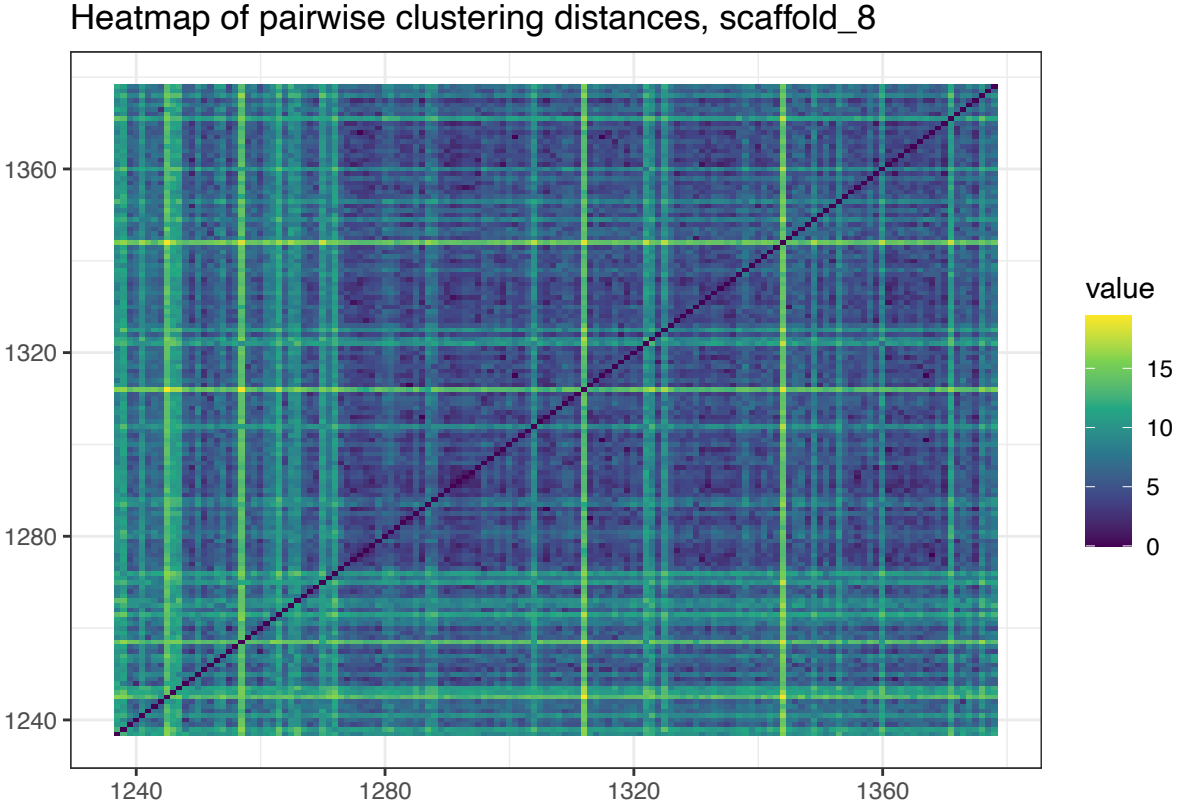

Heatmap of pairwise clustering distances, scaffold_3
640
600
560
520
520
560
600
640
value
20
15
10
5
0

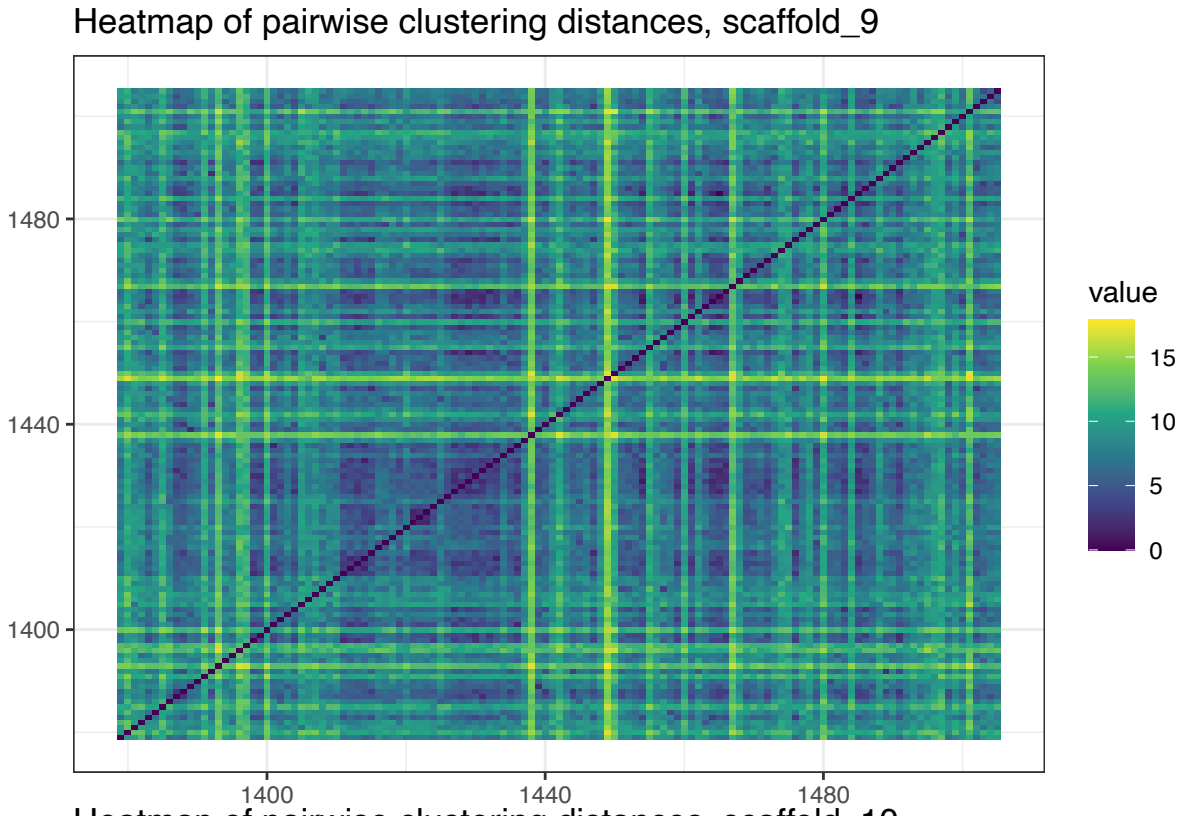

Heatmap of pairwise clustering distances, scaffold_4
850
800
750
700
700
750
800
850
value
20
15
10
5
0

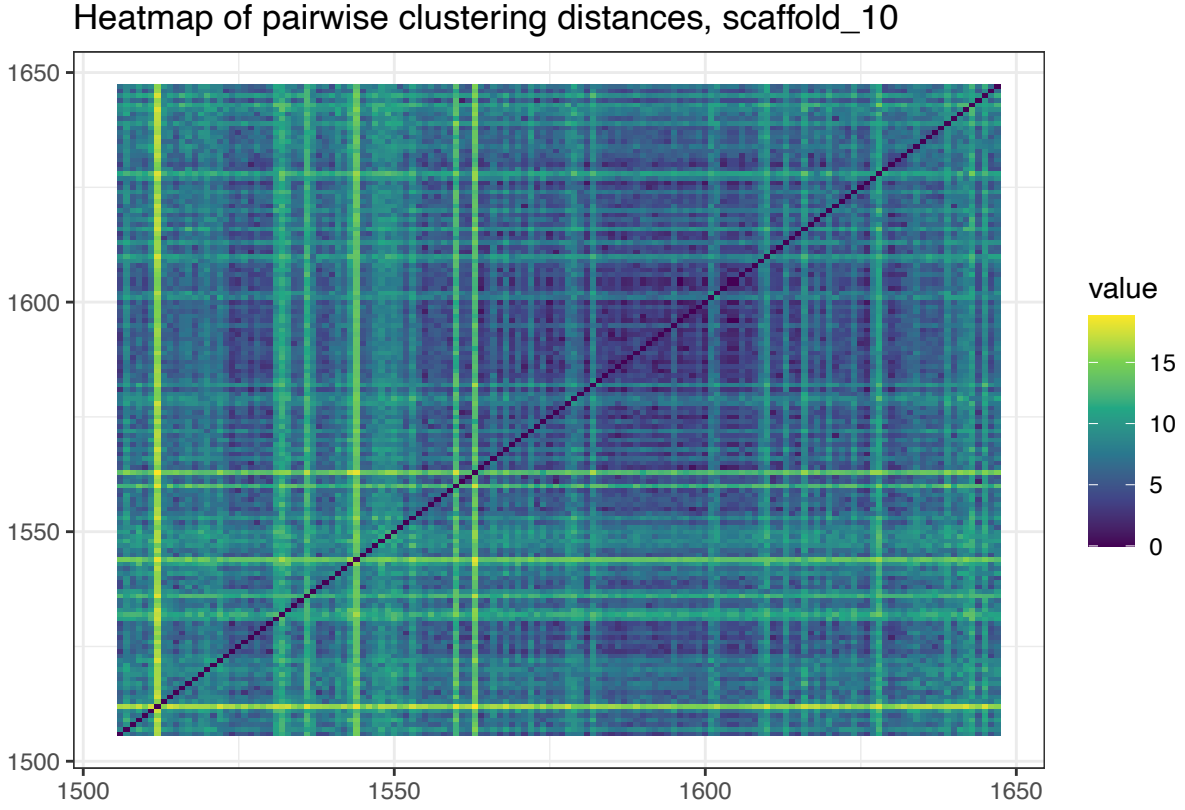

Heatmap of pairwise clustering distances, scaffold_5
1000
950
900
850
850
900
950
1000
value
10
5
0

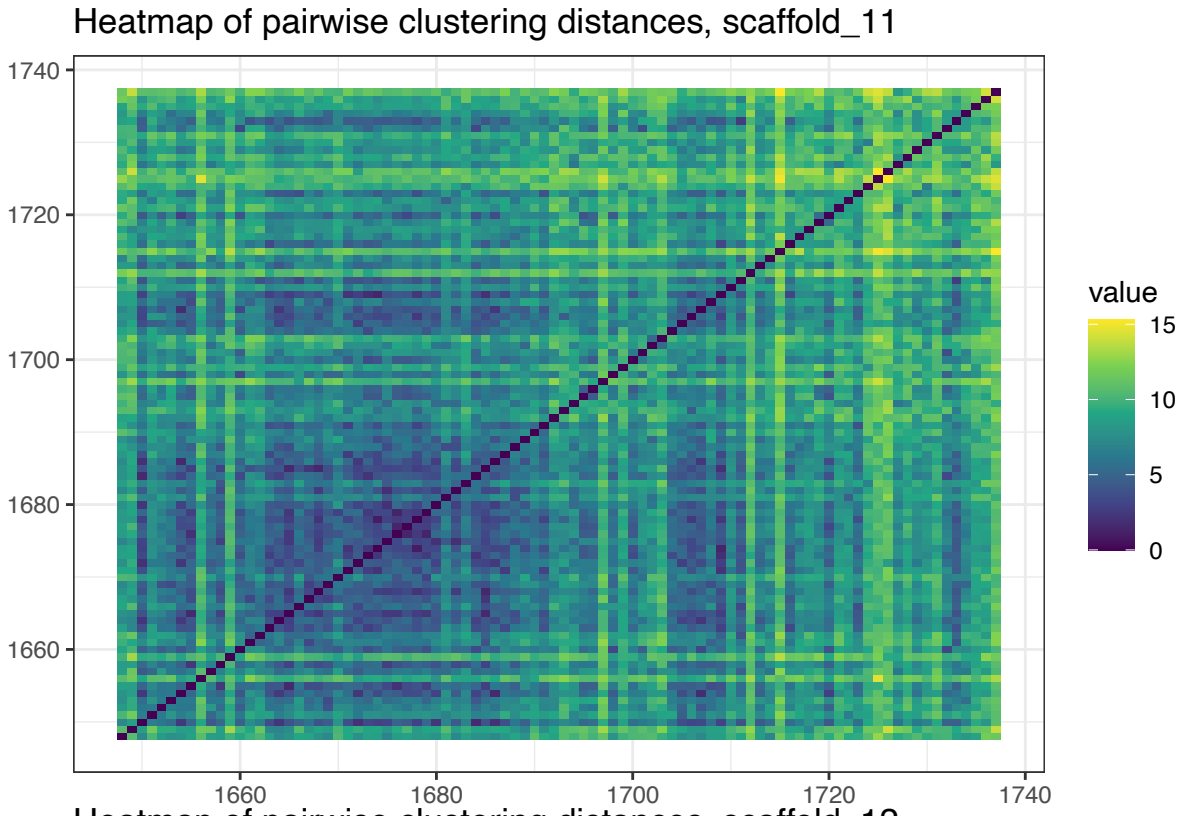

Heatmap of pairwise clustering distances, scaffold_6
1110
1080
1050
1020
1020
1050
1080
1110
value
15
10
5
0

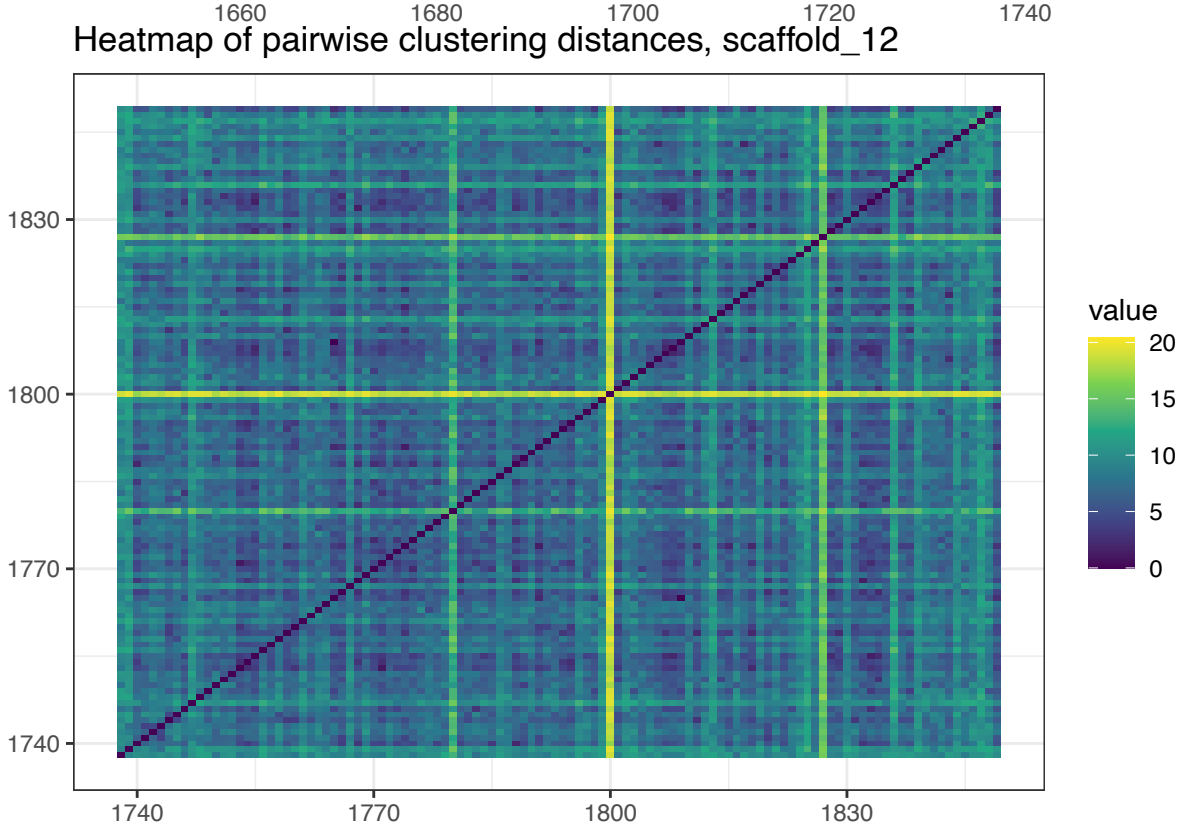

Heatmap of pairwise clustering distances, scaffold_7
1200
1150
1150
1200
value
16
12
8
4
0

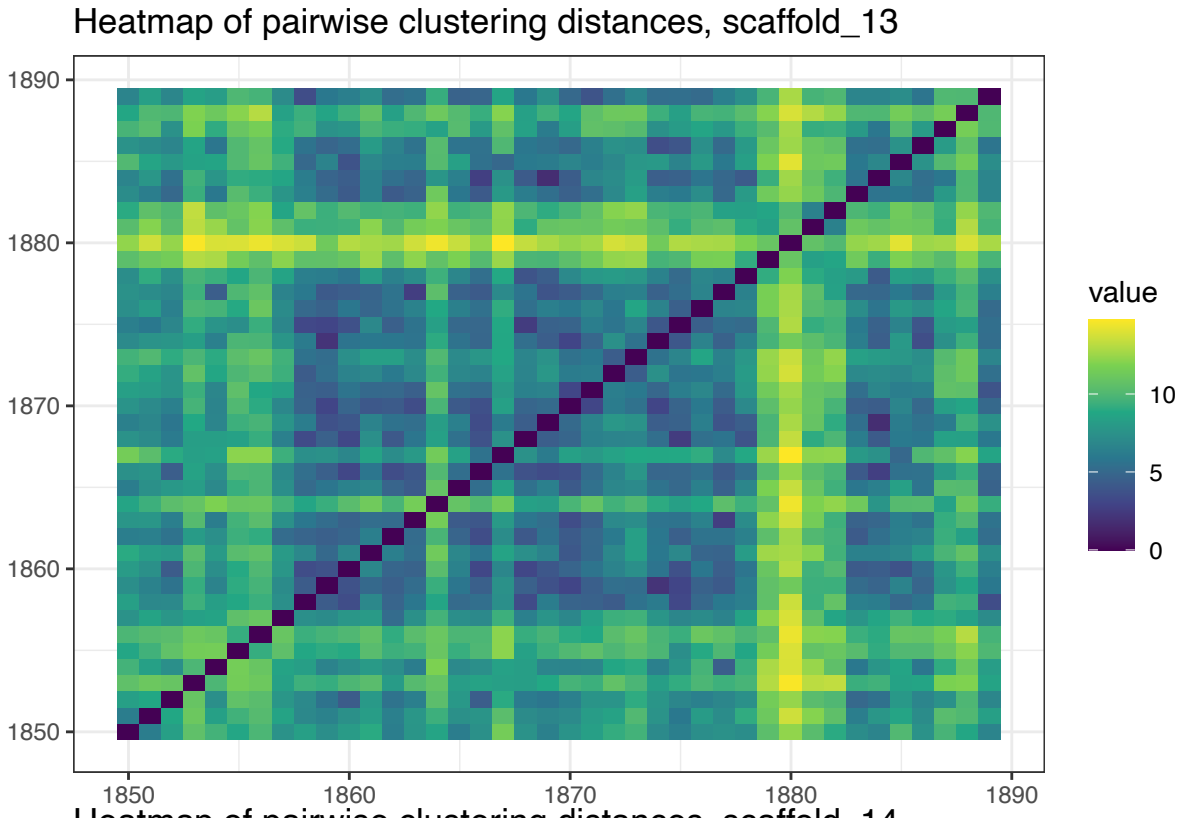

Heatmap of pairwise clustering distances, scaffold_8
1360
1320
1280
1240
1240
1280
1320
1360
value
15
10
5
0

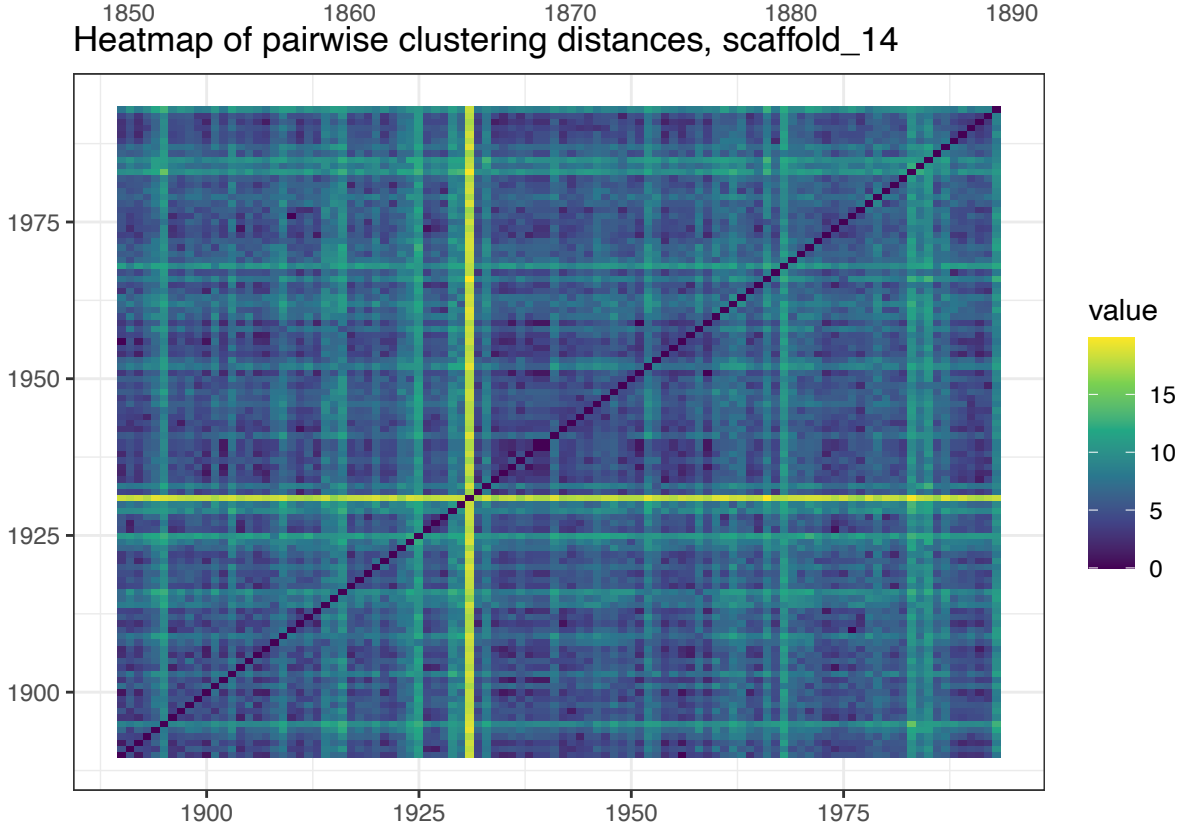

Heatmap of pairwise clustering distances, scaffold_9
1480
1440
1400
1400
1440
1480
value
15
10
5
0

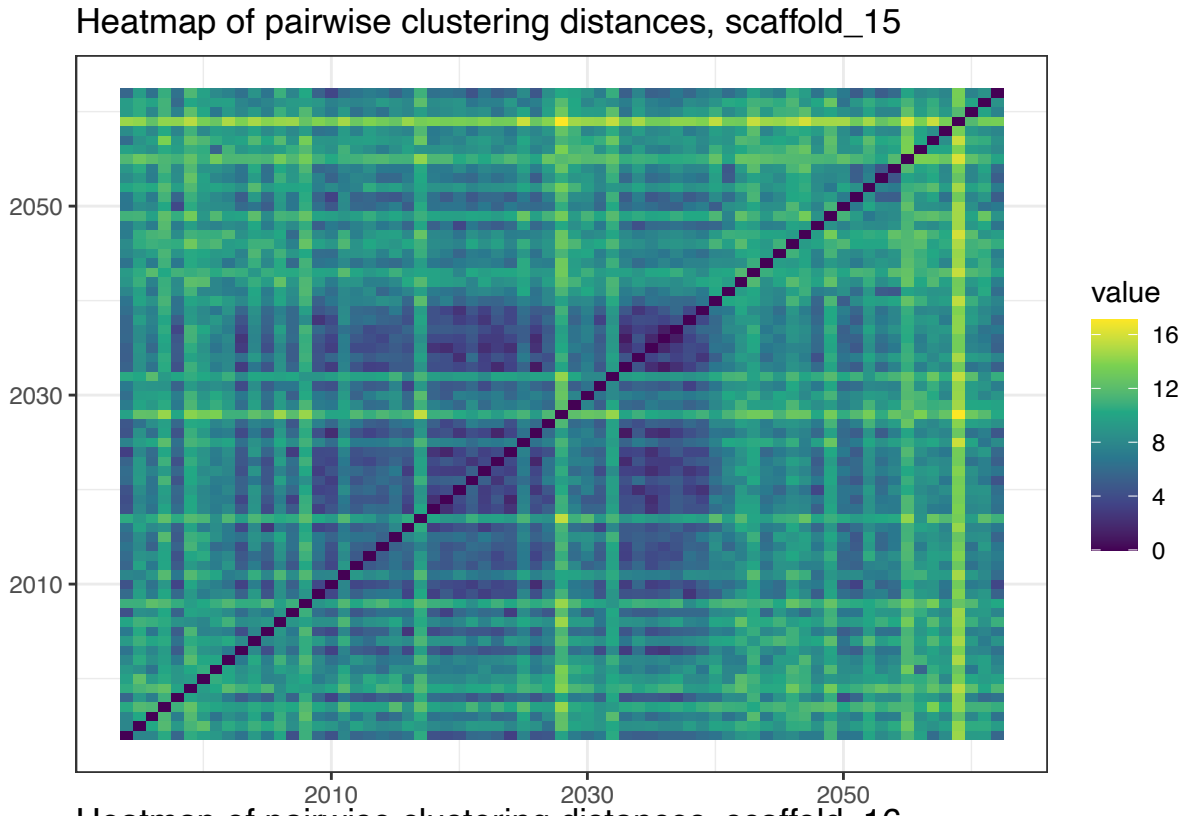

Heatmap of pairwise clustering distances, scaffold_10
1650
1600
1550
1500
1500
1550
1600
1650
value
15
10
5
0

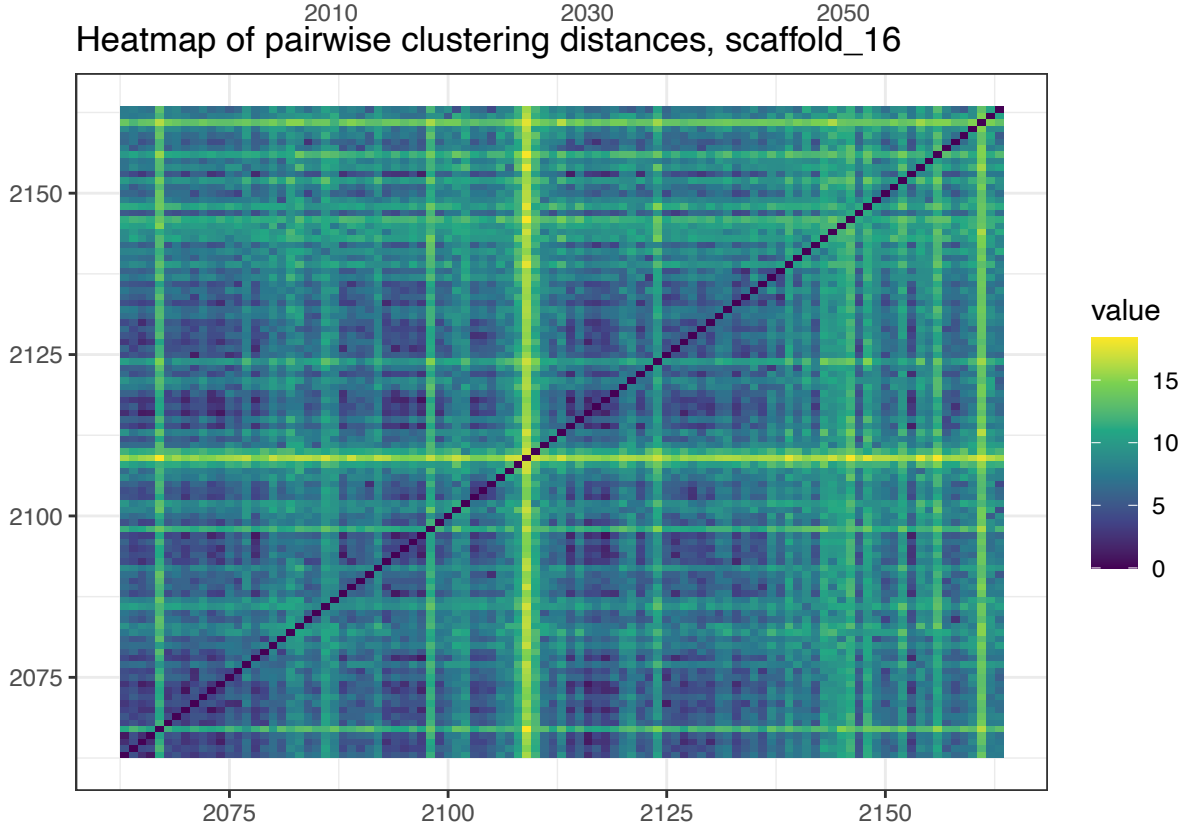

Heatmap of pairwise clustering distances, scaffold_11
1740
1720
1700
1680
1660
1660
1680
1700
1720
1740
value
15
10
5
0

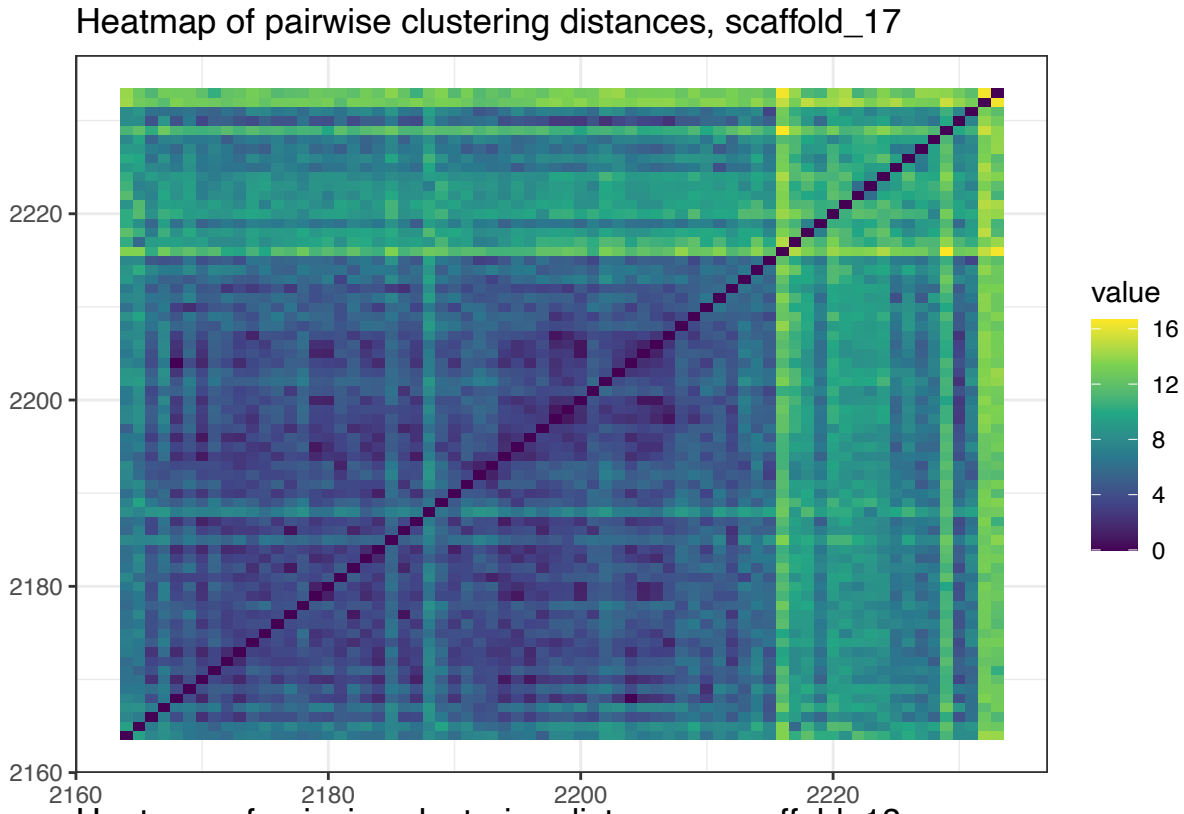

Heatmap of pairwise clustering distances, scaffold_12
1830
1800
1770
1740
1740
1770
1800
1830
value
20
15
10
5
0

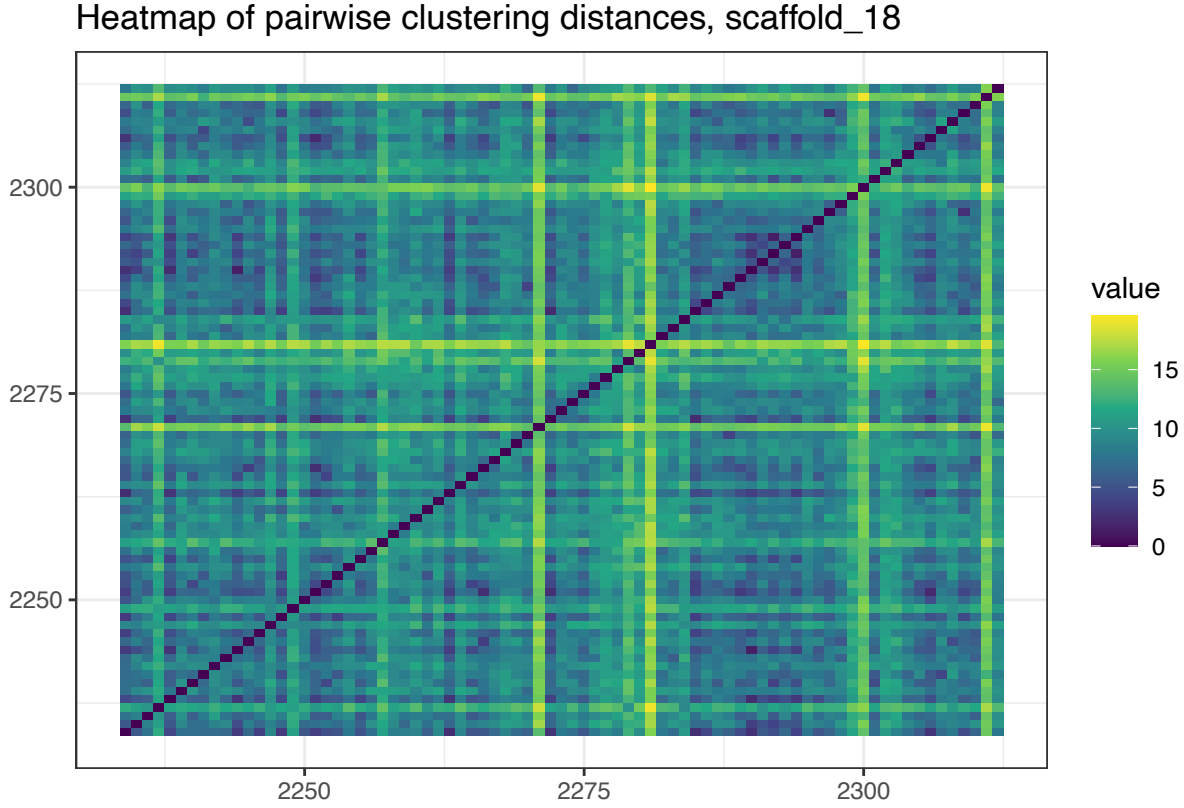

Heatmap of pairwise clustering distances, scaffold_13
1890
1880
1870
1860
1850
1850
1860
1870
1880
1890
value
10
5
0

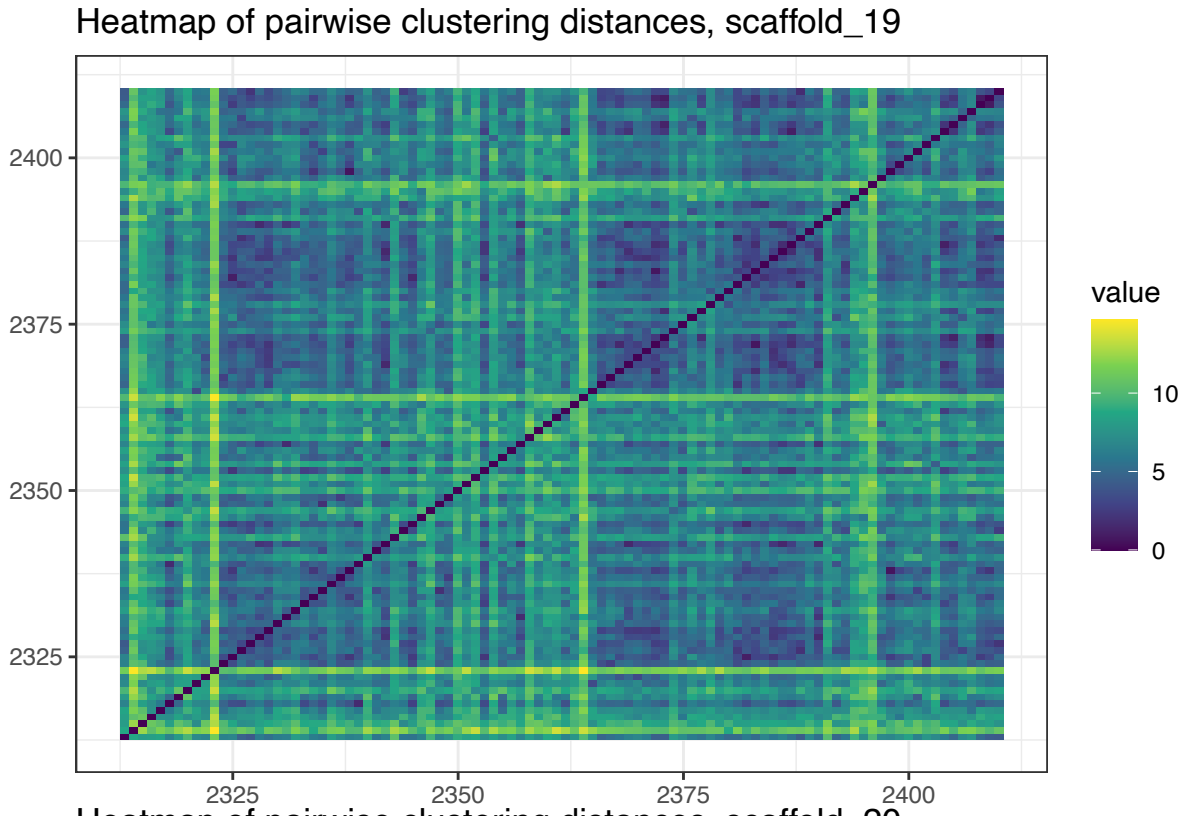

Heatmap of pairwise clustering distances, scaffold_14
1975
1950
1925
1900
1900
1925
1950
1975
value
15
10
5
0

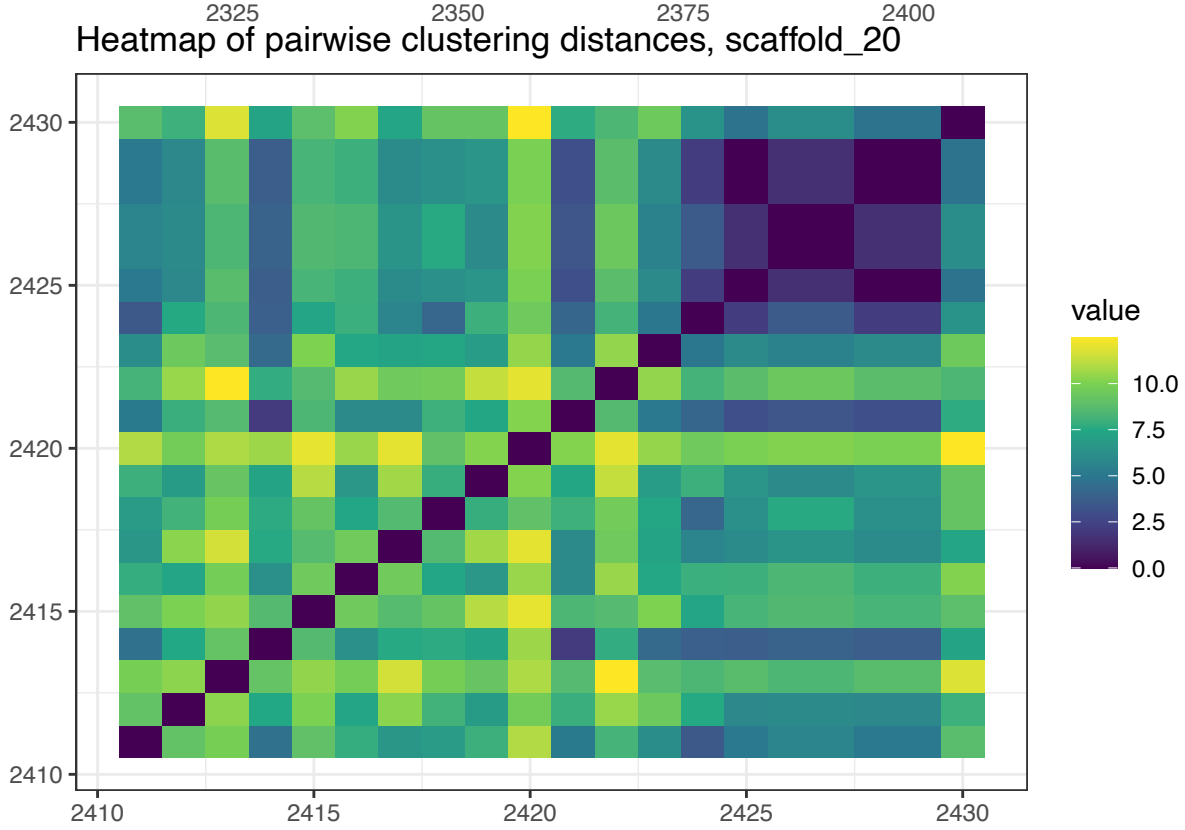

Heatmap of pairwise clustering distances, scaffold_15
2050
2030
2010
2010
2030
2050
value
16
12
8
4
0

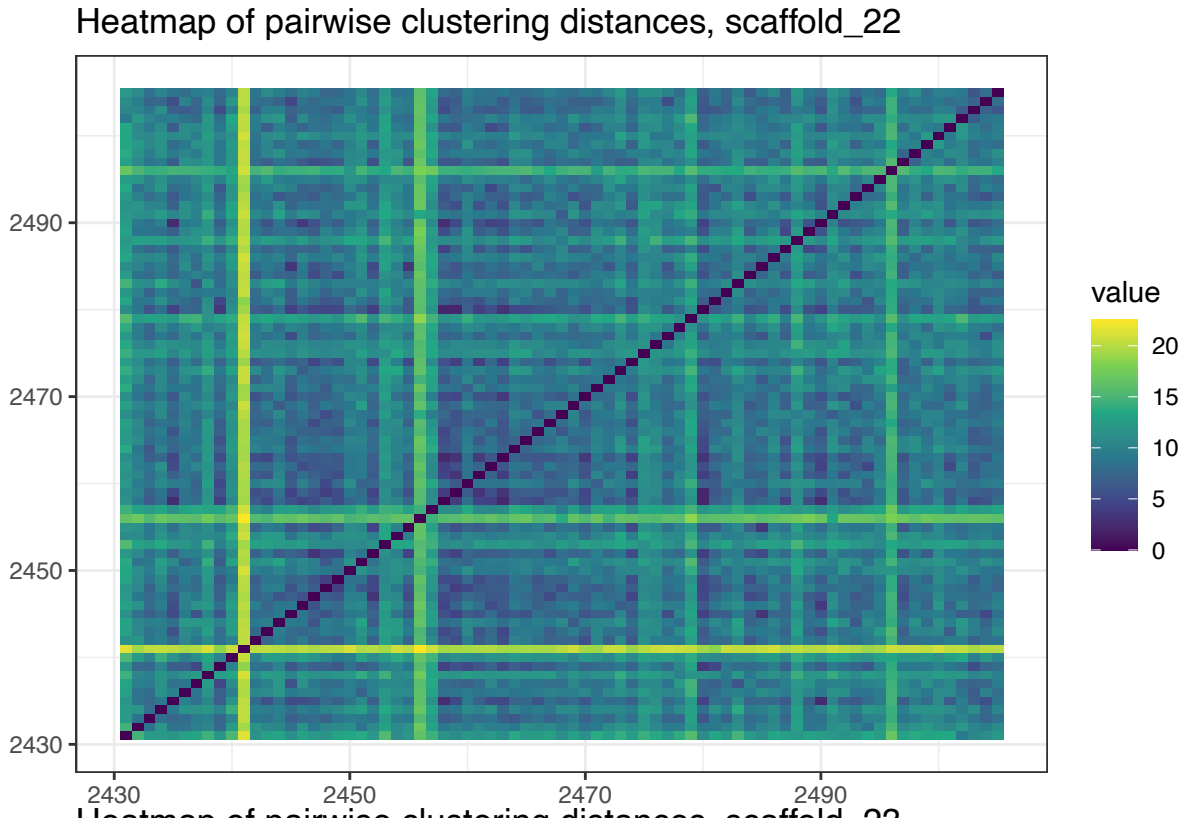

Heatmap of pairwise clustering distances, scaffold_16
2150
2125
2100
2075
2075
2100
2125
2150
value
15
10
5
0

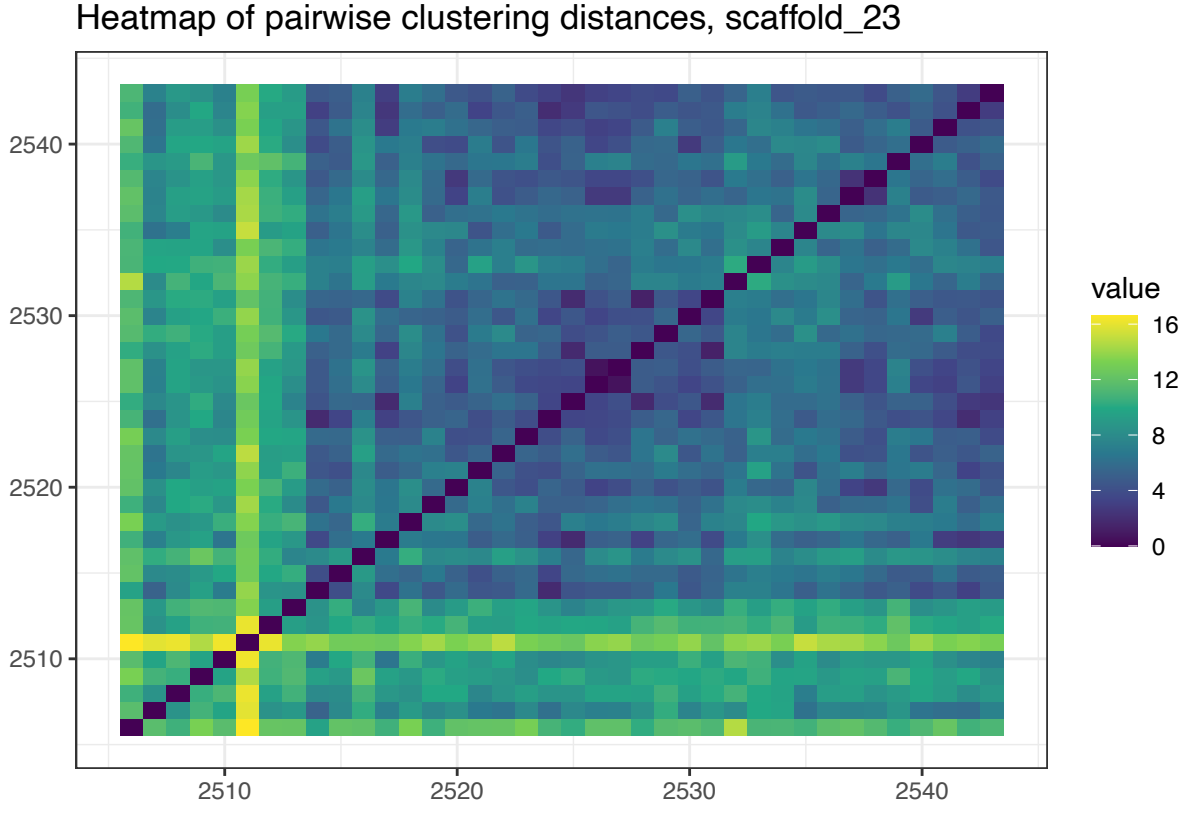

Heatmap of pairwise clustering distances, scaffold_17
2220
2200
2180
2160
2160
2180
2200
2220
value
16
12
8
4
0

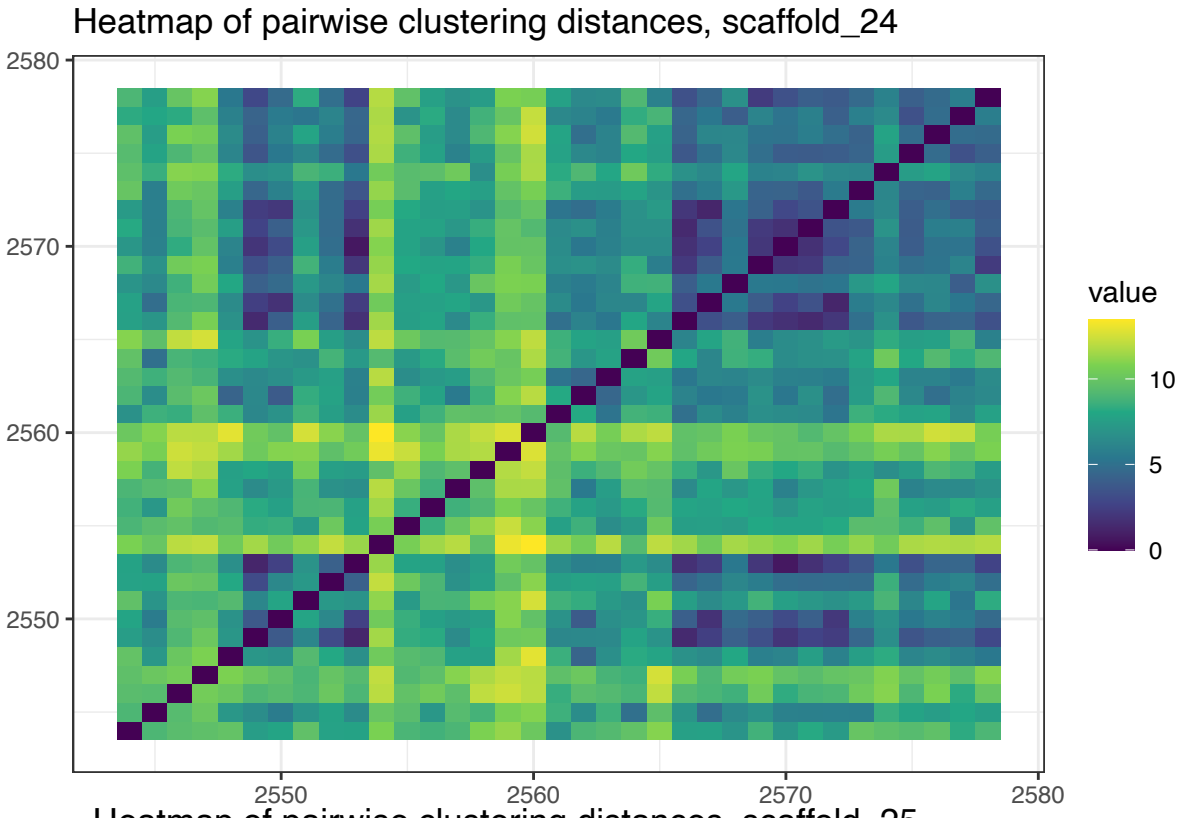

Heatmap of pairwise clustering distances, scaffold_18
2300
2275
2250
2250
2275
2300
value
15
10
5
0

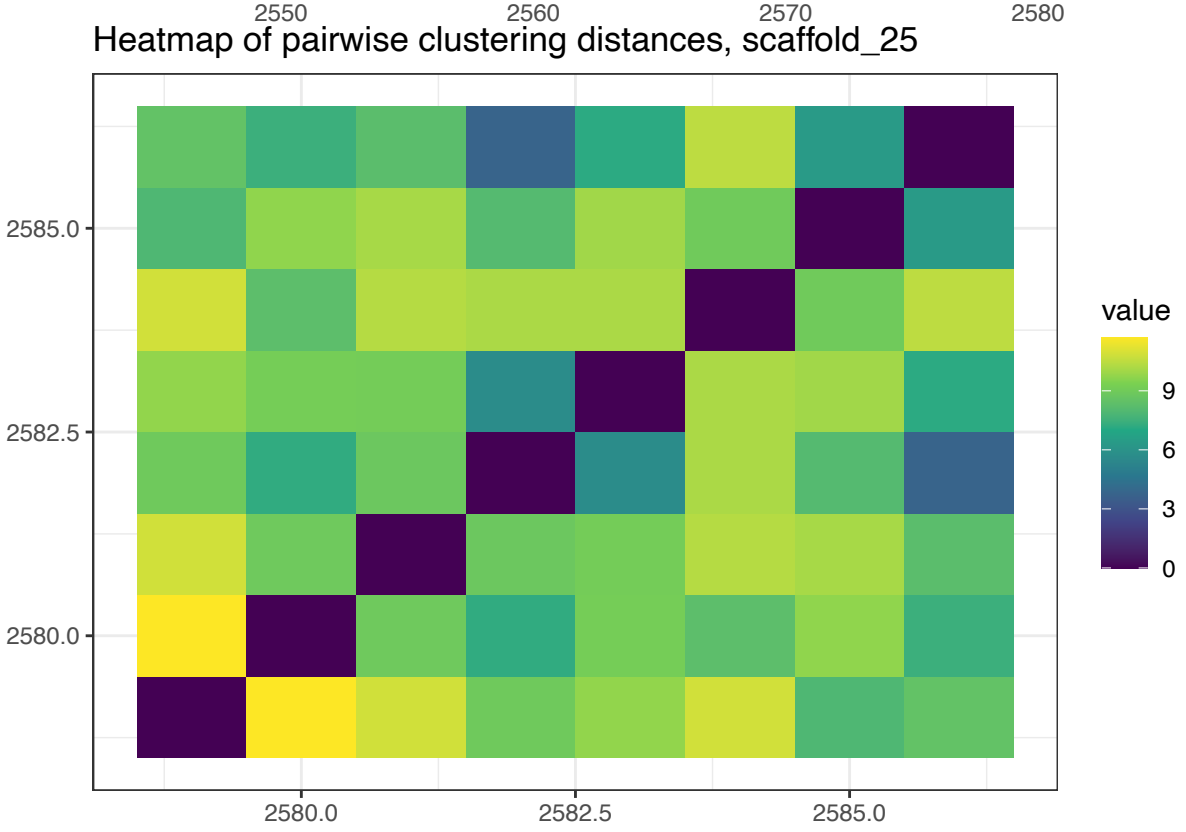

Heatmap of pairwise clustering distances, scaffold_19
2400
2375
2350
2325
2325
2350
2375
2400
value
10
5
0

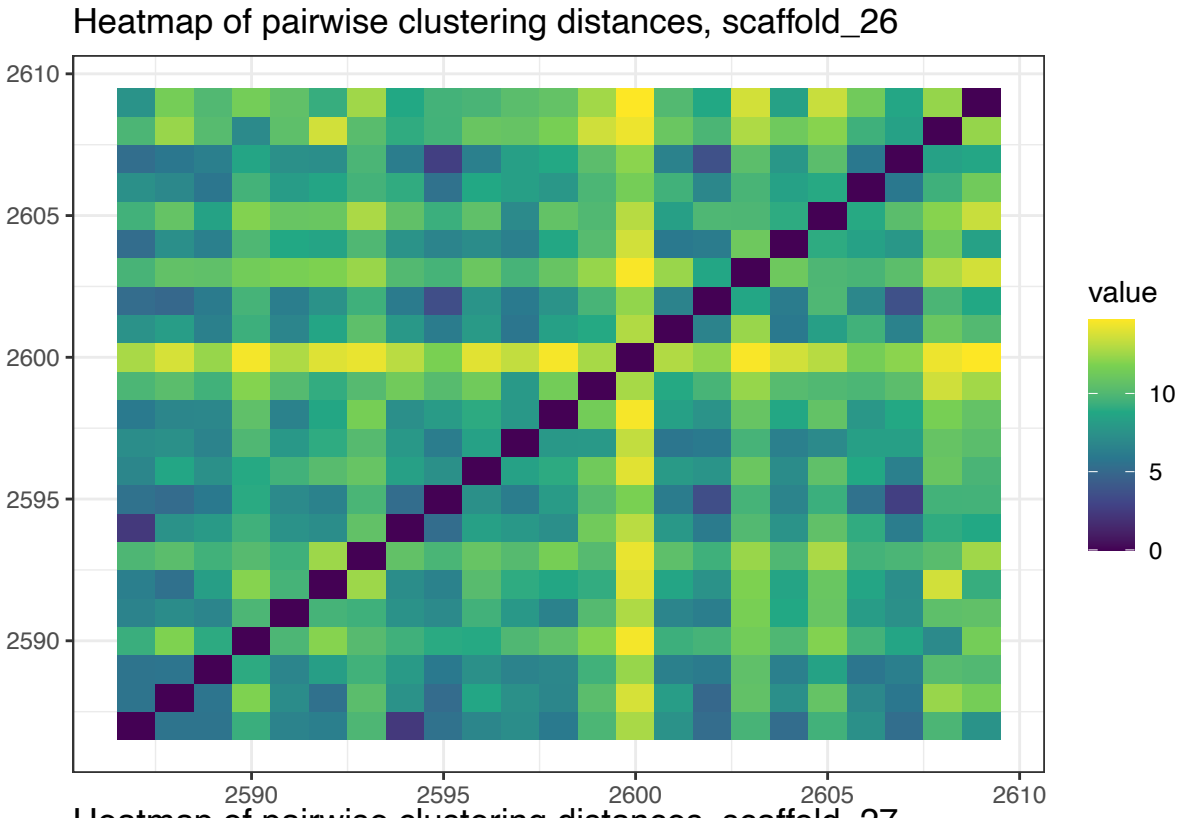

Heatmap of pairwise clustering distances, scaffold_20
2430
2425
2420
2415
2410
2410
2415
2420
2425
2430
value
10.0
7.5
5.0
2.5
0.0

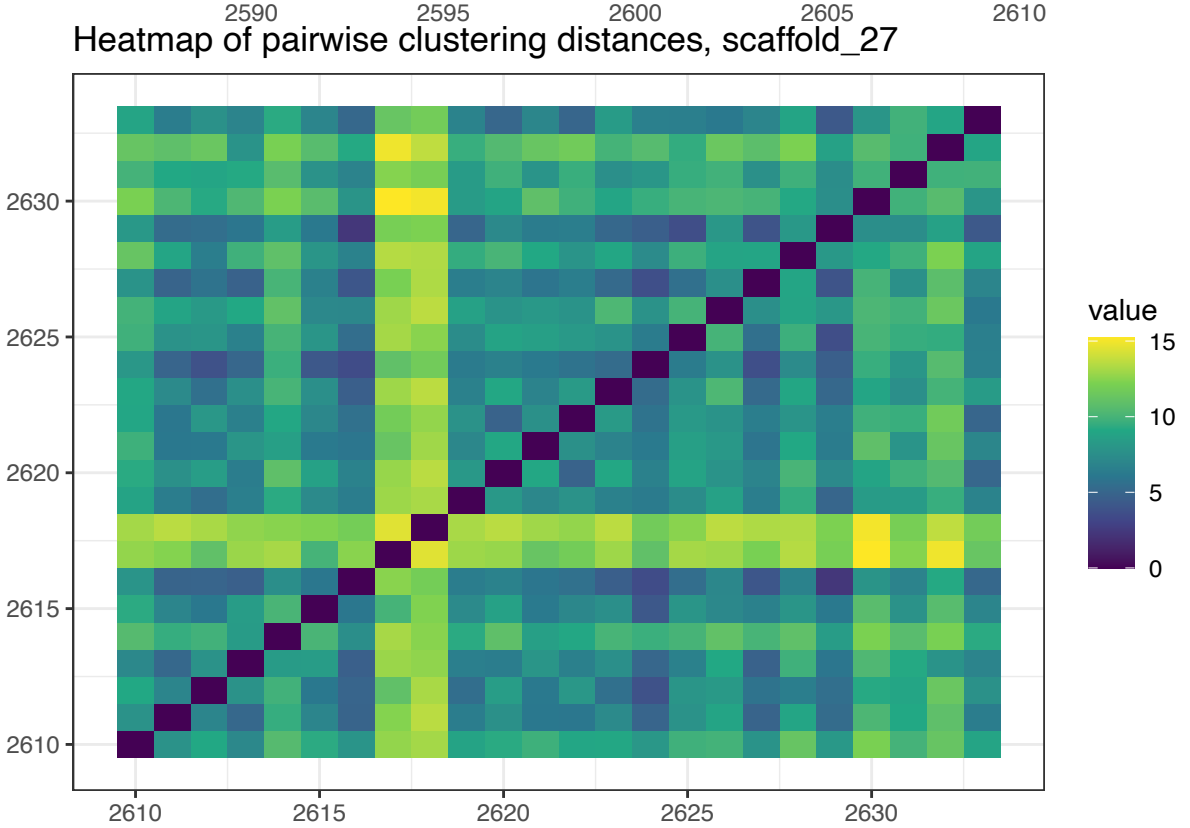

Heatmap of pairwise clustering distances, scaffold_22
2490
2470
2450
2430
2430
2450
2470
2490
value
20
15
10
5
0

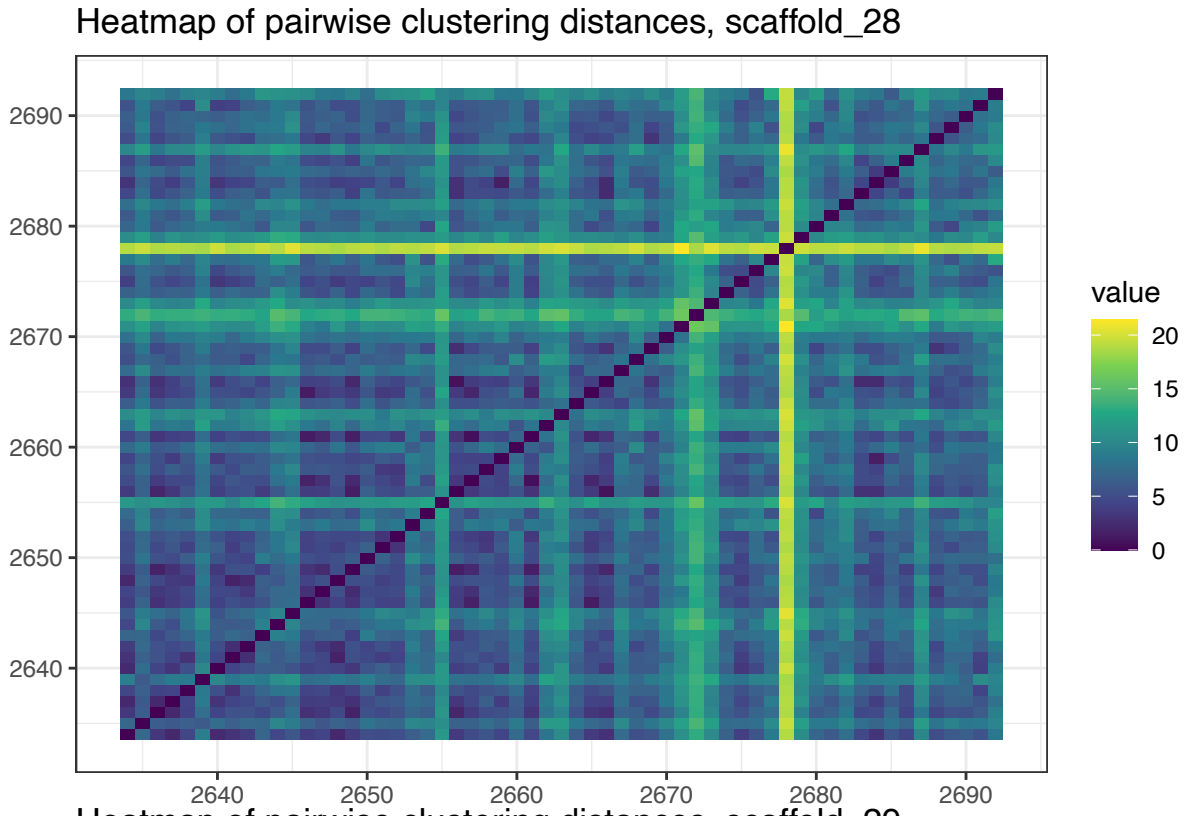

Heatmap of pairwise clustering distances, scaffold_23
2540
2530
2520
2510
2510
2520
2530
2540
value
16
12
8
4
0

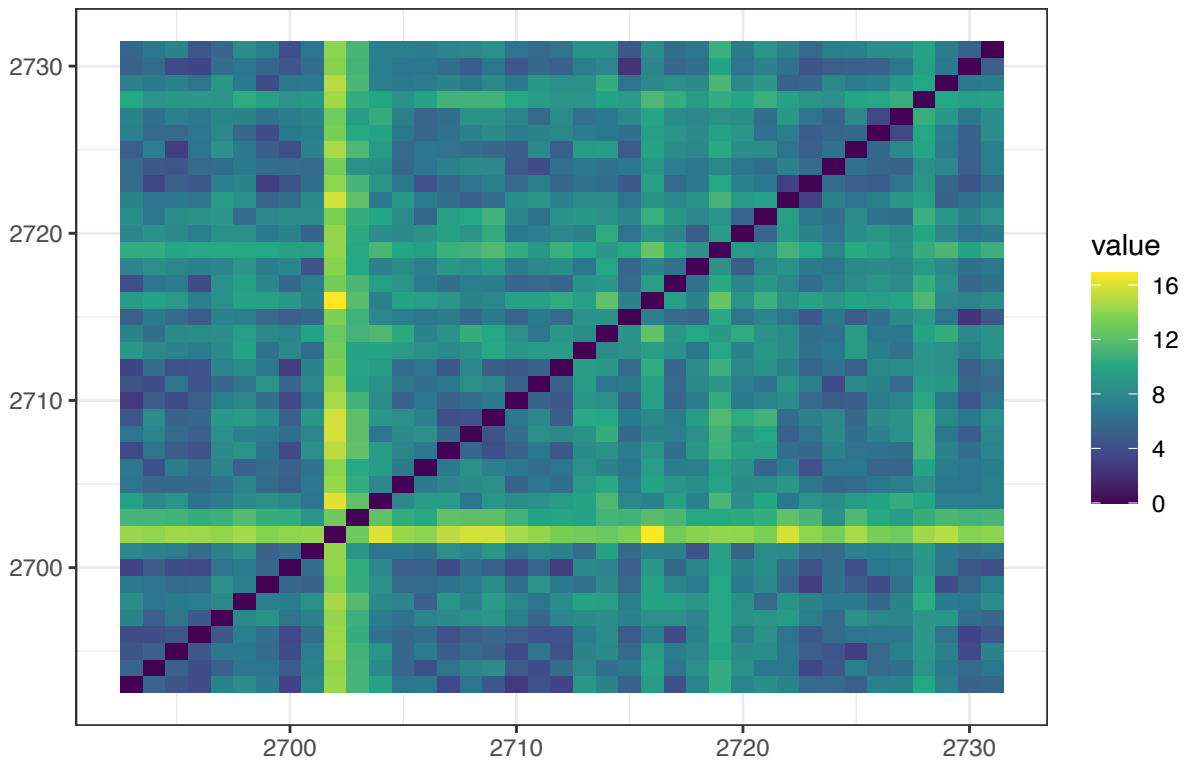

Heatmap of pairwise clustering distances, scaffold_24
2580
2570
2560
2550
2550
2560
2570
2580
value
10
5
0

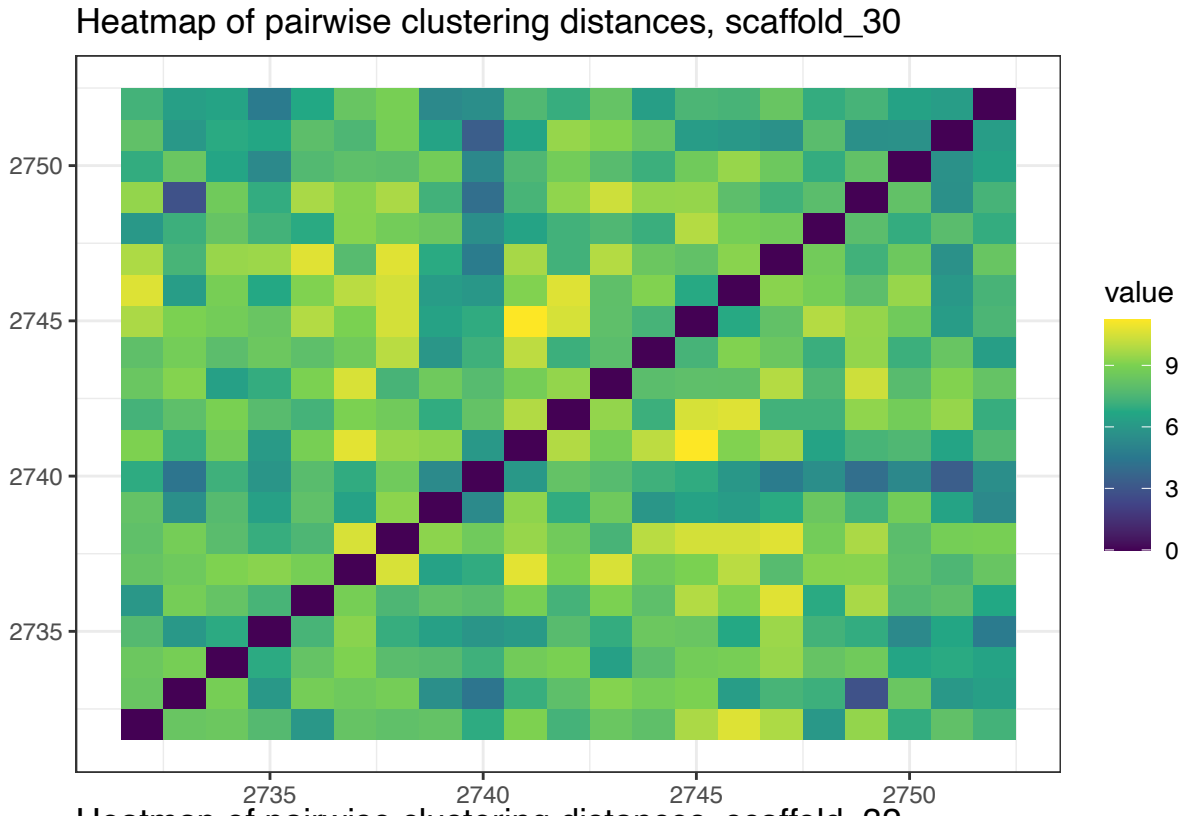

Heatmap of pairwise clustering distances, scaffold_25
2585.0
2582.5
2580.0
2580.0
2582.5
2585.0
value
9
6
3
0

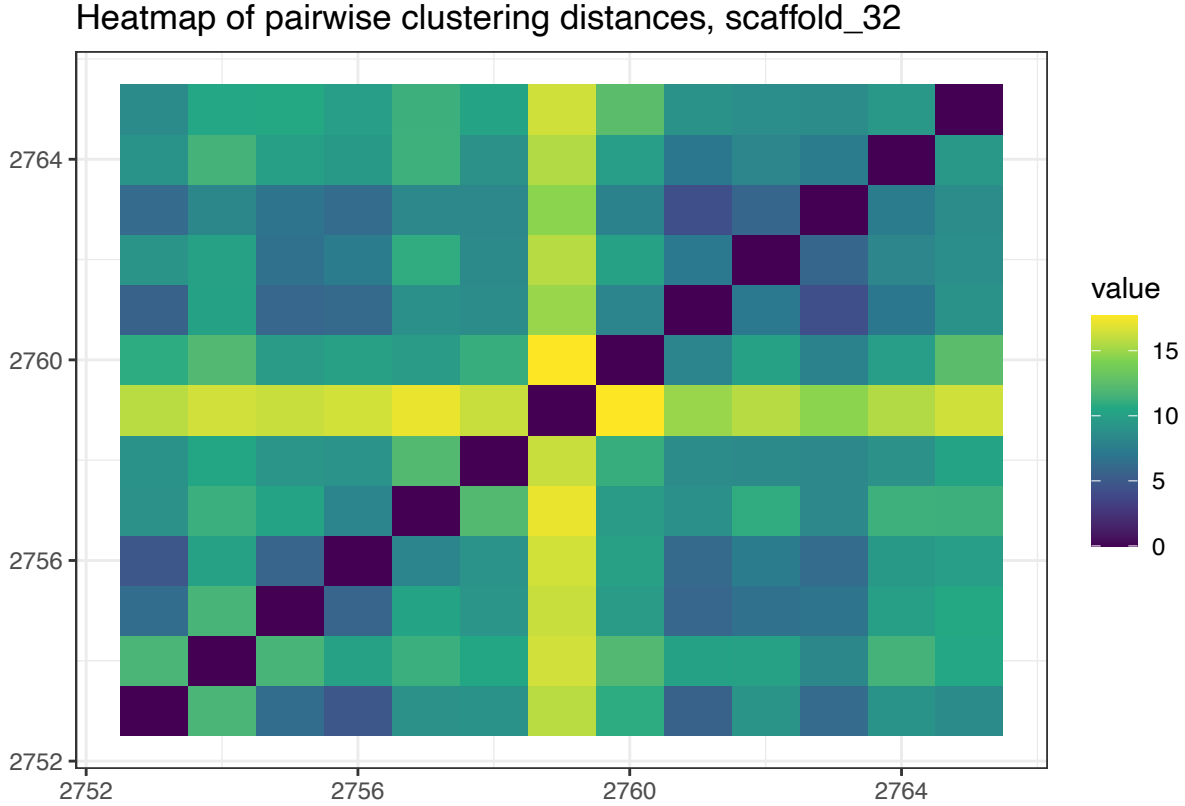

Heatmap of pairwise clustering distances, scaffold_26
2610
2605
2600
2595
2590
2590
2595
2600
2605
2610
value
10
5
0

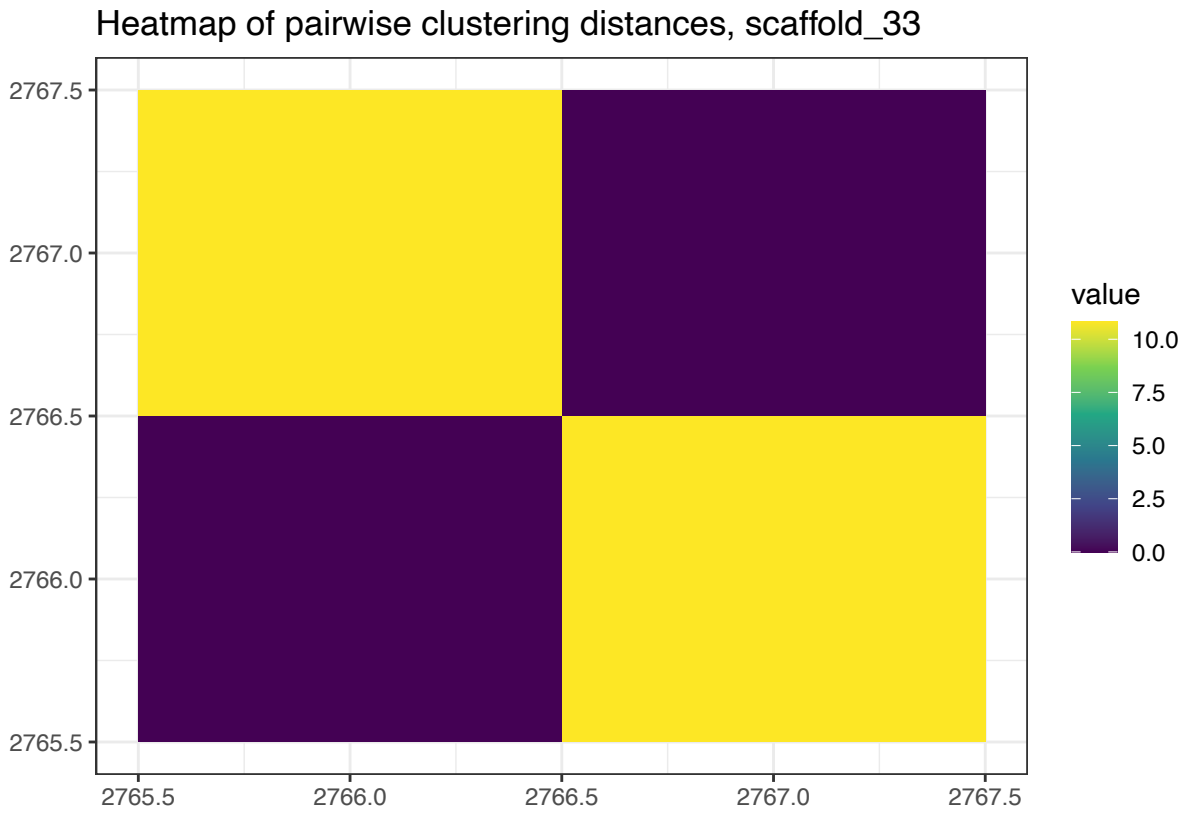

Heatmap of pairwise clustering distances, scaffold_27
2630
2625
2620
2615
2610
2610
2615
2620
2625
2630
value
15
10
5
0


Heatmap of pairwise clustering distances, scaffold_28
2690
2680
2670
2660
2650
2640
2640
2650
2660
2670
2680
2690
value
20
15
10
5
0


Heatmap of pairwise clustering distances, scaffold_29
2730
2720
2710
2700
2700
2710
2720
2730
value
16
12
8
4
0


Heatmap of pairwise clustering distances, scaffold_30
2750
2745
2740
2735
2735
2740
2745
2750
value
9
6
3
0

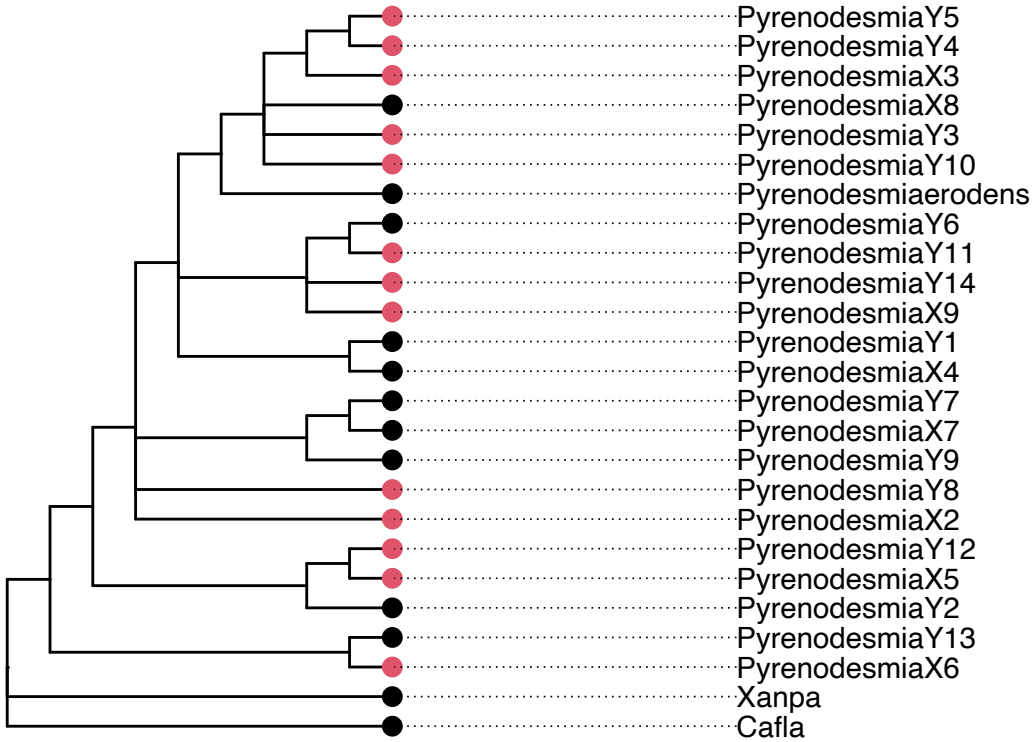

Heatmap of pairwise clustering distances, scaffold_32
2764
2760
2756
2752
2752
2756
2760
2764
value
15
10
5
0



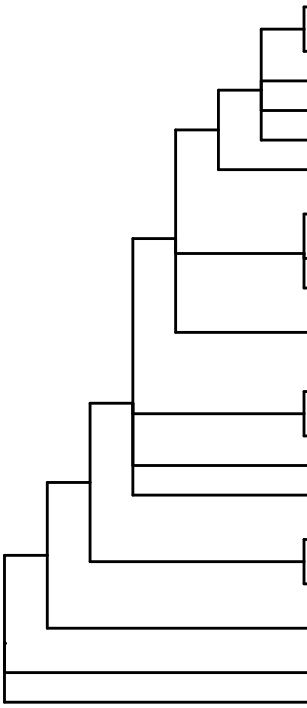
5.2.1 Plot consensus tree topology per scaffold

```
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2)
{
    consino<-consensus(loci_trees[genes[,1]==SCAF], p = 0.5, check.labels = TRUE)
  #plot(consino,main=paste("Consensus Topology of cluster",i),tip.color=as.numeric(factor(mat[consino$t
  p<-ggtree(consino) + geom_tippoint(color=as.numeric(factor(mat[consino$tip.label,2])),size=3) + geom_
  print(p)
}
```

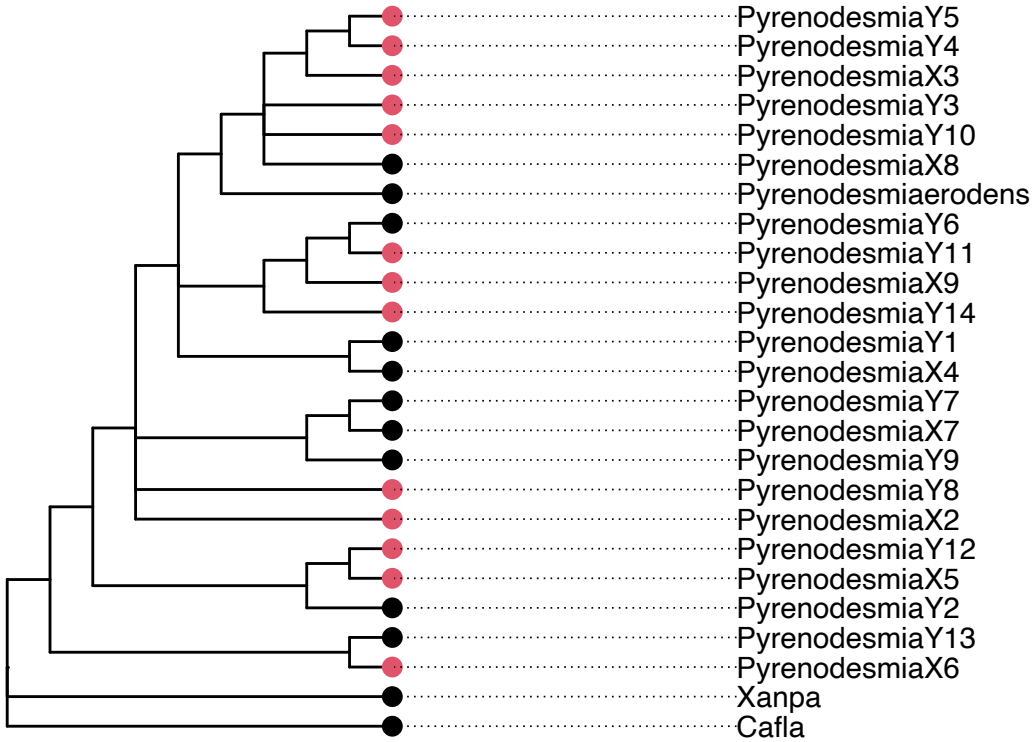
Consensus topology, scaffold_1



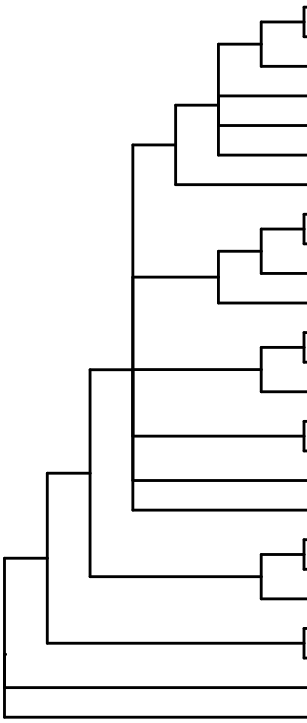
Consensus topology, sc



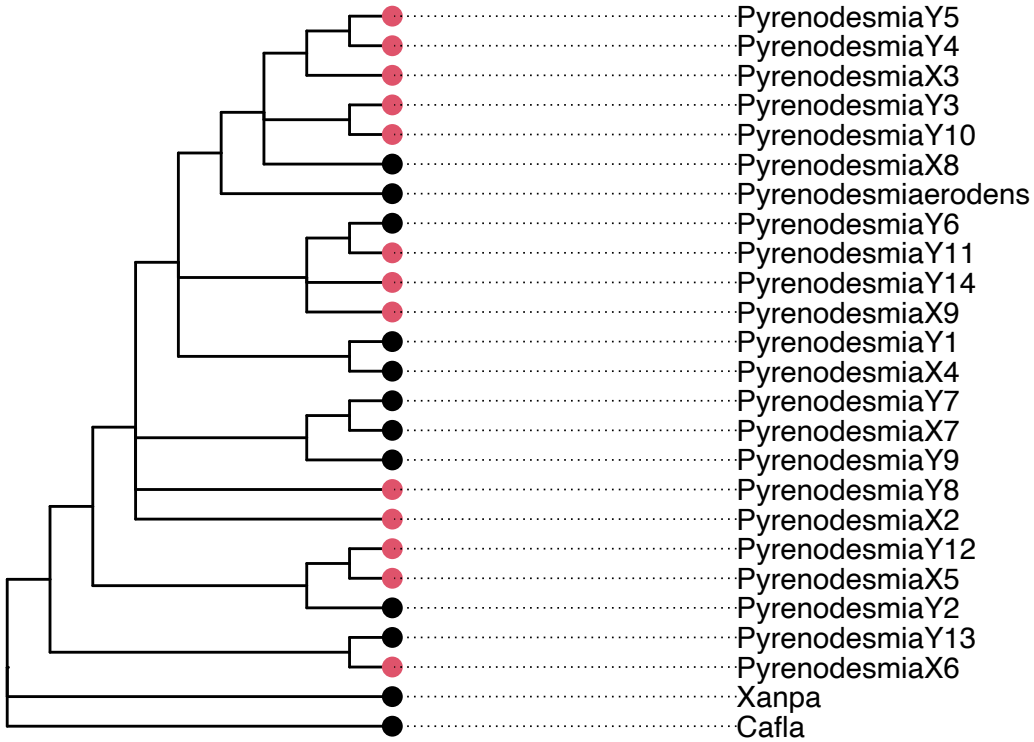
Consensus topology, scaffold_3



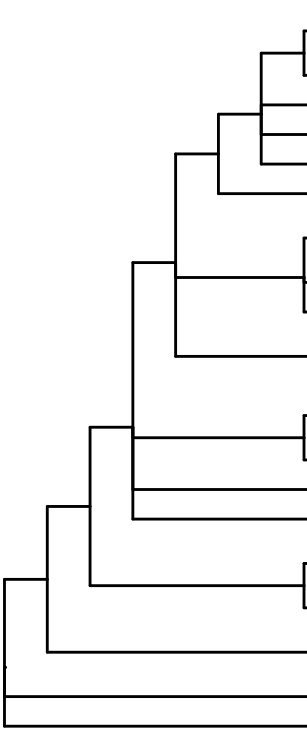
Consensus topology, sc



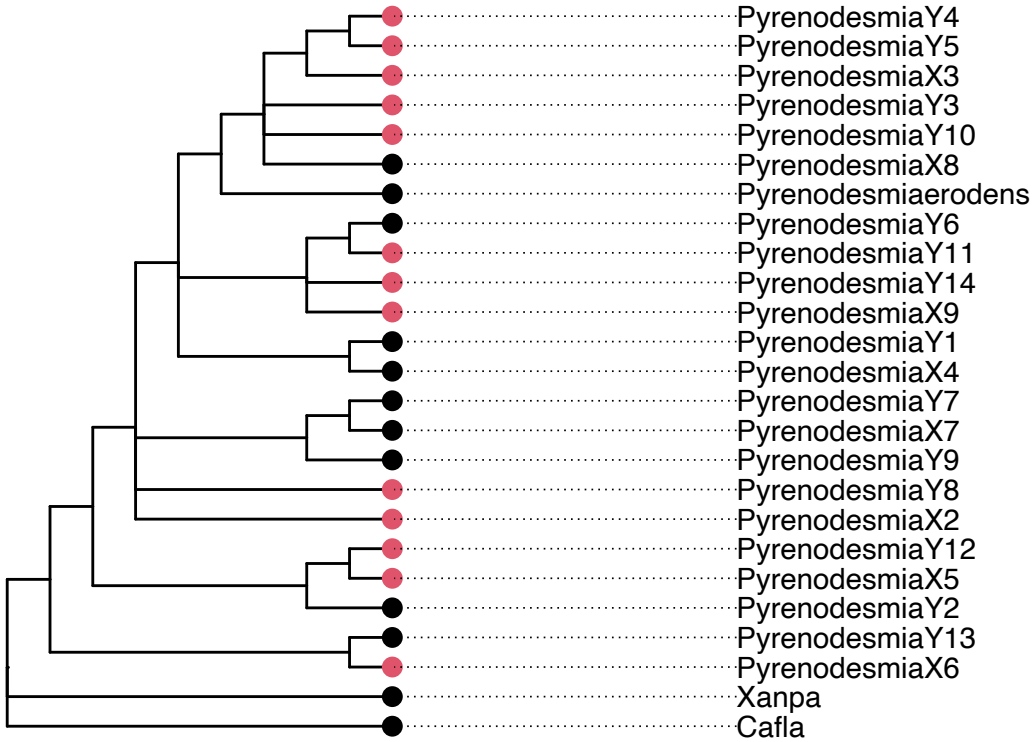
Consensus topology, scaffold_5



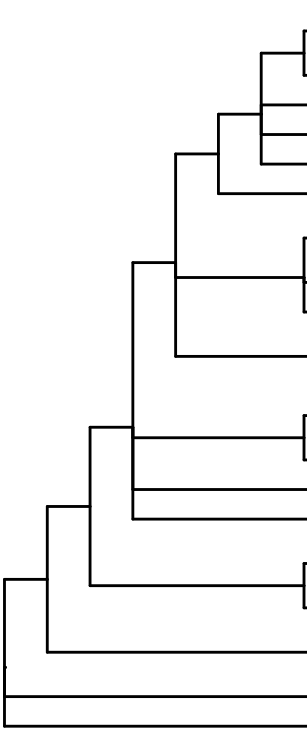
Consensus topology, sc



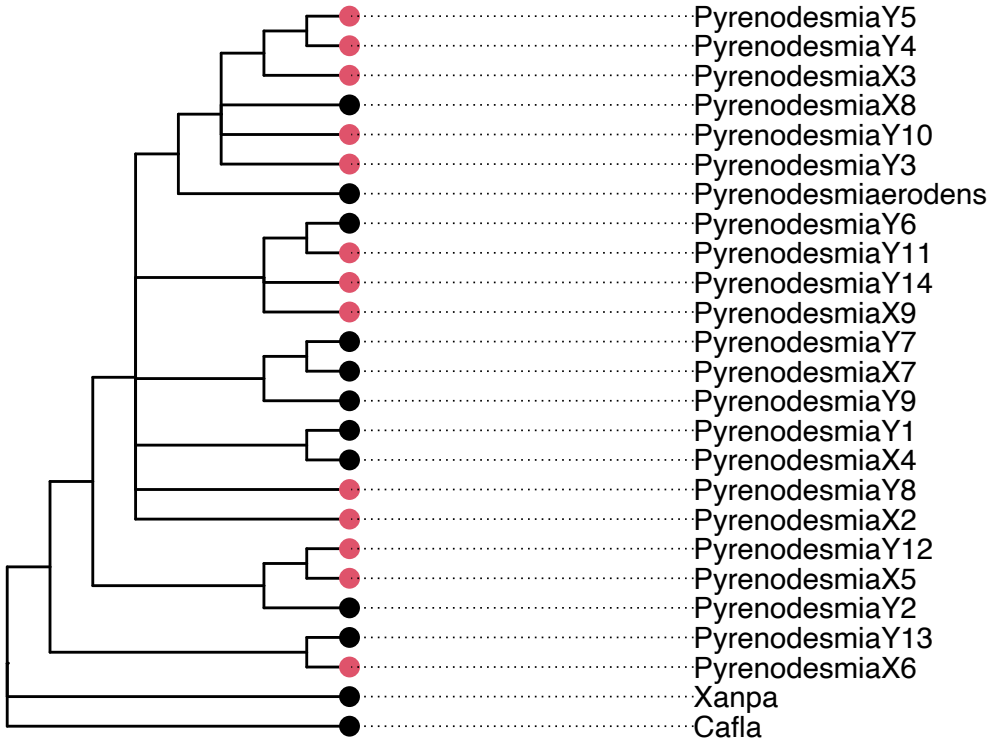
Consensus topology, scaffold_7



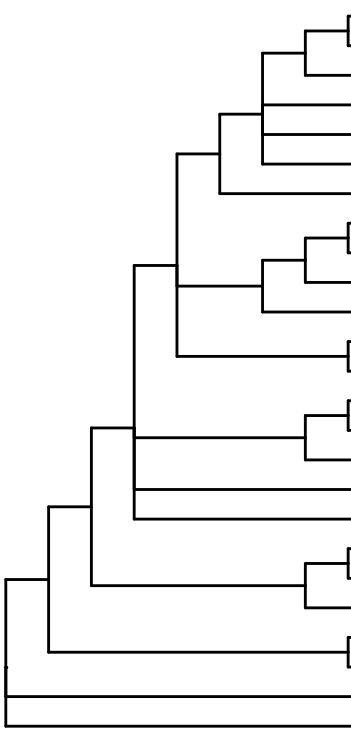
Consensus topology, sc



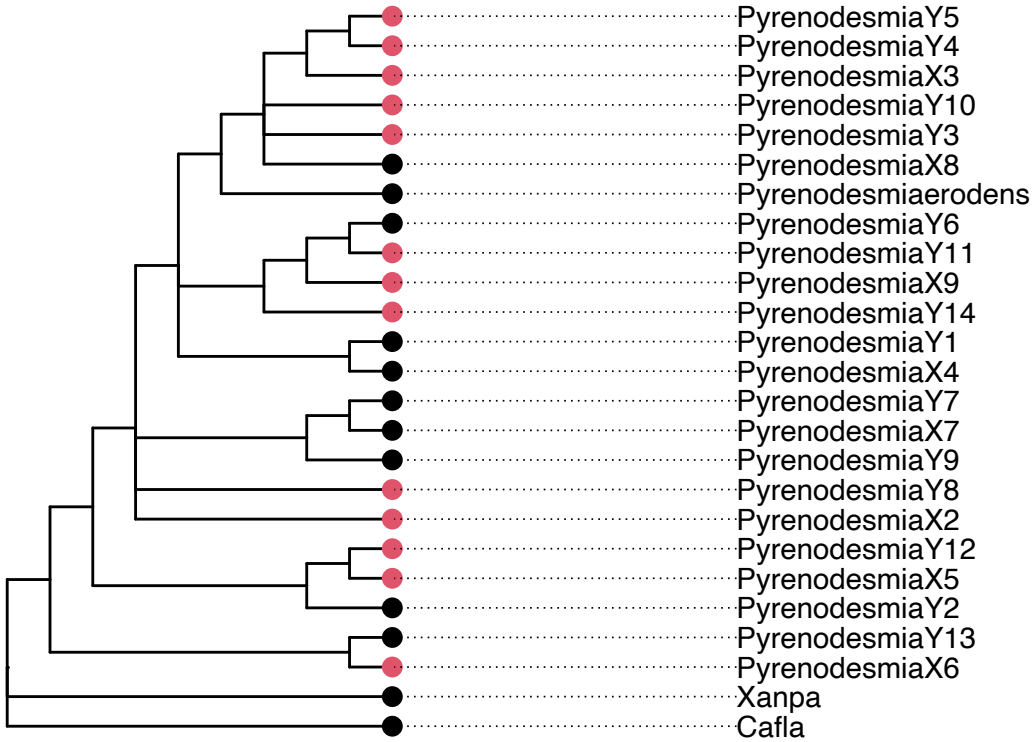
Consensus topology, scaffold_9



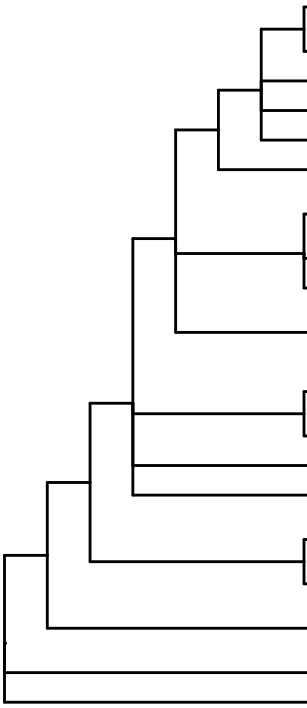
Consensus topology, scaff



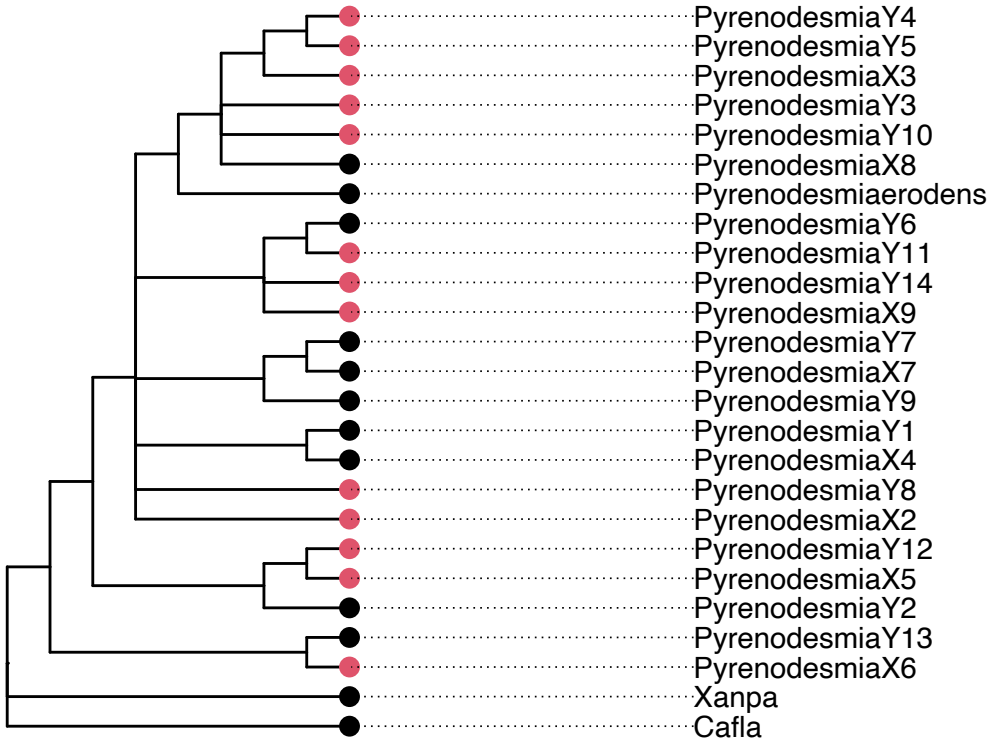
Consensus topology, scaffold_11



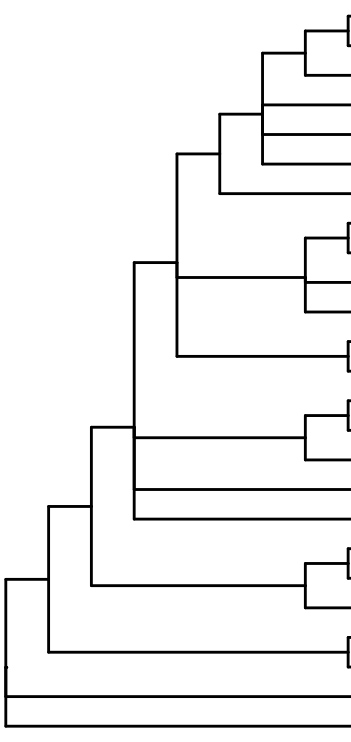
Consensus topology, sc



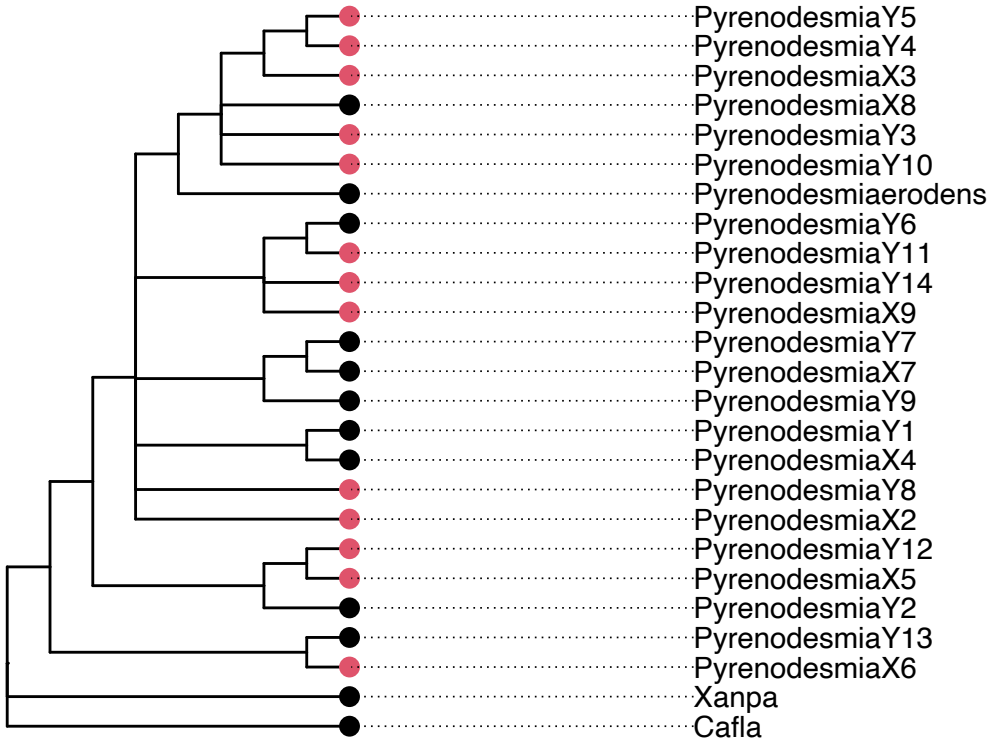
Consensus topology, scaffold_13



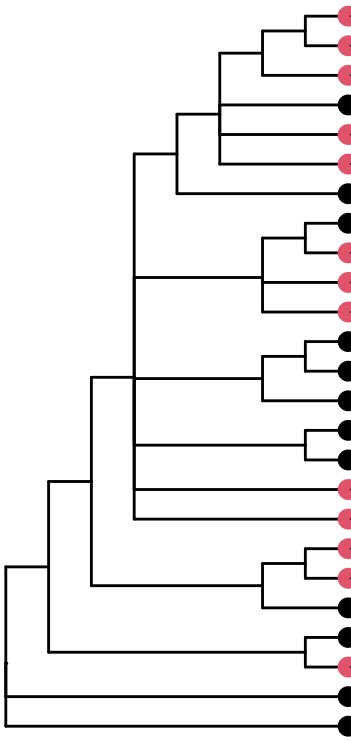
Consensus topology, scaff



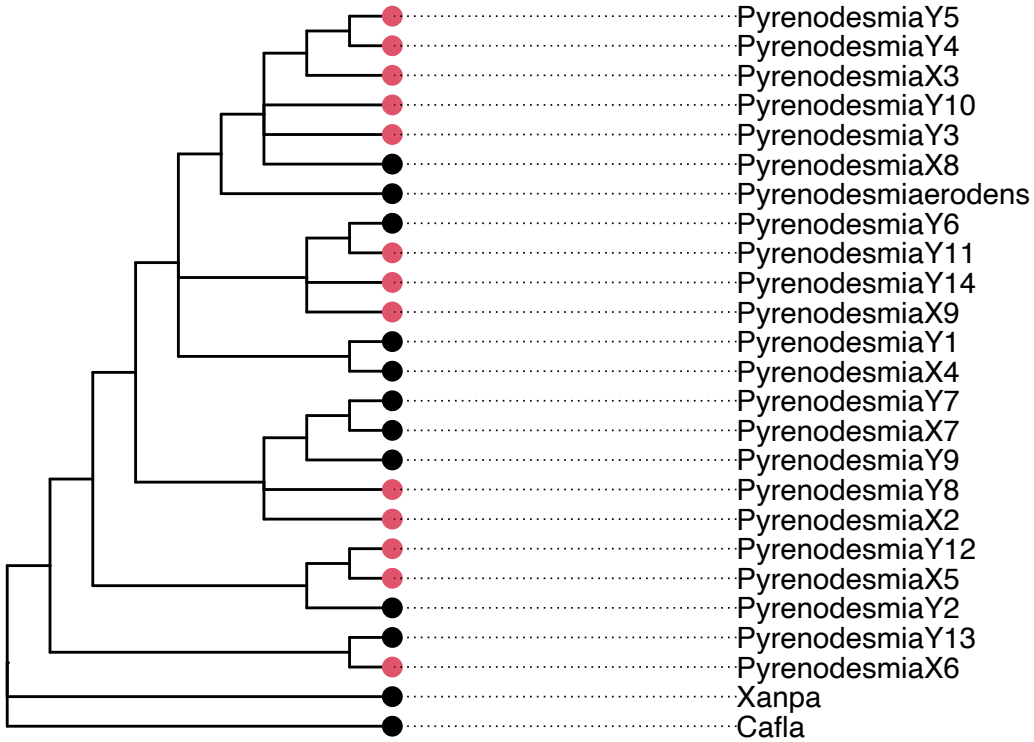
Consensus topology, scaffold_15



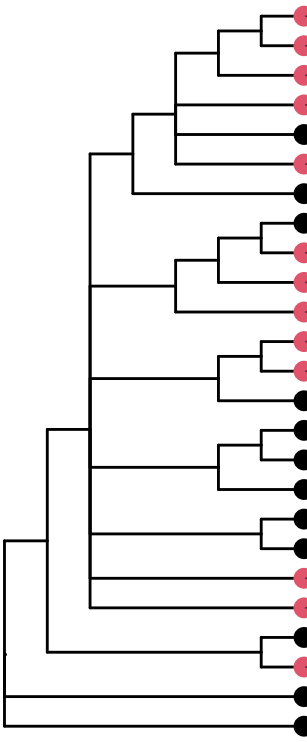
Consensus topology, scaff



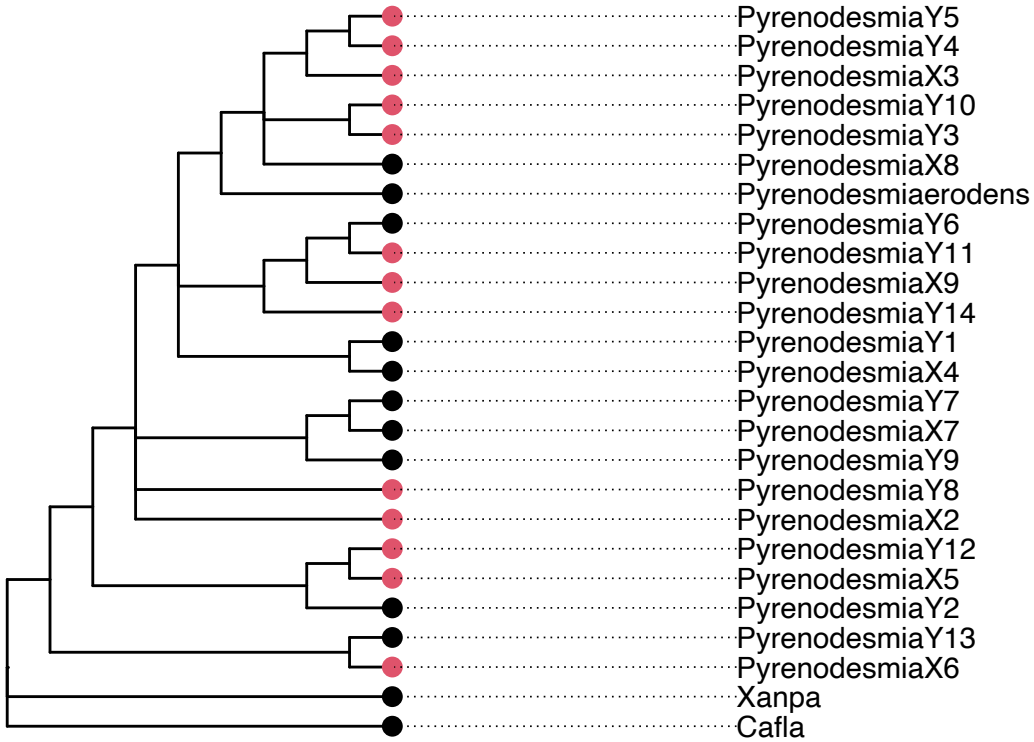
Consensus topology, scaffold_17



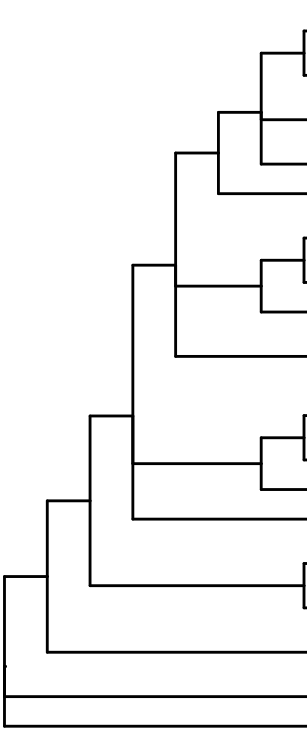
Consensus topology, sc



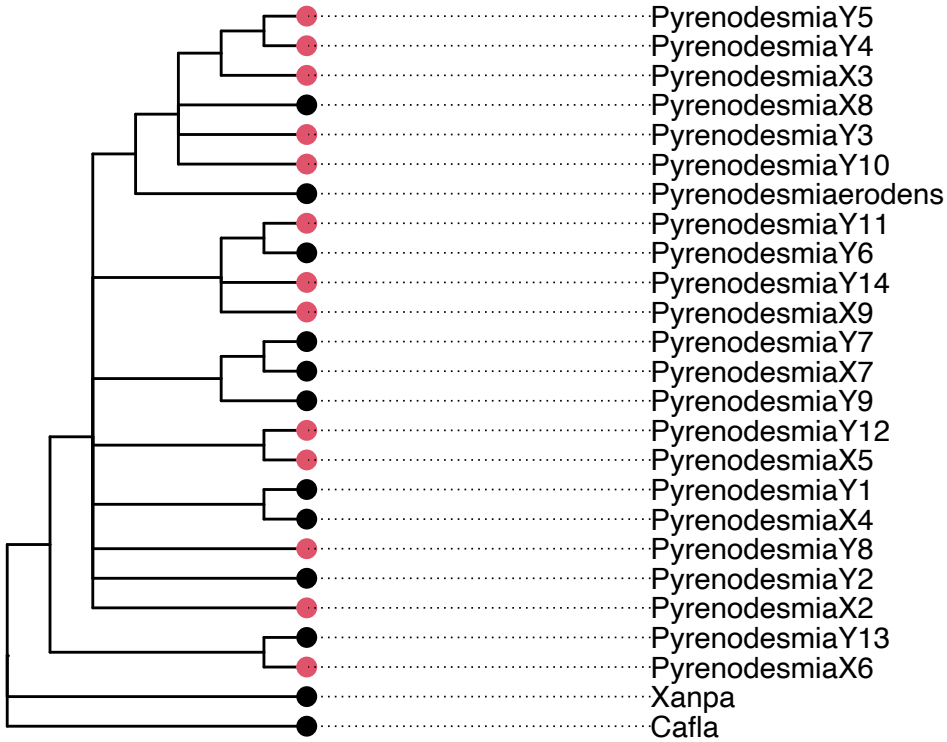
Consensus topology, scaffold_19



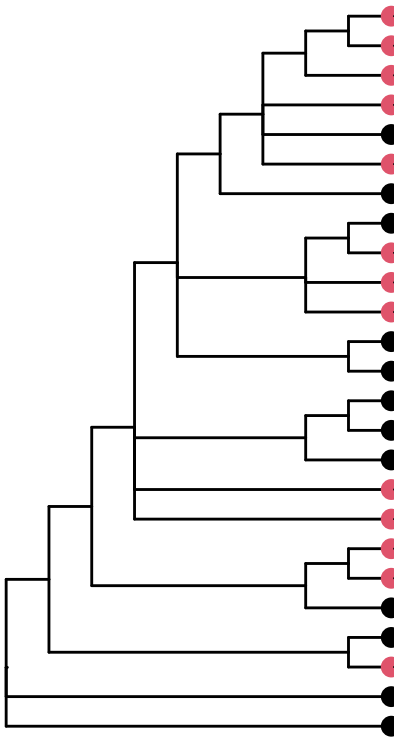
Consensus topology, sc



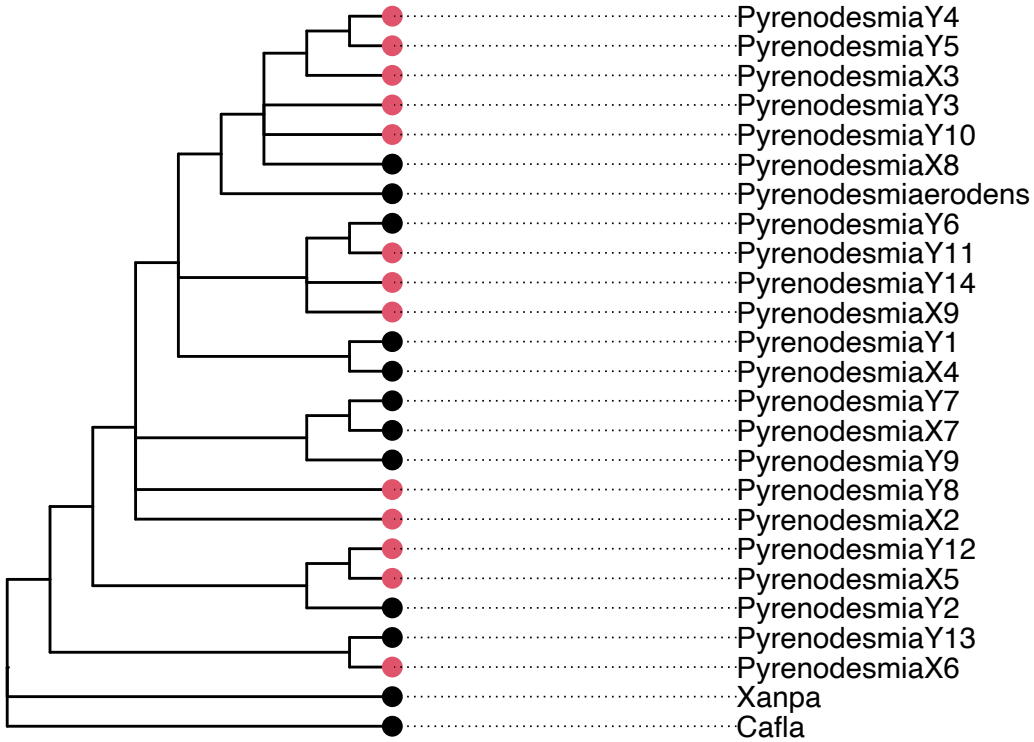
Consensus topology, scaffold_22



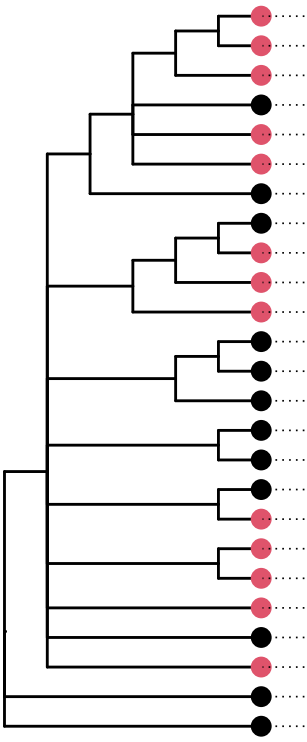
Consensus topology, scaffold



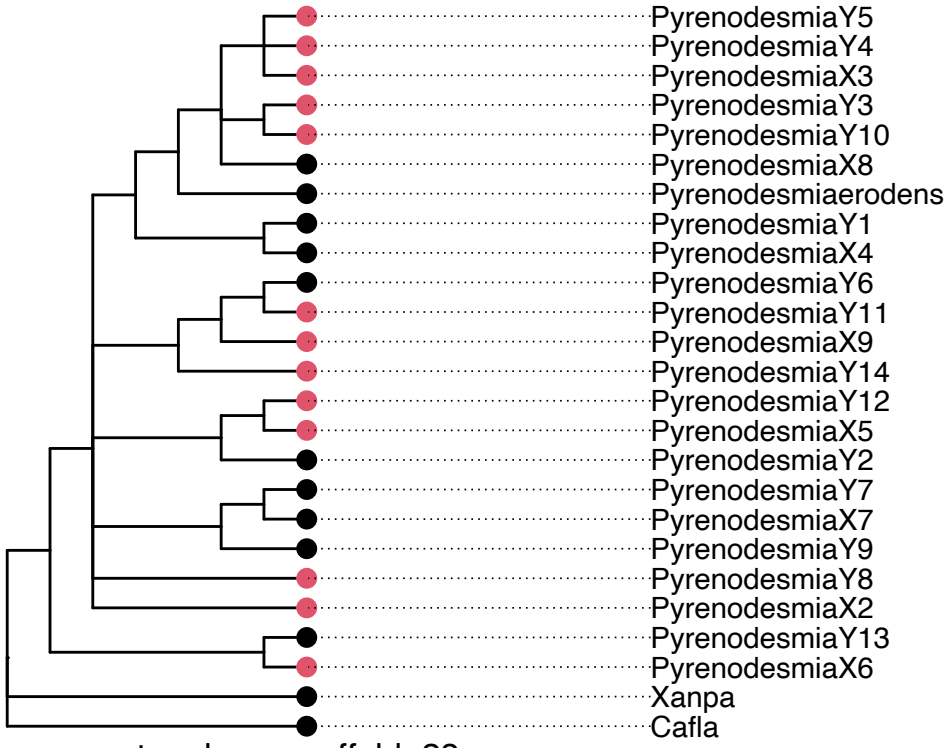
Consensus topology, scaffold_24



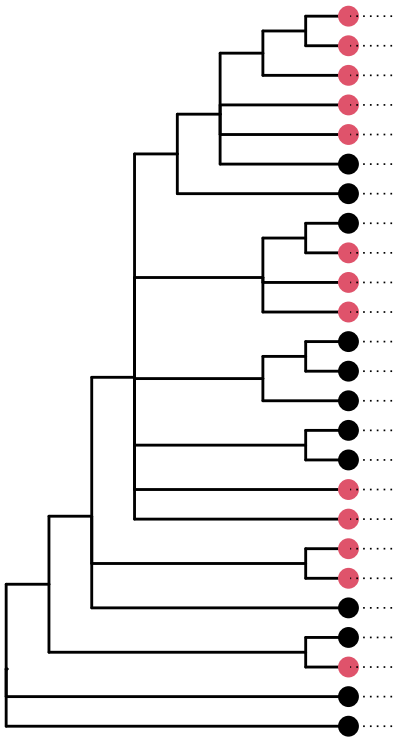
Consensus topology, sc



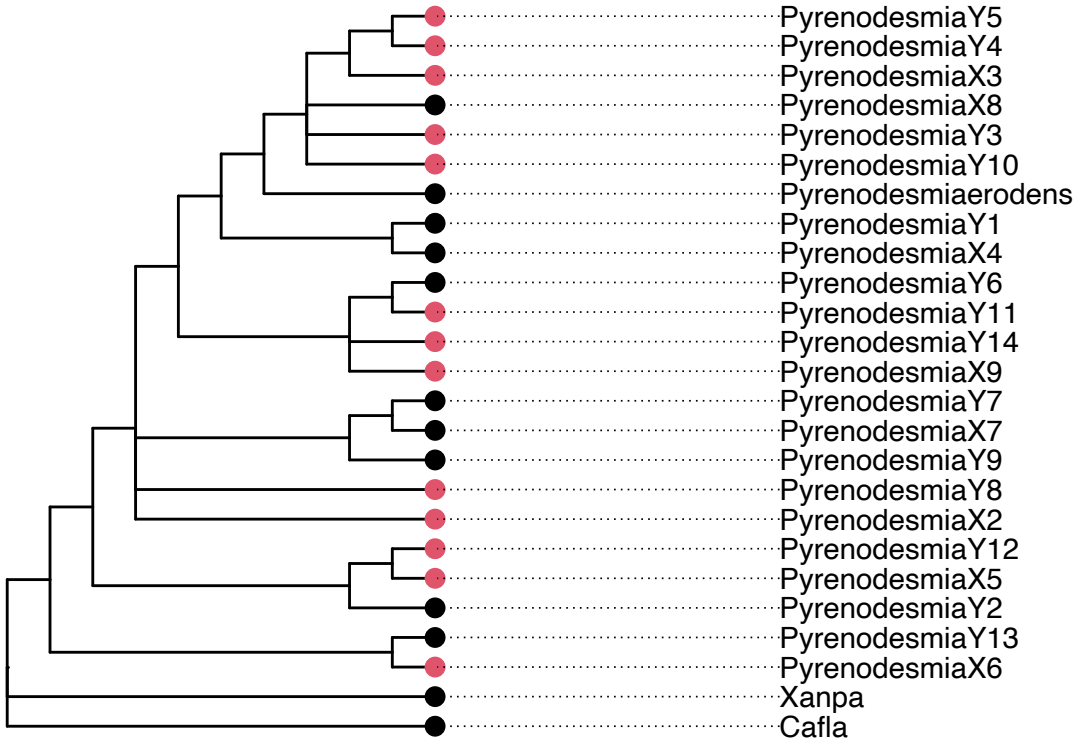
Consensus topology, scaffold_26



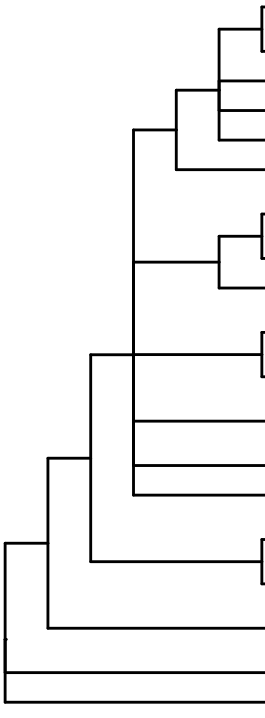
Consensus topology, scaffold



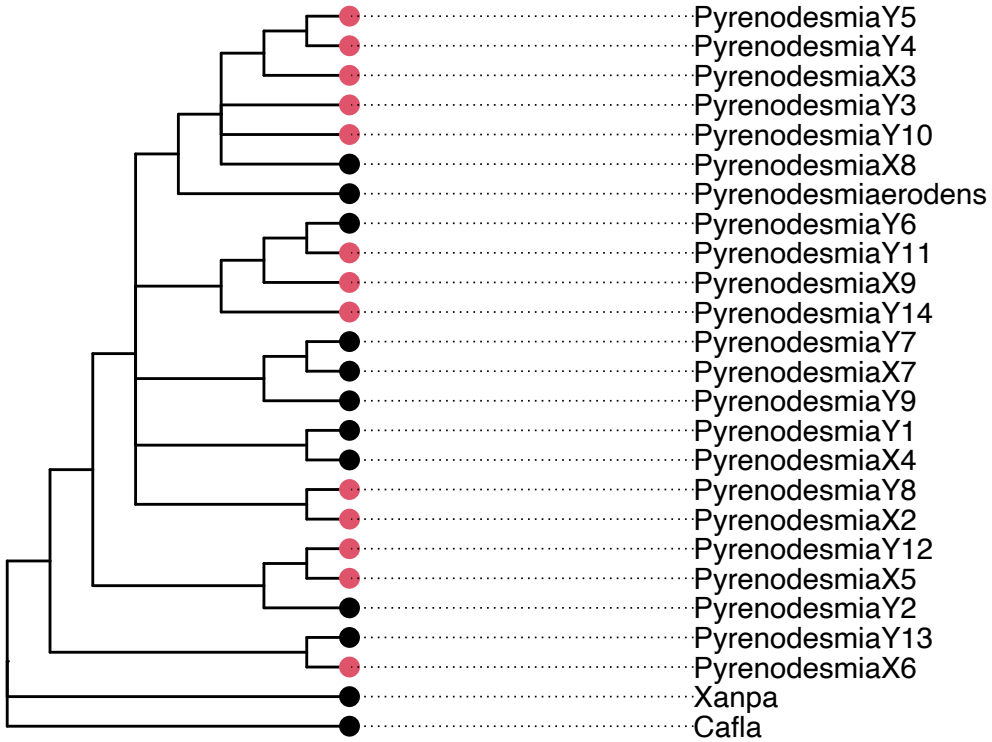
Consensus topology, scaffold_28



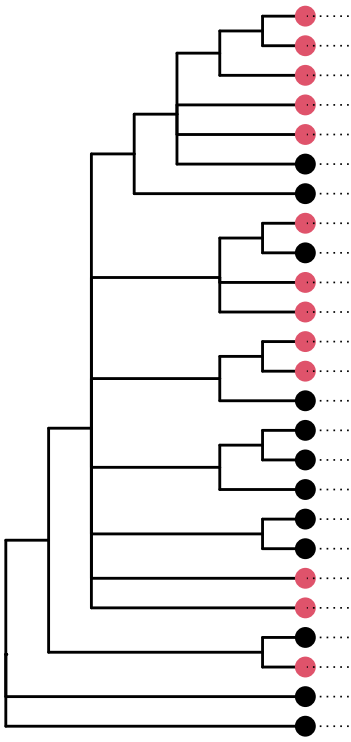
Consensus topology



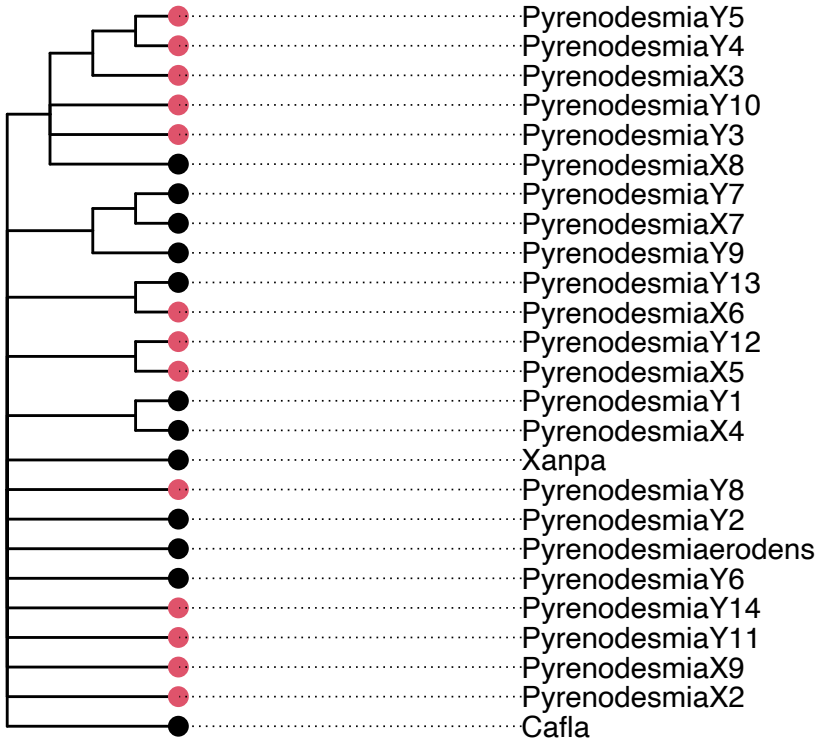
Consensus topology, scaffold_30



Consensus topology, scaff



Consensus topology, scaffold_33



5.2.2 Plot overall consensus topology

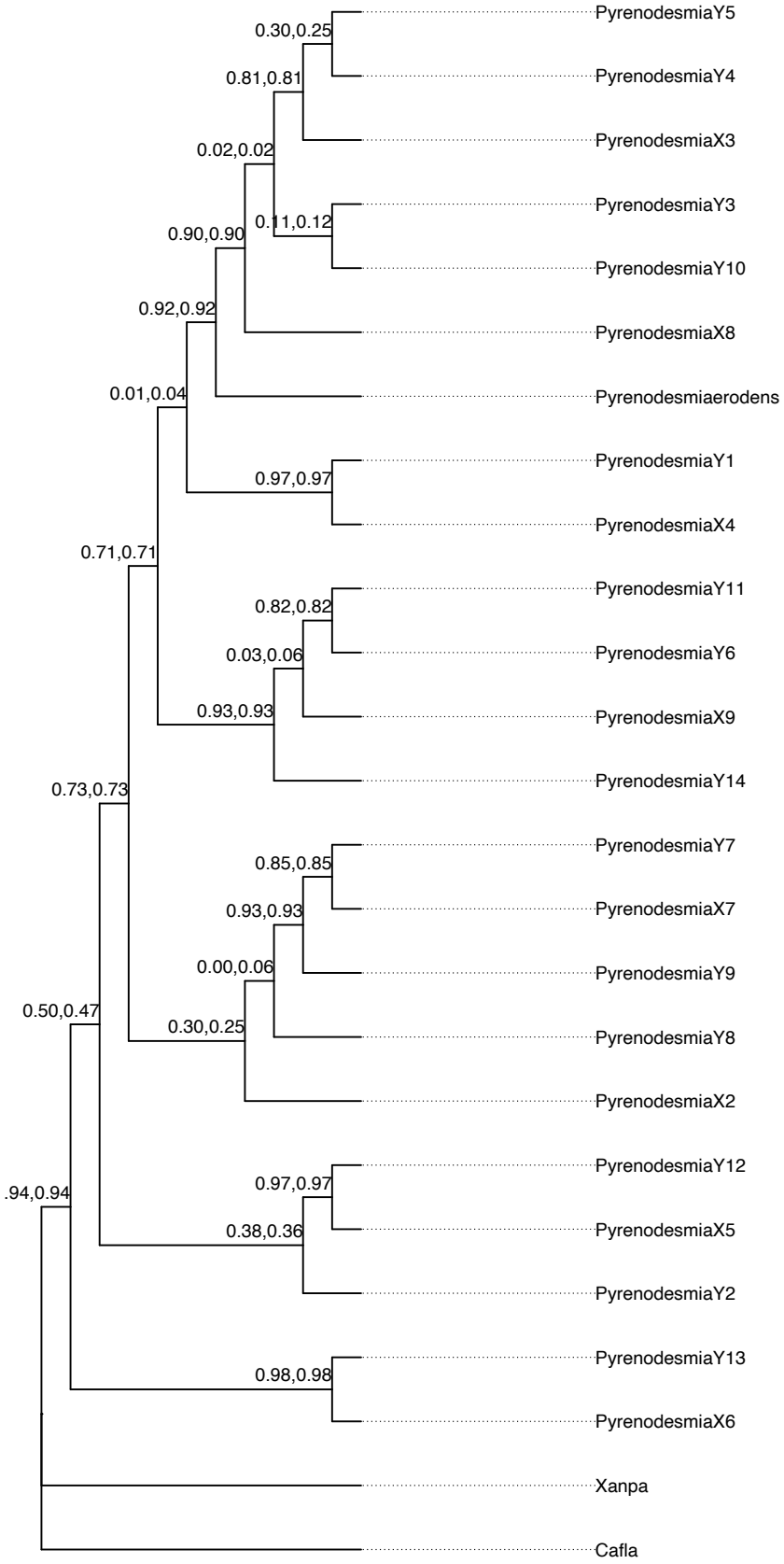
```
#for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2
#{
#    write.tree(loci_trees[genes[,1]==SCAF],paste("/Users/Fernando/Desktop/3_Phylogenomics/",SCAF,"arbo
#}
library(phylotate)
cupo<-read_annotated("/Users/Fernando/Desktop/3_Phylogenomics/RAxML_MajorityRuleExtendedConsensusTree_I
p<-ggtree(cupo) + geom_tiplab(align=TRUE, linetype="dotted", linesize=.3, offset=8) + xlim(0,25) + labs
```

```
## Warning in fortify.phylo(data, ...): 'edge.length' contains NA values...
## ## setting 'edge.length' to NULL automatically when plotting the tree...
```

```
print(p)
```

```
## Warning: Removed 26 rows containing missing values (geom_text).
```

Extended Majority Rule Consensus annotated with TC and IC support values.



```
# + geom_tippoint(color=as.numeric(factor(mat[consino$tip.label,2])),size=3)
```

5.2.3 Use the consensus tree to summarize comparative genomic data

```
library(aplot)
info<-synopsis_backup
rownames(info)<-info$Species<-c("Xanpa","Cafla","Pyrenodesmiaerodens","PyrenodesmiaX2","PyrenodesmiaX3"
info$name_correct<-c("Xanthoria parietina",
                     "Gyalolechia flavorubescens",
                     "Pyrenodesmia erodens",
                     "Pyrenodesmia transcaspica X2",
                     "Pyrenodesmia variabilis X3",
                     "Pyrenodesmia chalybaea X4",
                     "Pyrenodesmia micromontana X5",
                     "Pyrenodesmia medit1 X6",
                     "Pyrenodesmia alociza X7",
                     "Pyrenodesmia circumalbata X8",
                     "Pyrenodesmia micromarina X9",
                     "Pyrenodesmia chaplybaea Y1",
                     "Pyrenodesmia variabilis Y2",
                     "Pyrenodesmia variabilis Y3",
                     "Pyrenodesmia variabilis Y4",
                     "Pyrenodesmia variabilis Y5",
                     "Pyrenodesmia sp. Y6",
                     "Pyrenodesmia alociza Y7",
                     "Pyrenodesmia albovariegata Y8",
                     "Pyrenodesmia alociza Y9",
                     "Pyrenodesmia circumalbata Y10",
                     "Pyrenodesmia alociza Y11",
                     "Pyrenodesmia cf. albopruinosa Y12",
                     "Pyrenodesmia medit1 Y13",
                     "Pyrenodesmia cf. micromarina Y14")
info$cluster<-c(NA,
                NA,
                4,
                9,
                8,
                5,
                1,
                10,
                2,
                8,
                6,
                5,
                9,
                8,
                8,
                8,
                6,
                7,
                9,
                3,
                8,
```

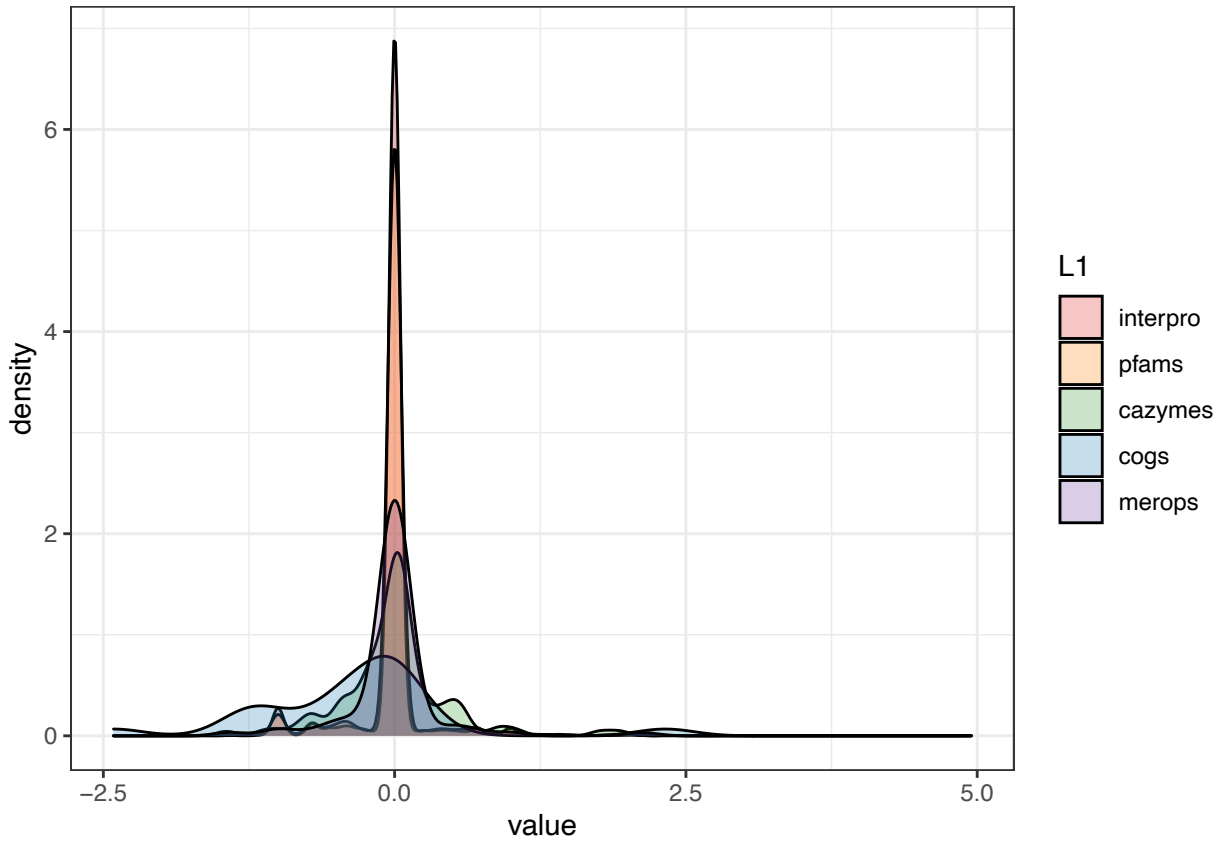
```
                6,
                1,
                10,
                6)

info<-info[cupo$tip.label,]
info$Percent.GC<-as.numeric(info$Percent.GC)
info$node<-1:25
info<-fortify(info)
#facet_plot(p,panel="Size",data=info,geom=geom_bar,mapping = aes (x=Percent.GC))

pfams<-read.csv("/Users/Fernando/Desktop/compare_all/pfam/pfam.results.csv",row.names = 1)
pfams_1<-pfams[,c("descriptions")]
pfams_2<-pfams[,!(colnames(pfams)%in%c("descriptions"))]

cazymes<-read.csv("/Users/Fernando/Desktop/compare_all/cazy/CAZyme.all.results.csv",row.names = 1)

merops<-read.csv("/Users/Fernando/Desktop/compare_all/merops/MEROPS.all.results.csv",row.names = 1)

cogs<-read.csv("/Users/Fernando/Desktop/compare_all/cogs/COGS.all.results.csv",row.names = 1)

interpro<-read.csv("/Users/Fernando/Desktop/compare_all/interpro/interproscan.results.csv",row.names = 
interpro_desc<-interpro$descriptions
interpro<-interpro[,!(colnames(interpro)%in%c("descriptions"))]
#
# Process and obtain residual distributions
#
results_all<-list(
  cazymes=apply(cazymes,1,FUN=function(x){(x[3]-mean(x[-3]))/sqrt(mean(x[-3]))}),
  pfams=apply(pfams_2,1,FUN=function(x){(x[3]-median(x[-3]))/sqrt(median(x[-3]))}),
  merops=apply(merops,1,FUN=function(x){(x[3]-median(x[-3]))/sqrt(median(x[-3]))}),
  cogs=apply(cogs,1,FUN=function(x){(x[3]-median(x[-3]))/sqrt(median(x[-3]))}),
  interpro=apply(interpro,1,FUN=function(x){(x[3]-median(x[-3]))/sqrt(median(x[-3]))})
)
results_all<-melt(results_all)
#
# TAKE OUT NAS DUE TO 0/0
#
results_all<-results_all[!is.na(results_all$value),]
results_all$L1<-factor(results_all$L1,levels=c("interpro","pfams","cazymes","cogs","merops"))
```

5.2.4 Most PFAMs are equally present across all samples

The observed value of P.erodens is mostly equal to the median of the PFAm table occurrences (The residual Obs-Expected/sqrt(Expected) is zero).

```
ggplot(data.frame(results_all),aes(x=value,group=L1,fill=L1)) + geom_density(alpha=0.25) + theme_bw() +
```

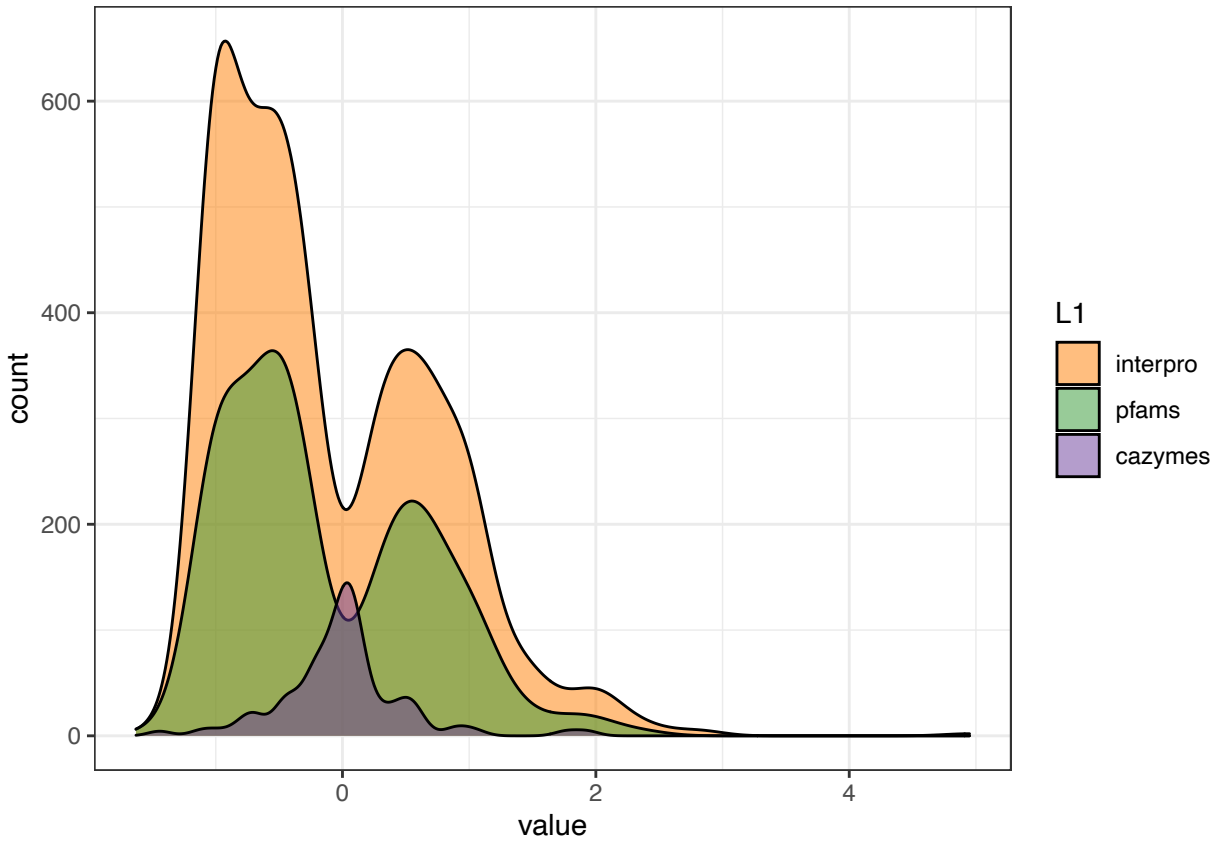
```
## Warning: Removed 83 rows containing non-finite values (stat_density).
```



Taking out all PFAMs with residuals different to zero we observe a higher frequency of negative than possitive residuals

```
results_all<-results_all[results_all$value!=0,]

ggplot(results_all[results_all$L1%in%c("cazymes","interpro","pfams"),],aes(x=value,group=L1,fill=L1)) +
```

```
## Warning: Removed 83 rows containing non-finite values (stat_density).
```




```
ipr<-apply(interpro,1,FUN=function(x){(x[3]-median(x[-3]))/sqrt(median(x[-3]))})
foo<-ipr!=0&!(is.na(ipr))&!(is.infinite(ipr))
foo<-data.frame(desc=interpro_desc[foo],res=ipr[foo])
foo<-foo[order(foo$res),]
```

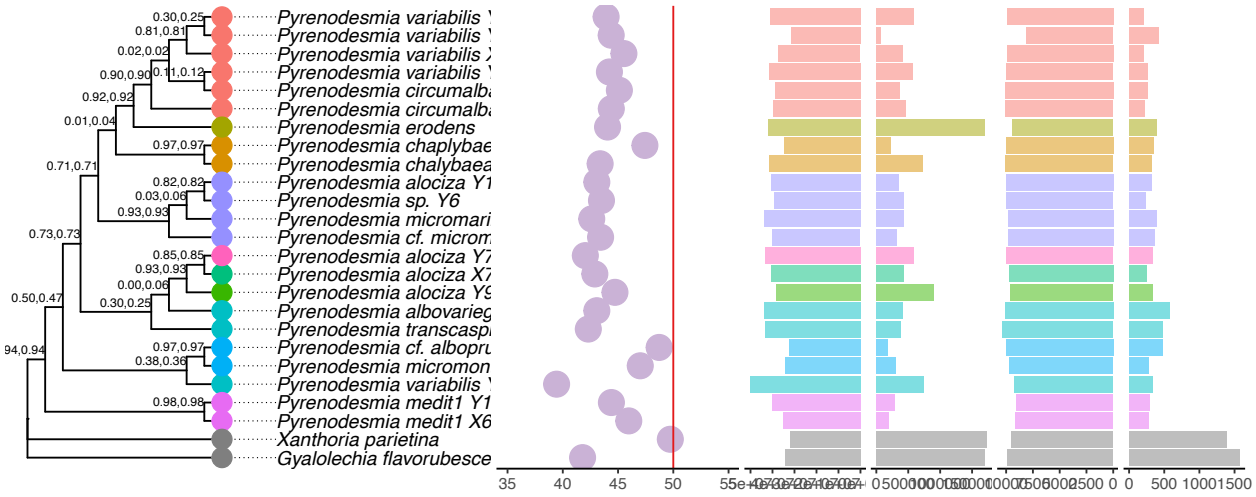
```
p<-ggtree(cupo) %<+% info  + geom_tiplab(aes(label=name_correct),fontface=3,align=TRUE, linetype="dotte
```

```
## Warning in fortify.phylo(data, ...): 'edge.length' contains NA values...
## ## setting 'edge.length' to NULL automatically when plotting the tree...
```

```
p2<-ggplot(data=info,aes(y=Percent.GC,x=Species))+geom_point(,col=colorinos.bipolar[1],pch=20,size=10)+

p3<-ggplot(data=info,aes(y=Assembly.Size,x=Species))+geom_col(aes(fill=colorinos.bipolar[cluster]),alpha

p4<-ggplot(data=info,aes(y=Num.Genes,x=Species))+geom_col(aes(fill=colorinos.bipolar[cluster]),alpha=0.5

p5<-ggplot(data=info,aes(y=Unique.Proteins,x=Species))+geom_col(aes(fill=colorinos.bipolar[cluster]),al

   p6<-ggplot(data=info,aes(y=Scaffold.N50,x=Species))+geom_col(aes(fill=colorinos.bipolar[cluster]),al
```

```
p2  %>% insert_left(p,width = 2) %>% insert_right(p3 +scale_y_reverse(),width = 0.5) %>% insert_right(p
```

```
## Warning: Removed 26 rows containing missing values (geom_text).
```



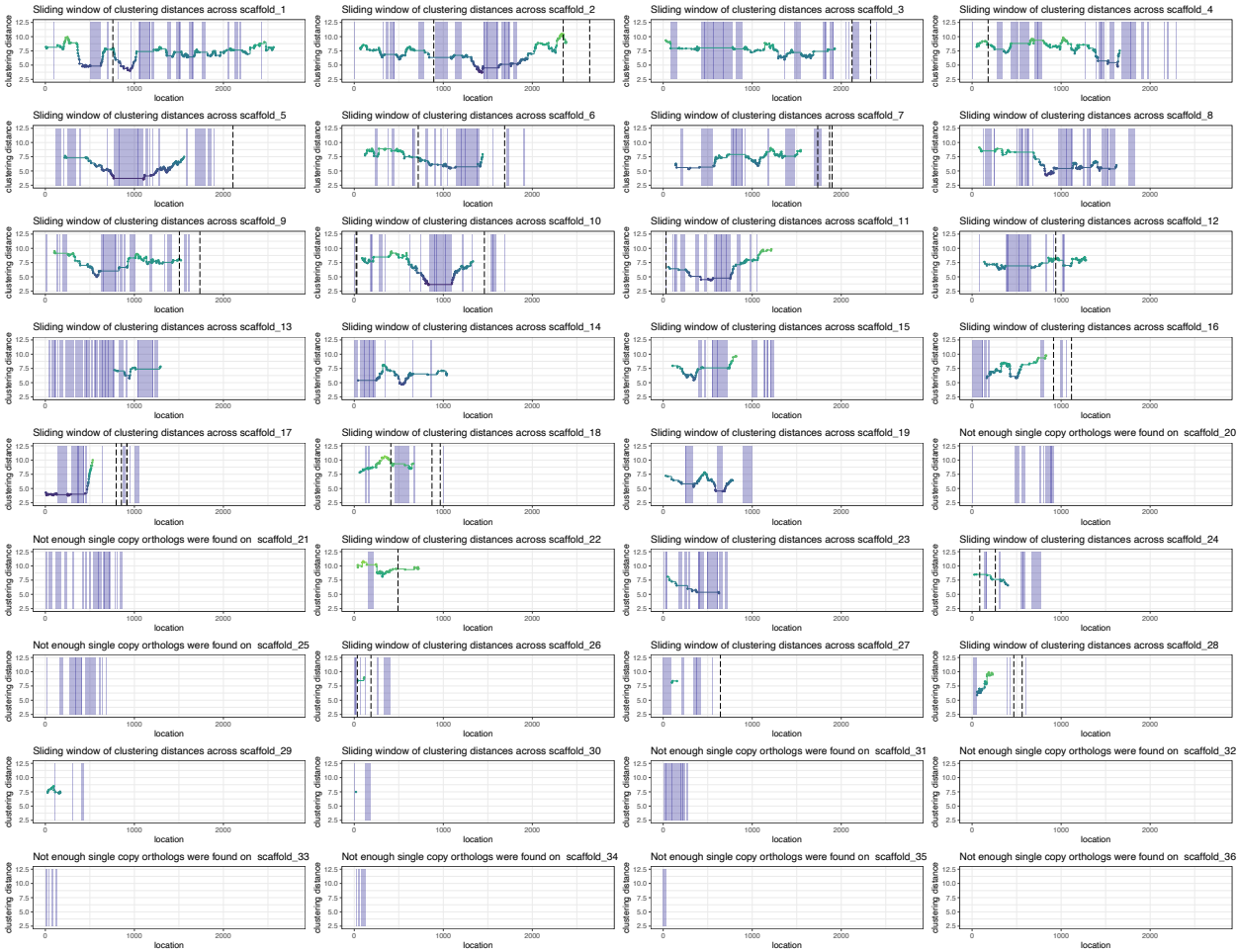
5.3 Phylogenetic concordance between loci

```
logfile<-NULL
swindow<-NULL
plots<-list()
limite<-length(pe_genome[[1]])/1000
step=20
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2))
  {
    foo.dist<-as.matrix(distances)[genes$scaffold==SCAF,genes$scaffold==SCAF]
    foo.loc<-as.numeric(as.character(genes$start[genes$scaffold==SCAF]))/1000
    if(dim(foo.dist)[1]>=21)
    {
      for (i in 1:(dim(foo.dist)[1]-step))
        {
        swindow<-c(swindow,mean(foo.dist[c(i:(i+step)),c(i:(i+step))]))
        }
        logfile<-rbind(logfile,cbind(swindow,foo.loc[1:length(swindow)],as.numeric(sapply(strsplit(SCAF
        culo<-lrar[lrar$Name==SCAF,]
        #culo2<-rip[lrar$Name==SCAF,]
        p<-ggplot(
          data.frame(mean_dist=swindow,x=foo.loc[1:length(swindow)]),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(2.5,12.5)) +
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=2.5,ymax=12.5,alpha = .3,fill = s
          geom_point(size=0.5) +
          geom_step() +
          scale_color_continuous(type = "viridis", limits = range(2.5, 12.5))+
          theme_bw() +
          labs (title=paste("Sliding window of clustering distances across",SCAF),x="location",y="cluste
    if (length(unlist(het[het[,1]==SCAF,]))!=0)
    {
          plots[[SCAF]]<-p+geom_vline(xintercept = as.numeric(as.character(het[het[,1]==SCAF,2]))/1000,
    }else{
    plots[[SCAF]]<-p + theme(legend.position = "none")
    }
}
#annotate("path",x=culo2$Start[order(culo2$Start)]/1000,y=culo2$GC.Content[order(culo2$Start)]/10, ymax
```

```
swindow<-NULL
}
logfile_distances<-logfile
```

```
pempty<-list()
for (SCAF in c("scaffold_20","scaffold_21","scaffold_25","scaffold_31","scaffold_32","scaffold_33","sca
  {
    culo<-lrar[lrar$Name==SCAF,]
    pempty[[SCAF]]<-ggplot(
          data.frame(mean_dist=c(5,5,5),x=c(0,1000,10000)),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(2.5,12.5))+theme_bw()+
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=2.5,ymax=12.5,alpha = .3,fill = s
          labs (title=paste("Not enough single copy orthologs were found on ",SCAF),x="location",y="clus
}
plots[[1]]/plots[[5]]/plots[[9]]/plots[[13]]/plots[[17]]/pempty[[2]]/pempty[[3]]/plots[[26]]/pempty[[6]]
plots[[2]]/plots[[6]]/plots[[10]]/plots[[14]]/plots[[18]]/plots[[20]]/plots[[23]]/plots[[27]]/pempty[[7]
plots[[3]]/plots[[7]]/plots[[11]]/plots[[15]]/plots[[19]]/plots[[21]]/plots[[24]]/pempty[[4]]/pempty[[8]
plots[[4]]/plots[[8]]/plots[[12]]/plots[[16]]/pempty[[1]]/plots[[22]]/plots[[25]]/pempty[[5]]/pempty[[9]
```

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```



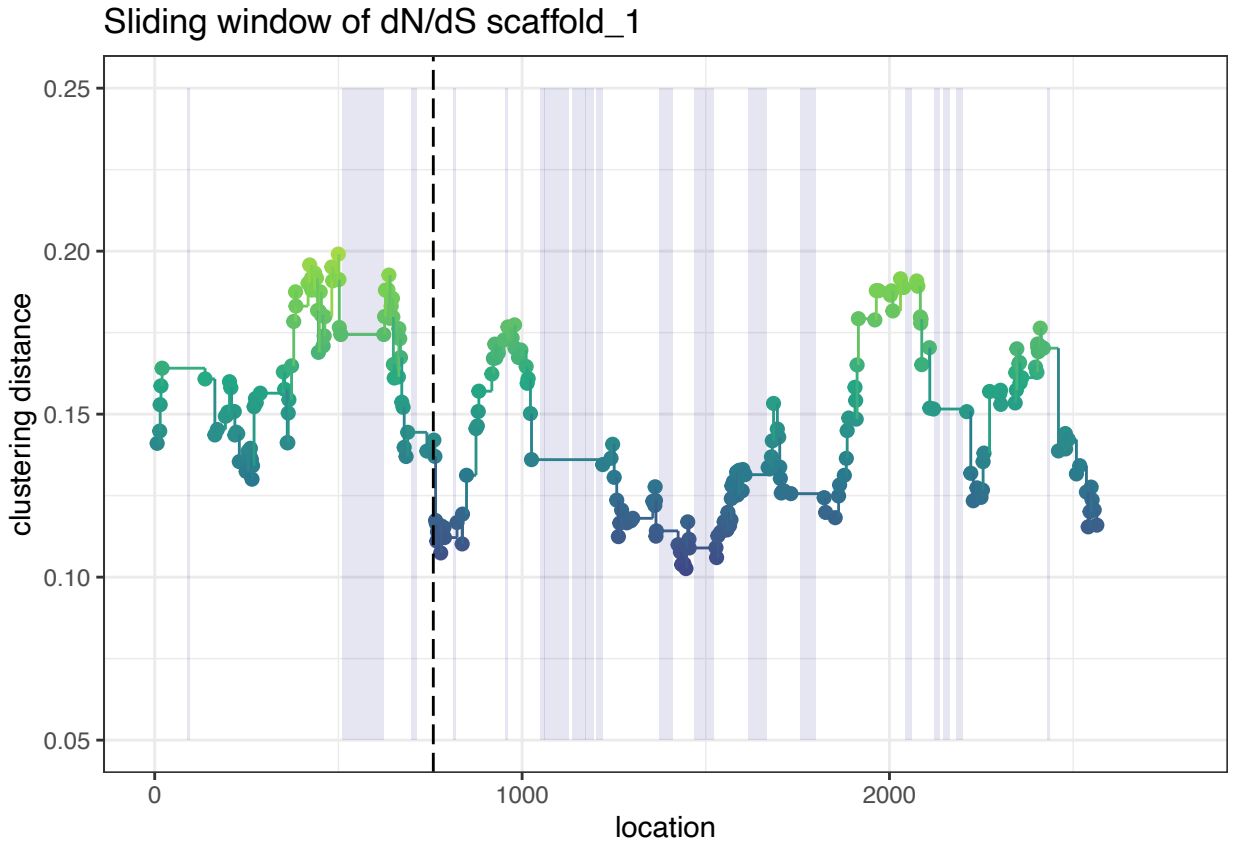
5.4 Mapping dN/dS ratios along the genome

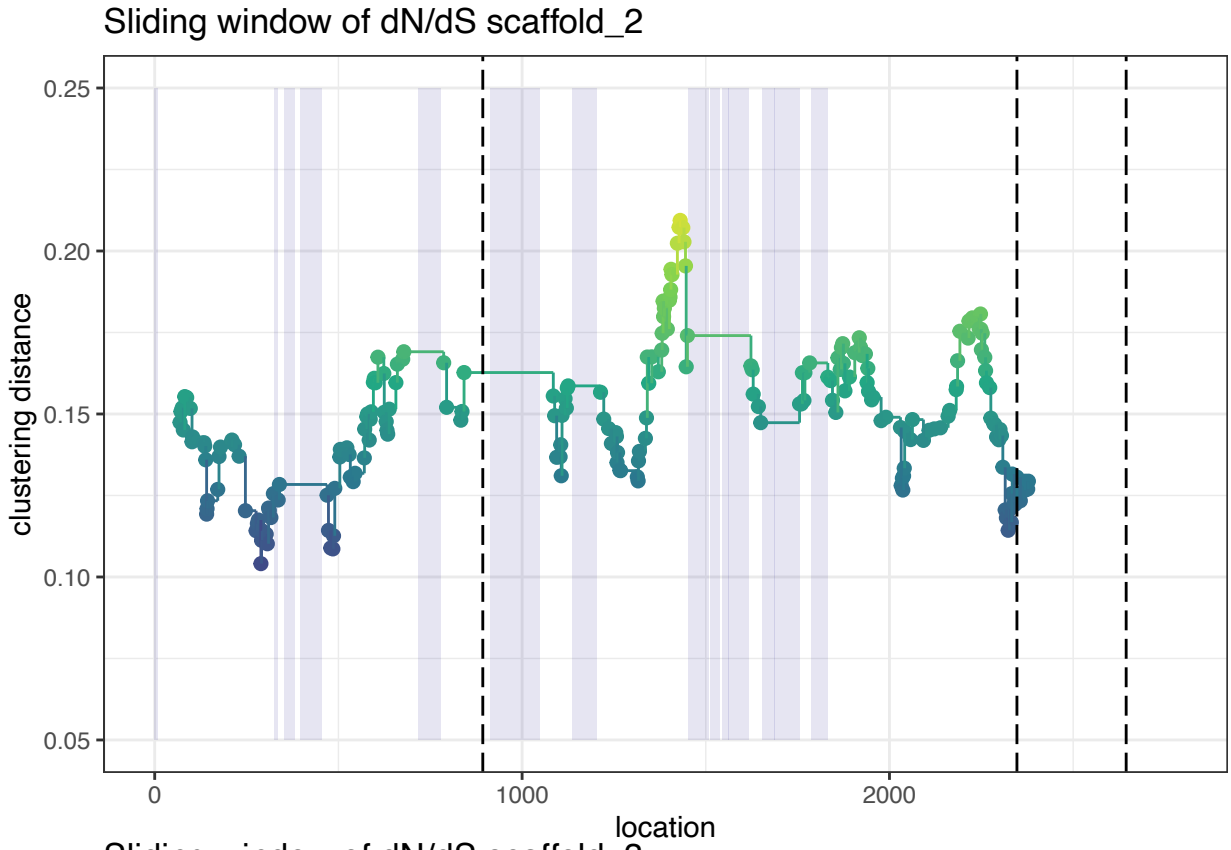
5.4.1 Sliding window of dN/dS

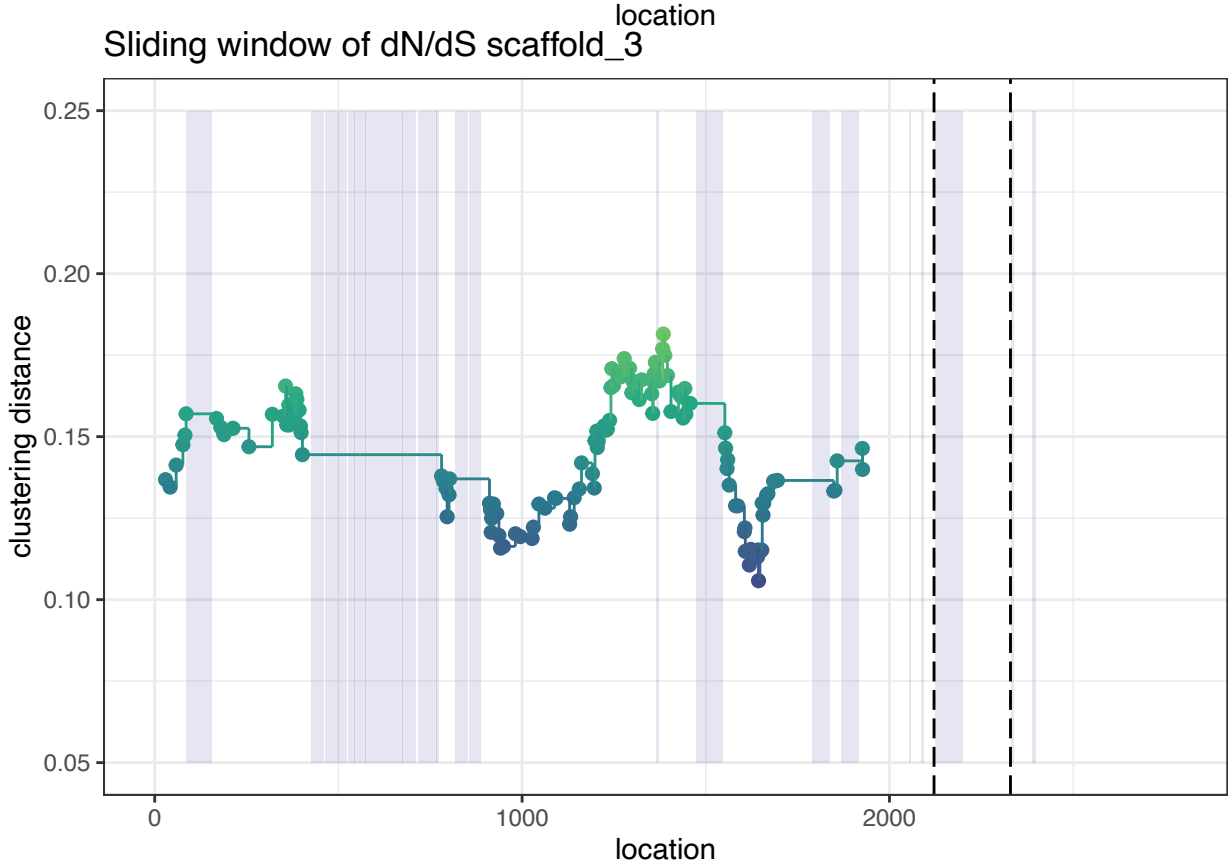
```
logfile<-NULL
swindow<-NULL
step=20
foo.genes<-genes[rownames(ogs),]
plots<-list()
limite<-length(pe_genome[[1]])/1000
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2)
  {
    foo.dnds<-ogs$dn_ds[foo.genes$scaffold==SCAF]
    foo.loc<-as.numeric(as.character(foo.genes$start[foo.genes$scaffold==SCAF]))/1000
    if(length(foo.dnds)>=21)
    {
      for (i in 1:(length(foo.dnds)-step))
        {
        swindow<-c(swindow,mean(foo.dnds[c(i:(i+step))]))
        }
        logfile<-rbind(logfile,cbind(swindow,foo.loc[1:length(swindow)],as.numeric(sapply(strsplit(SCAF
        culo<-lrar[lrar$Name==SCAF,]
        p<-ggplot(
          data.frame(mean_dnds=swindow,x=foo.loc[1:length(swindow)]),
          aes(x=x, y=mean_dnds, color=mean_dnds)) +
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0.05,ymax=0.25,alpha =.1,fill = s
          geom_point(size=2) +
          geom_step() +
          scale_color_continuous(type = "viridis", limits = range(0.07,0.22)) +
          xlim(c(0,limite)) + ylim(c(0.05,0.25)) +
          theme_bw() +
          labs (title=paste("Sliding window of dN/dS",SCAF),x="location",y="clustering distance")
    if (length(unlist(het[het[,1]==SCAF,]))!=0)
    {
          plots[[SCAF]]<- p + geom_vline(xintercept = as.numeric(as.character(het[het[,1]==SCAF,2]))/10
    }else{
    plots[[SCAF]]<-p + theme(legend.position = "none")
      }
    }
    swindow<-NULL
}
logfile_dnds<-logfile
```

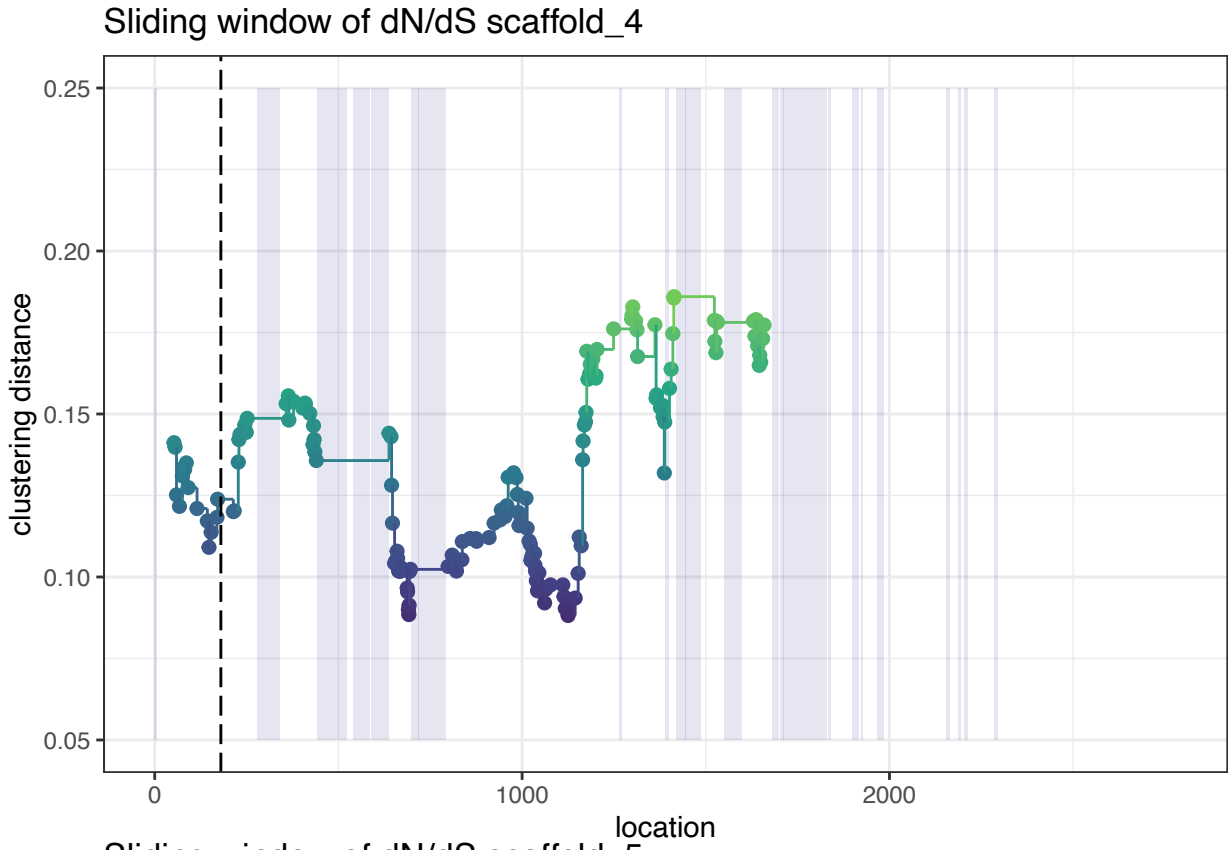
```
#, fig.width=22,fig.height=17}
pempty<-list()
for (SCAF in c("scaffold_20","scaffold_21","scaffold_25","scaffold_31","scaffold_32","scaffold_33","sca
  {
    culo<-lrar[lrar$Name==SCAF,]
    pempty[[SCAF]]<-ggplot(
          data.frame(mean_dist=c(5,5,5),x=c(0,1000,10000)),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(0.05,0.25))+theme_bw()+
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0.05,ymax=0.25,alpha = .3,fill =
          labs (title=paste("Not enough single copy orthologs were found on ",SCAF),x="location",y="clus
}
#plots[[1]]/plots[[5]]/plots[[9]]/plots[[13]]/plots[[17]]/pempty[[2]]/pempty[[3]]/plots[[26]]/pempty[[6]
```

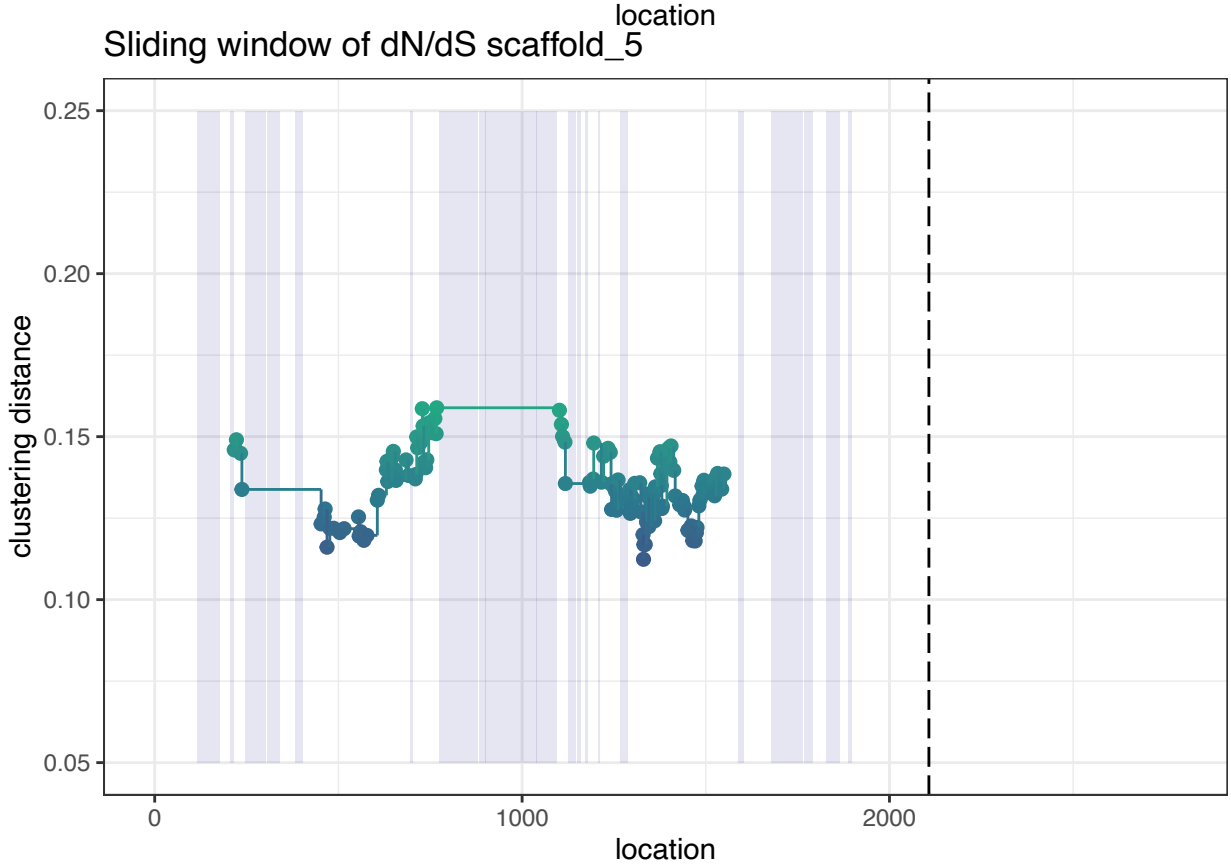
```
#plots[[2]]/plots[[6]]/plots[[10]]/plots[[14]]/plots[[18]]/plots[[20]]/plots[[23]]/plots[[27]]/pempty[[
#plots[[3]]/plots[[7]]/plots[[11]]/plots[[15]]/plots[[19]]/plots[[21]]/plots[[24]]/pempty[[4]]/pempty[[
#plots[[4]]/plots[[8]]/plots[[12]]/plots[[16]]/pempty[[1]]/plots[[22]]/plots[[25]]/pempty[[5]]/pempty[[
lapply(plots,print)
```

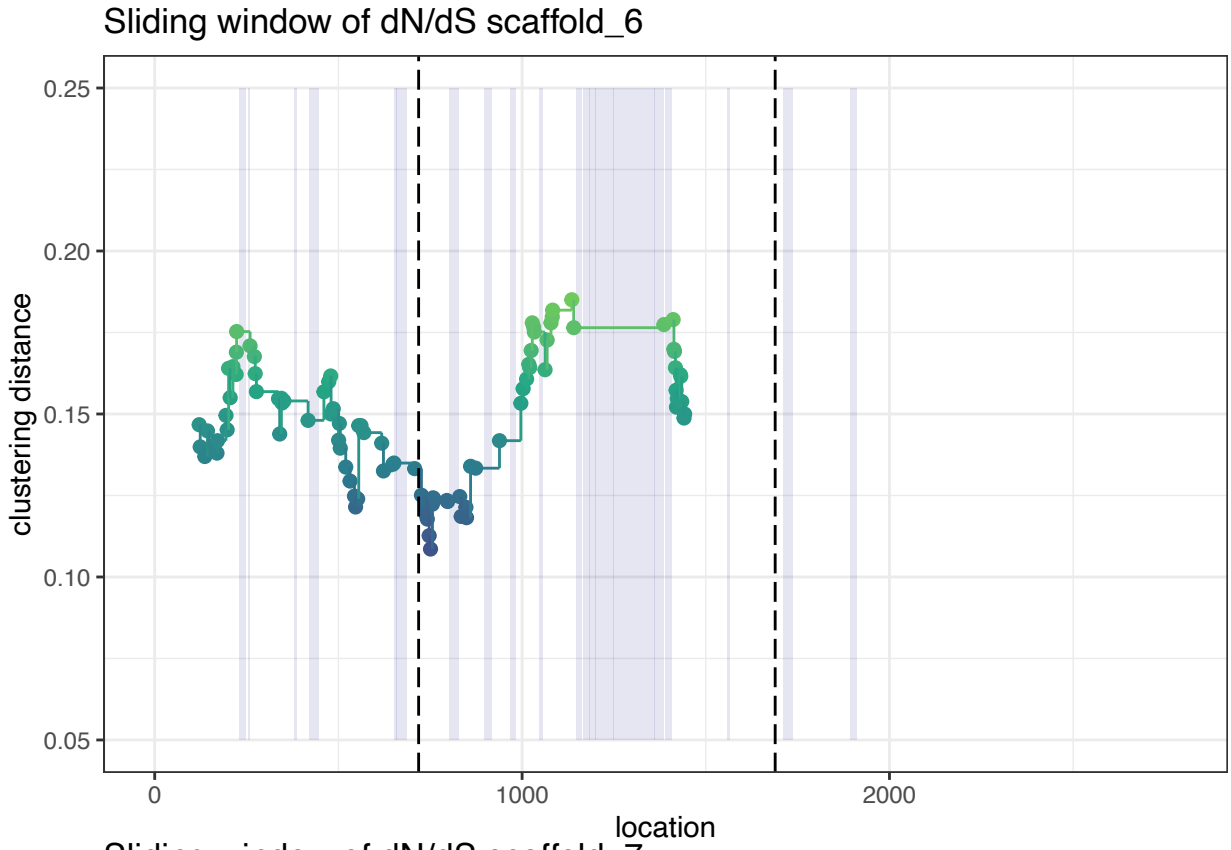


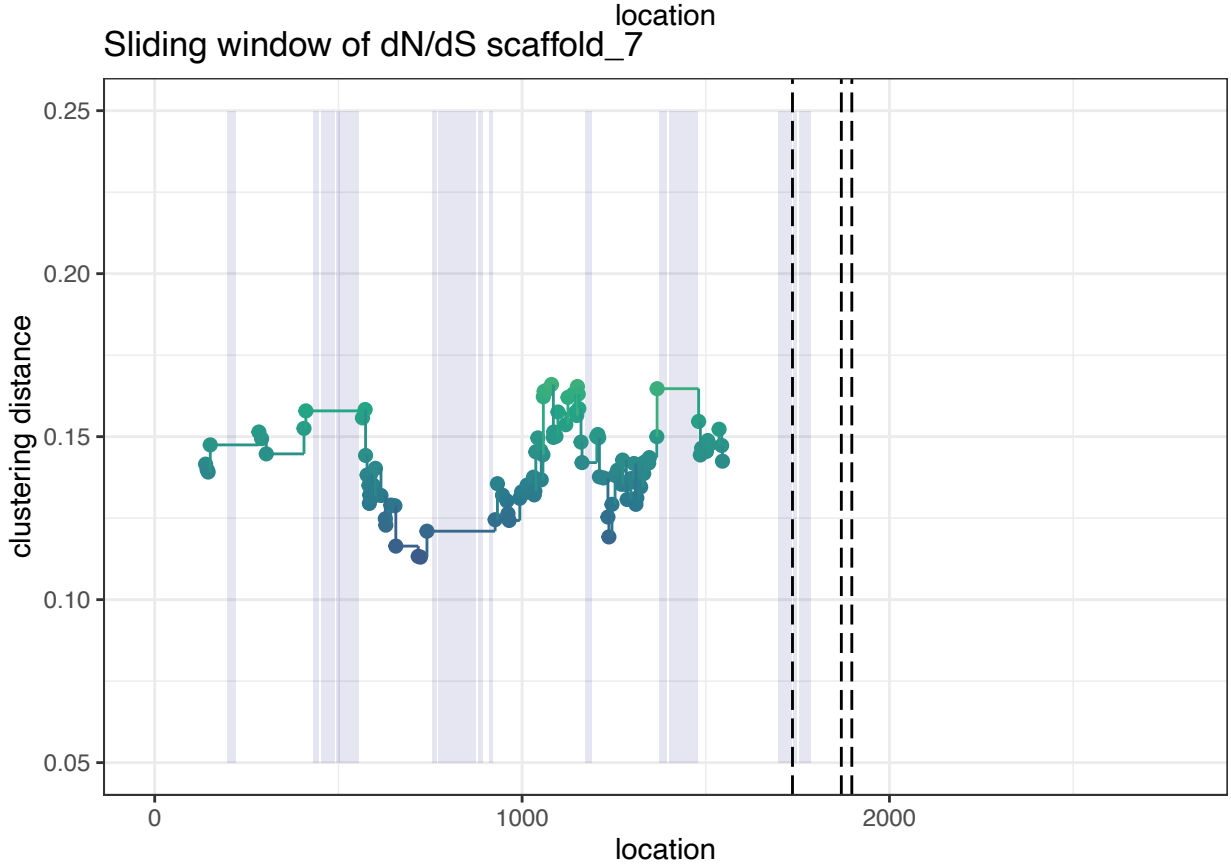

Sliding window of dN/dS scaffold_2
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

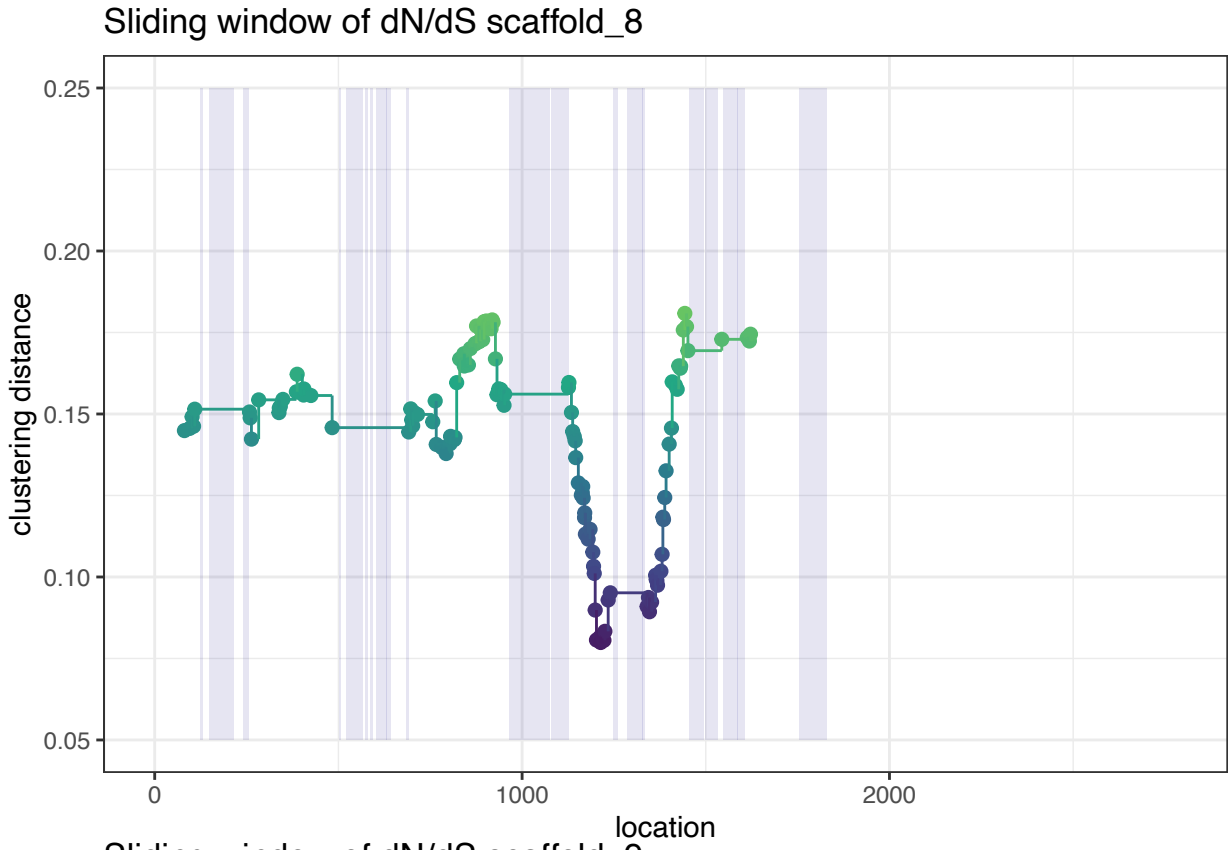

Sliding window of dN/dS scaffold_3
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

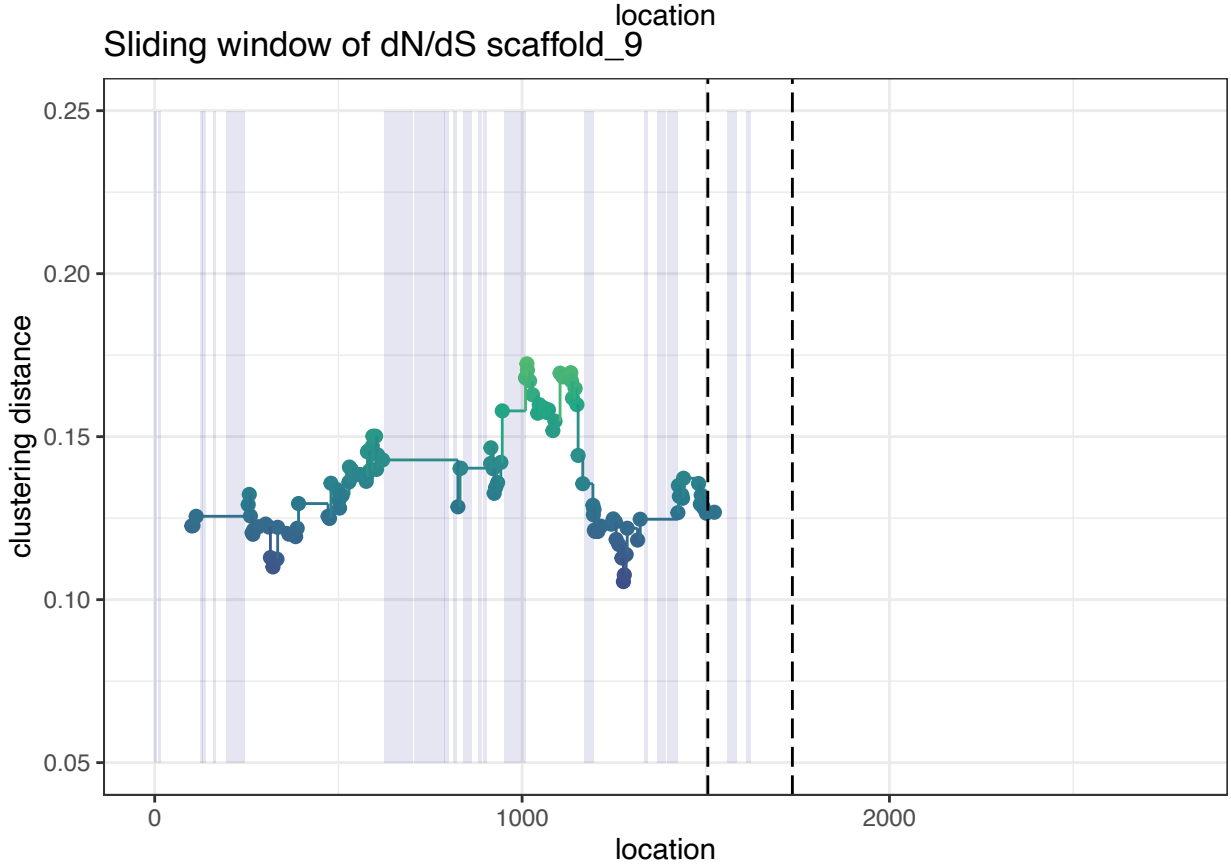

Sliding window of dN/dS scaffold_4
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

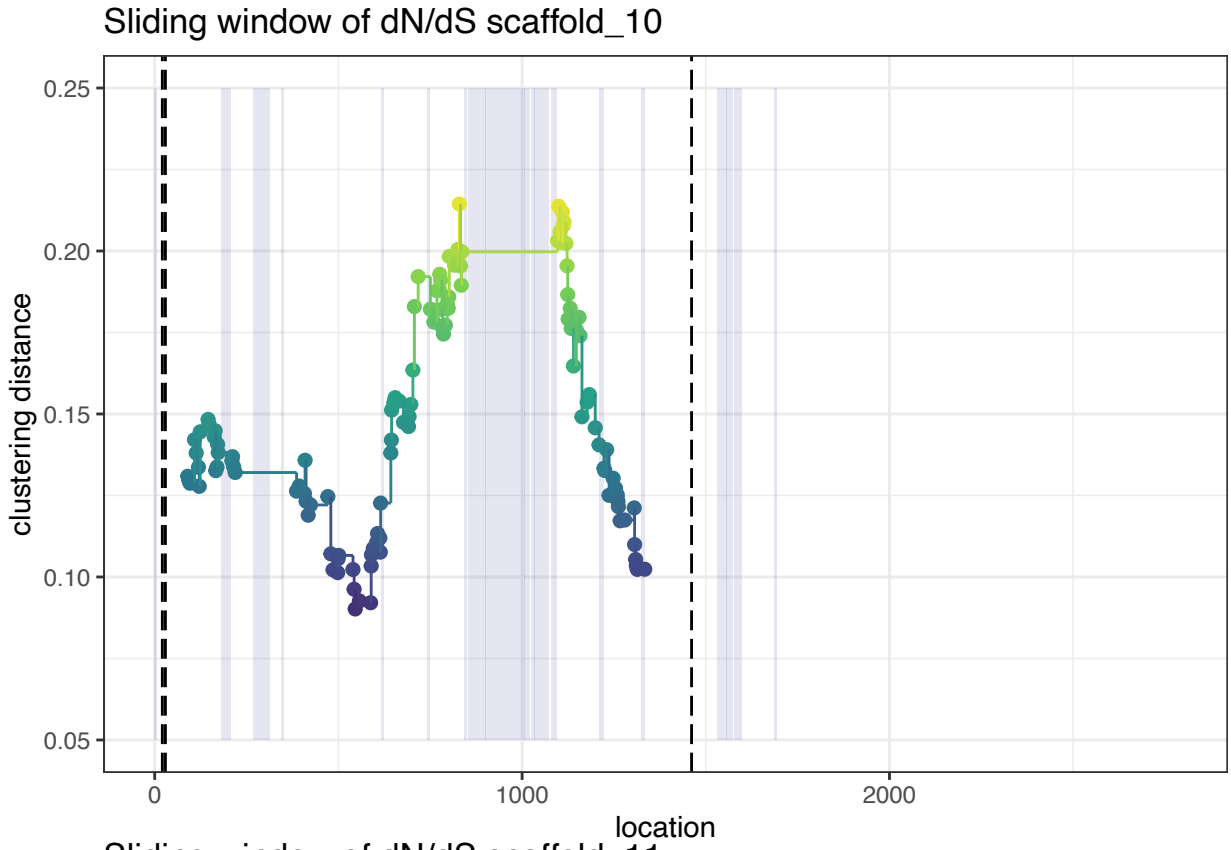

Sliding window of dN/dS scaffold_5
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

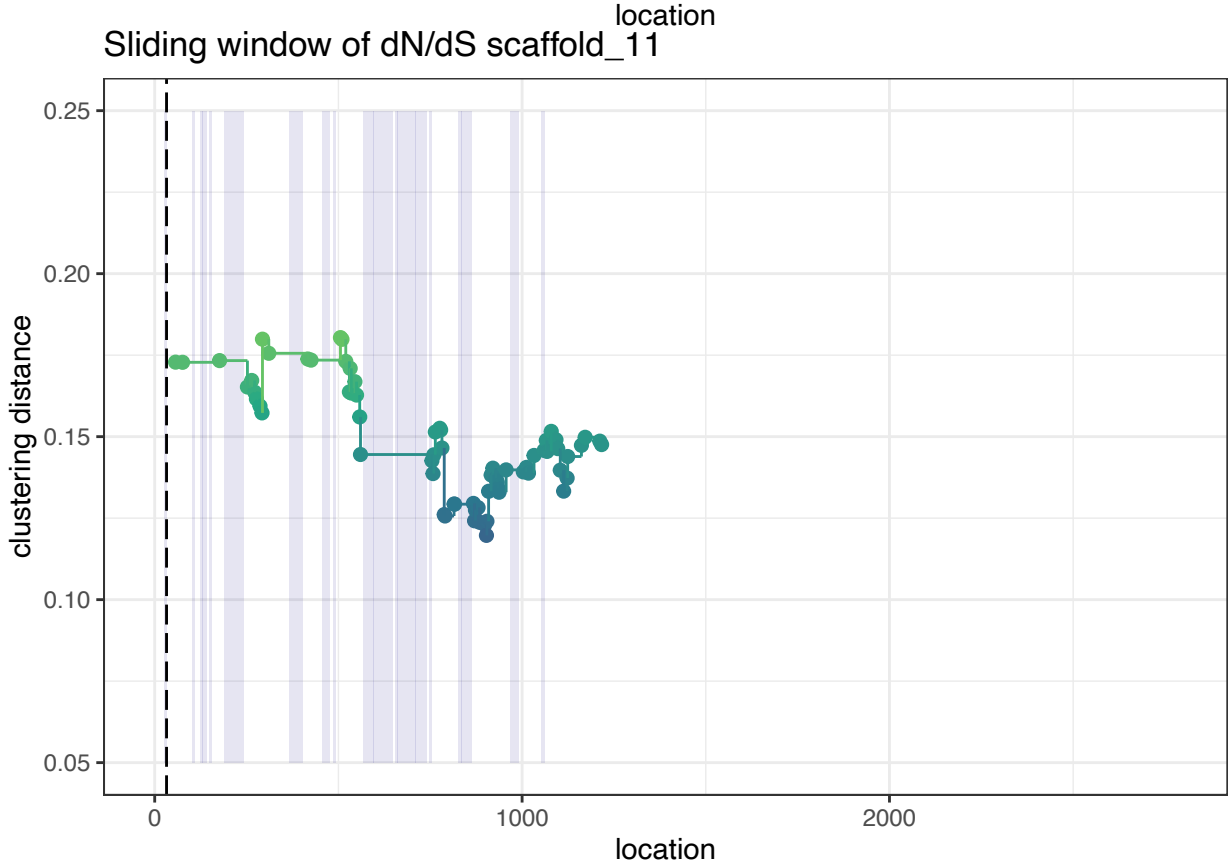

Sliding window of dN/dS scaffold_6
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

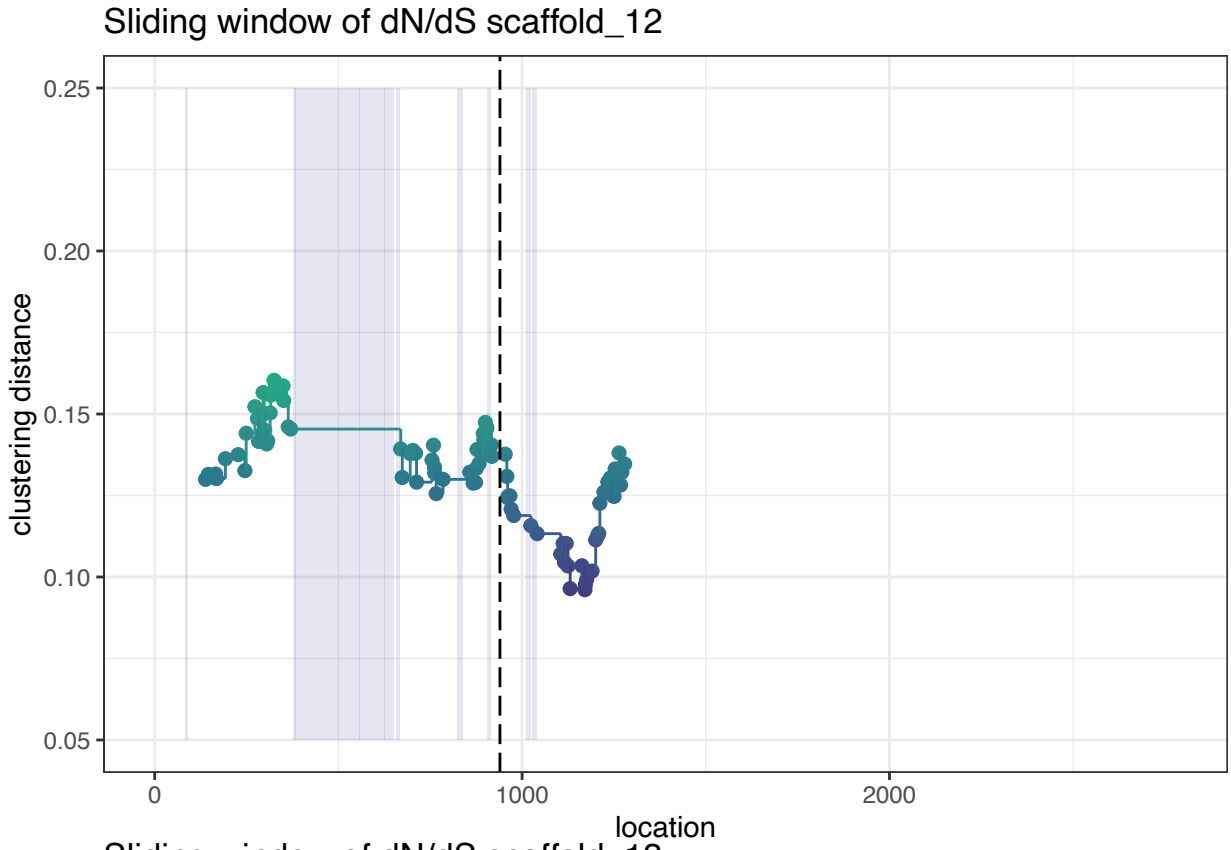

Sliding window of dN/dS scaffold_7
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

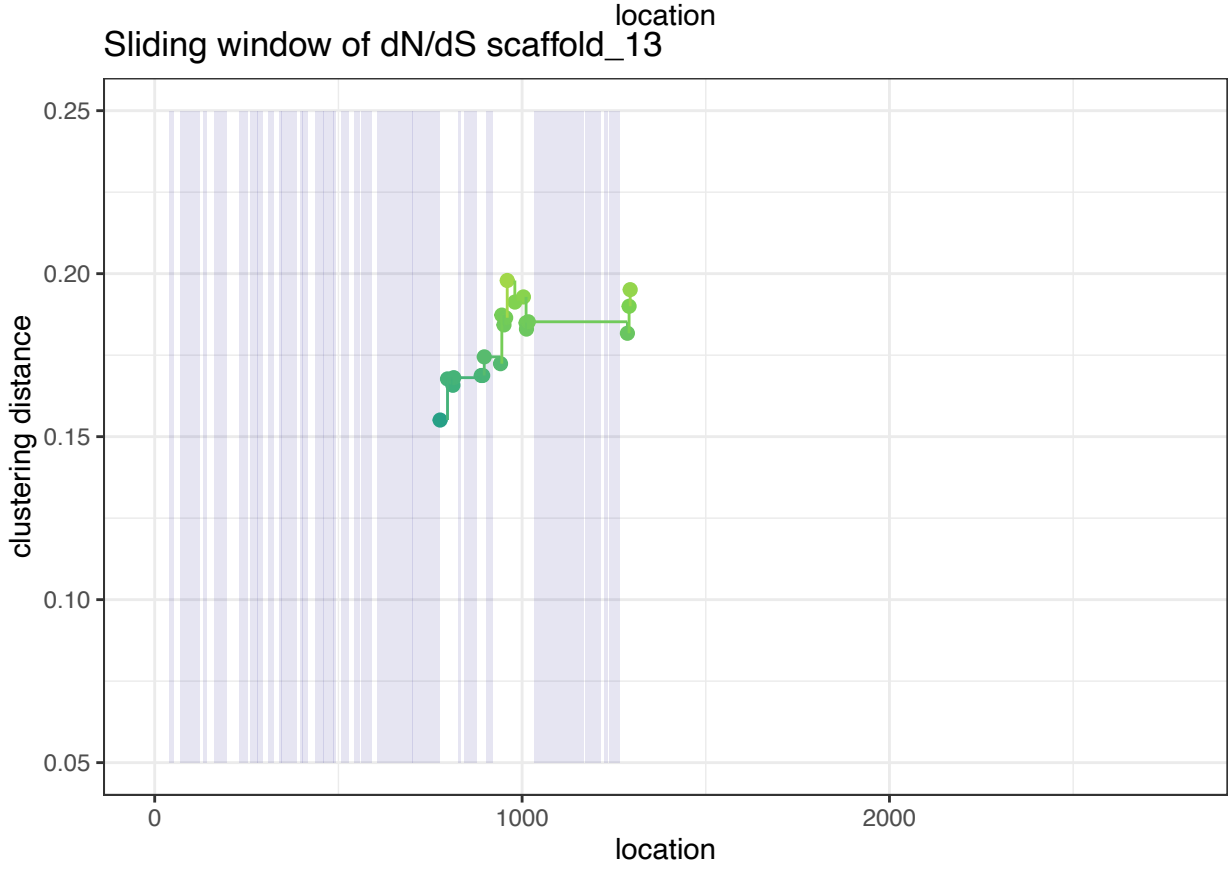

Sliding window of dN/dS scaffold_8
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

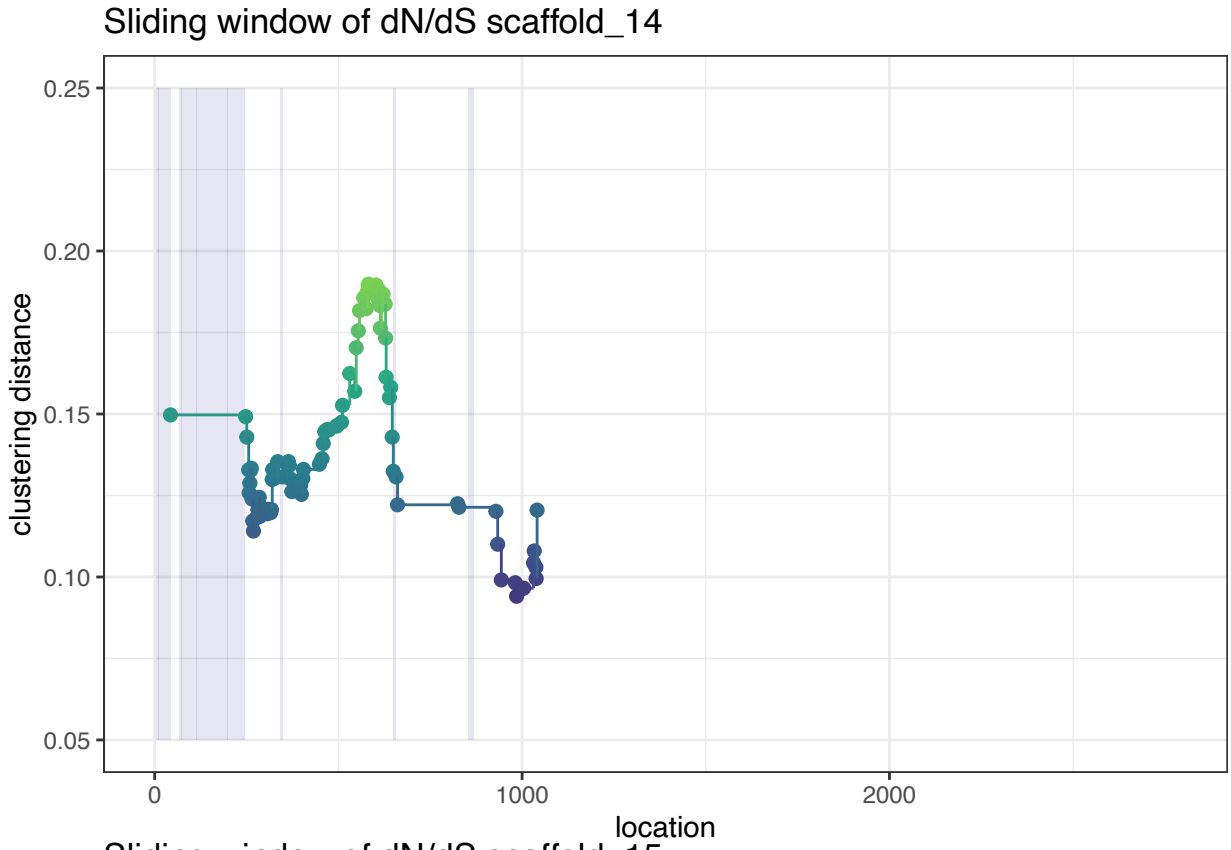

Sliding window of dN/dS scaffold_9
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

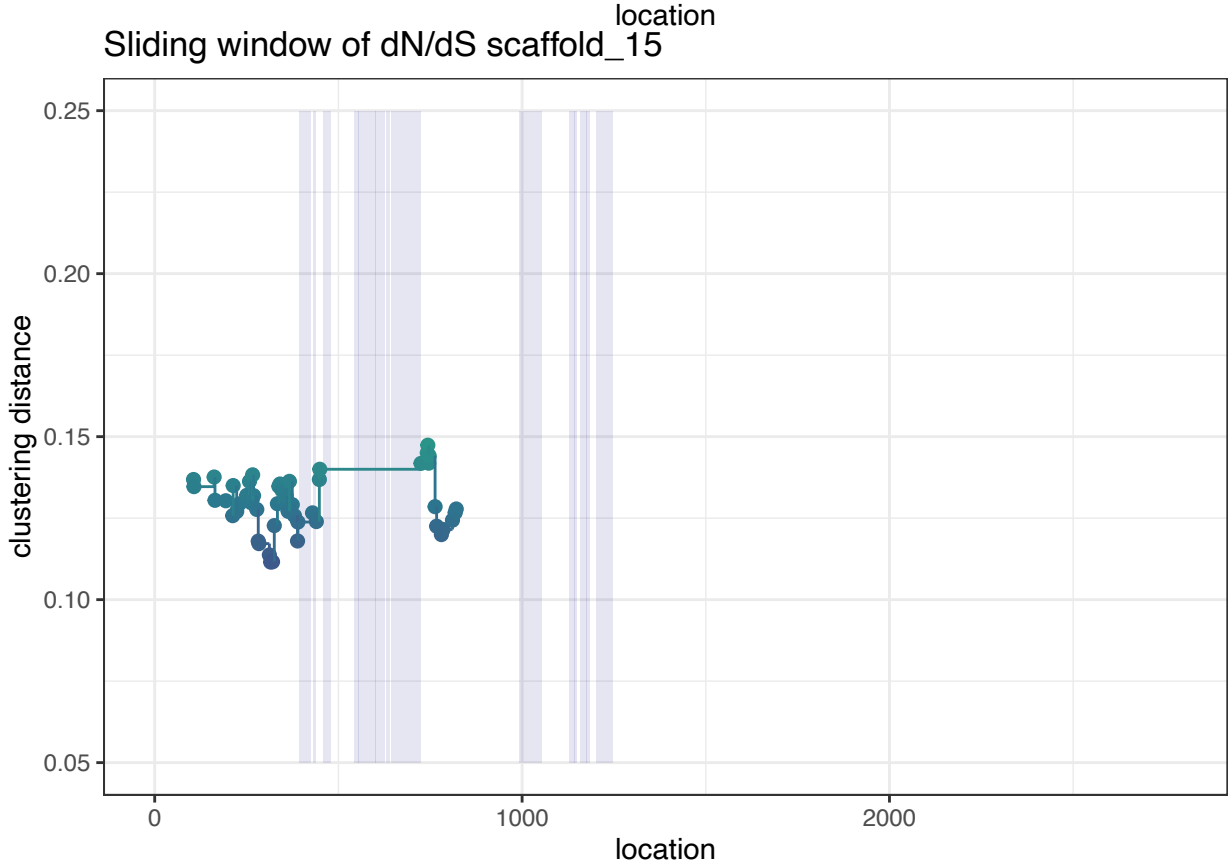

Sliding window of dN/dS scaffold_10
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

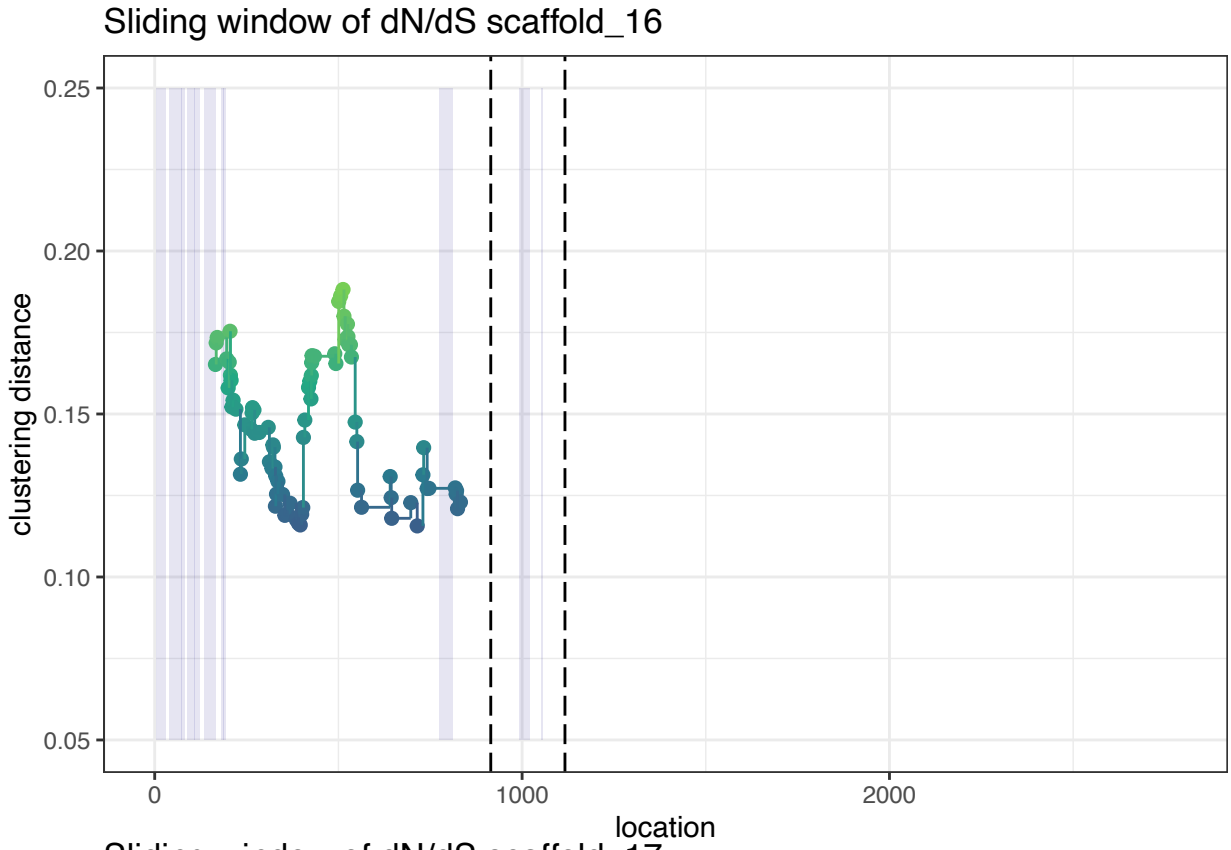

Sliding window of dN/dS scaffold_11
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

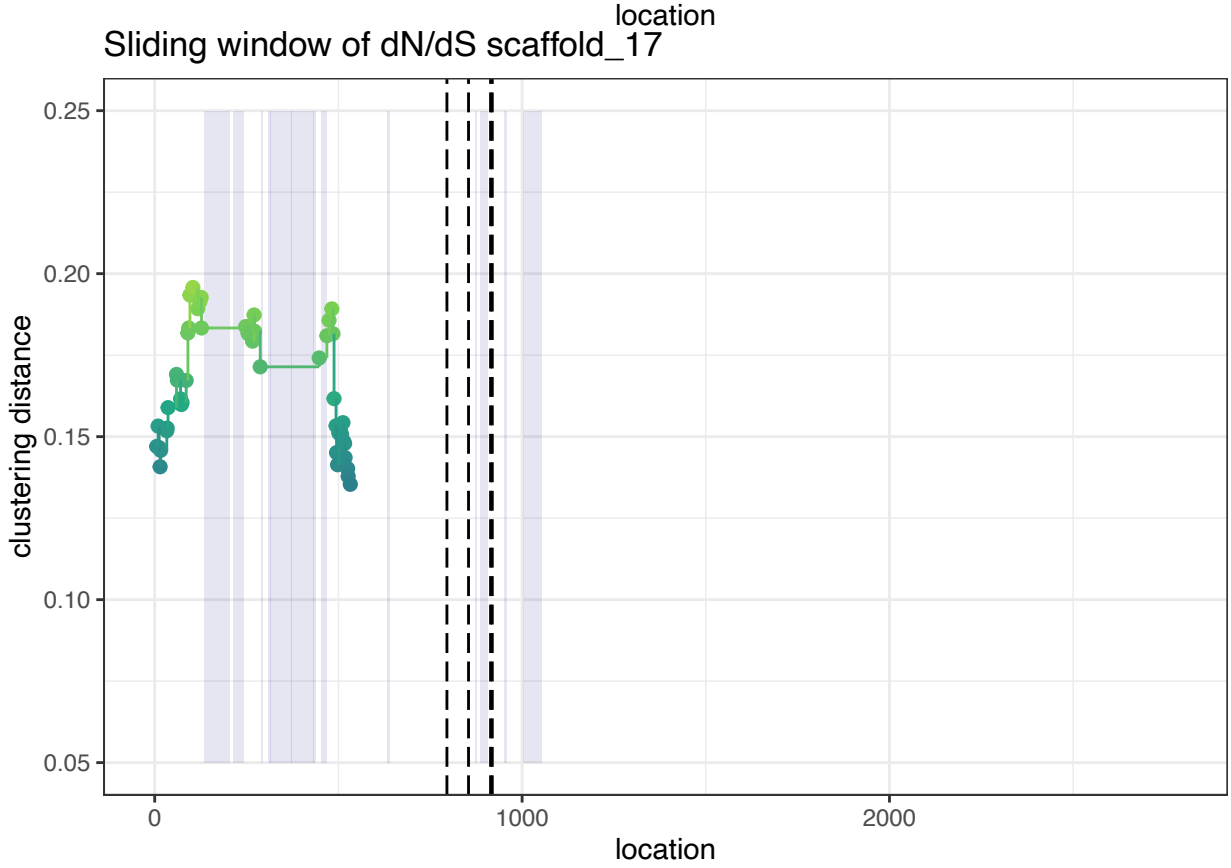

Sliding window of dN/dS scaffold_12
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

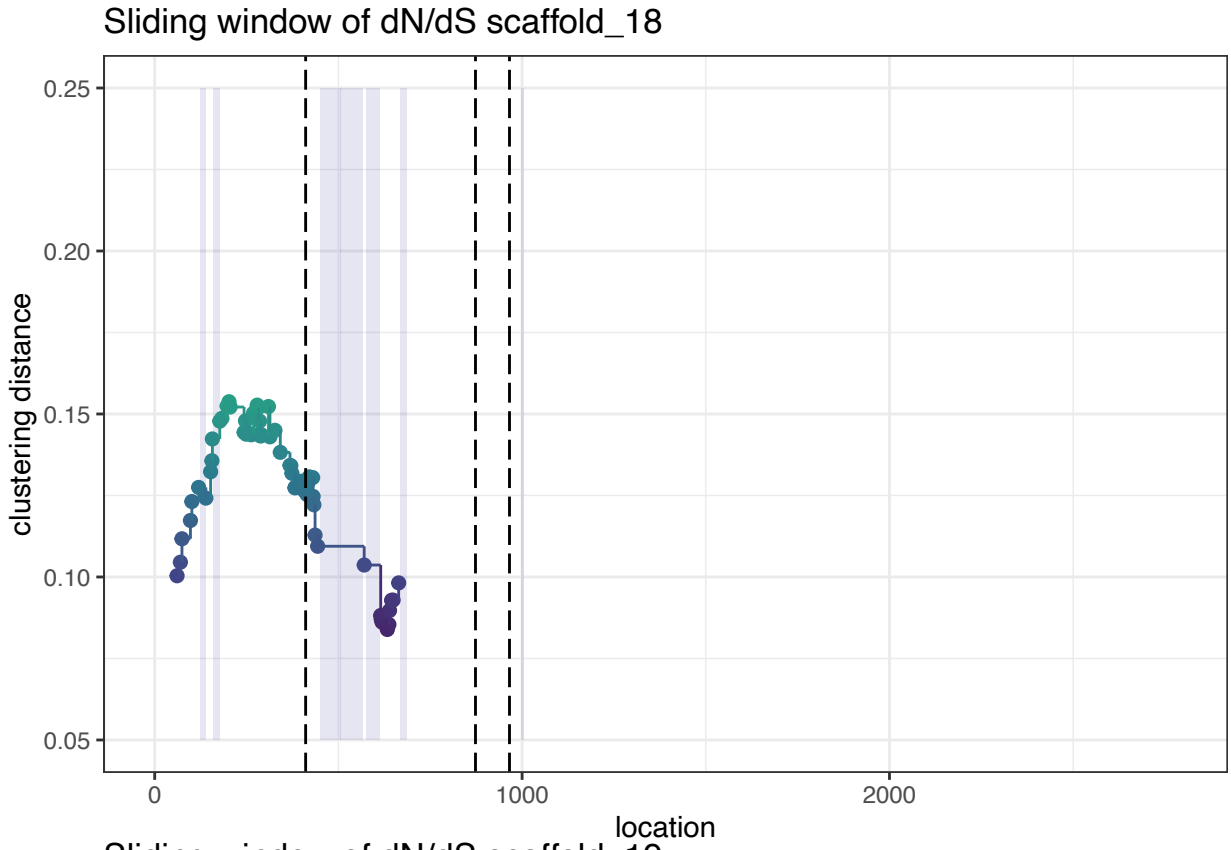

Sliding window of dN/dS scaffold_13
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

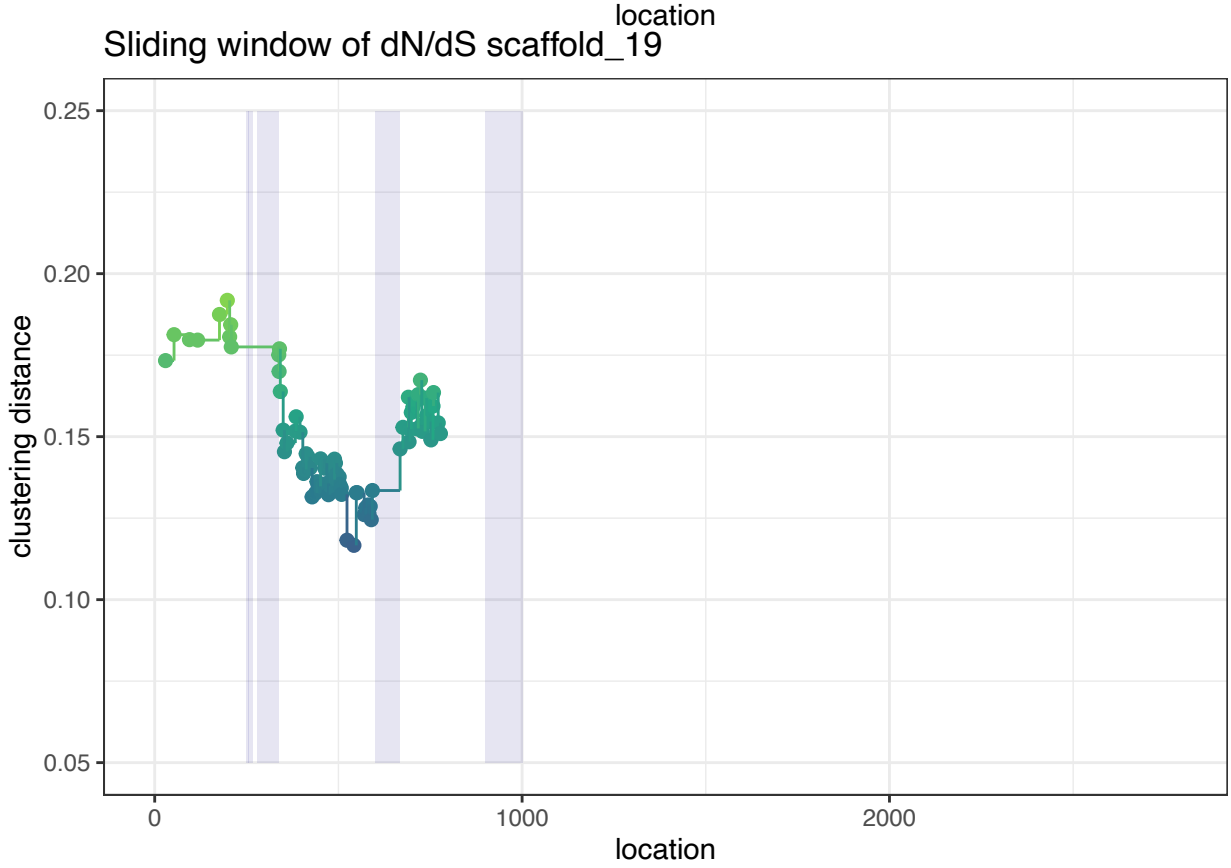

Sliding window of dN/dS scaffold_14
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

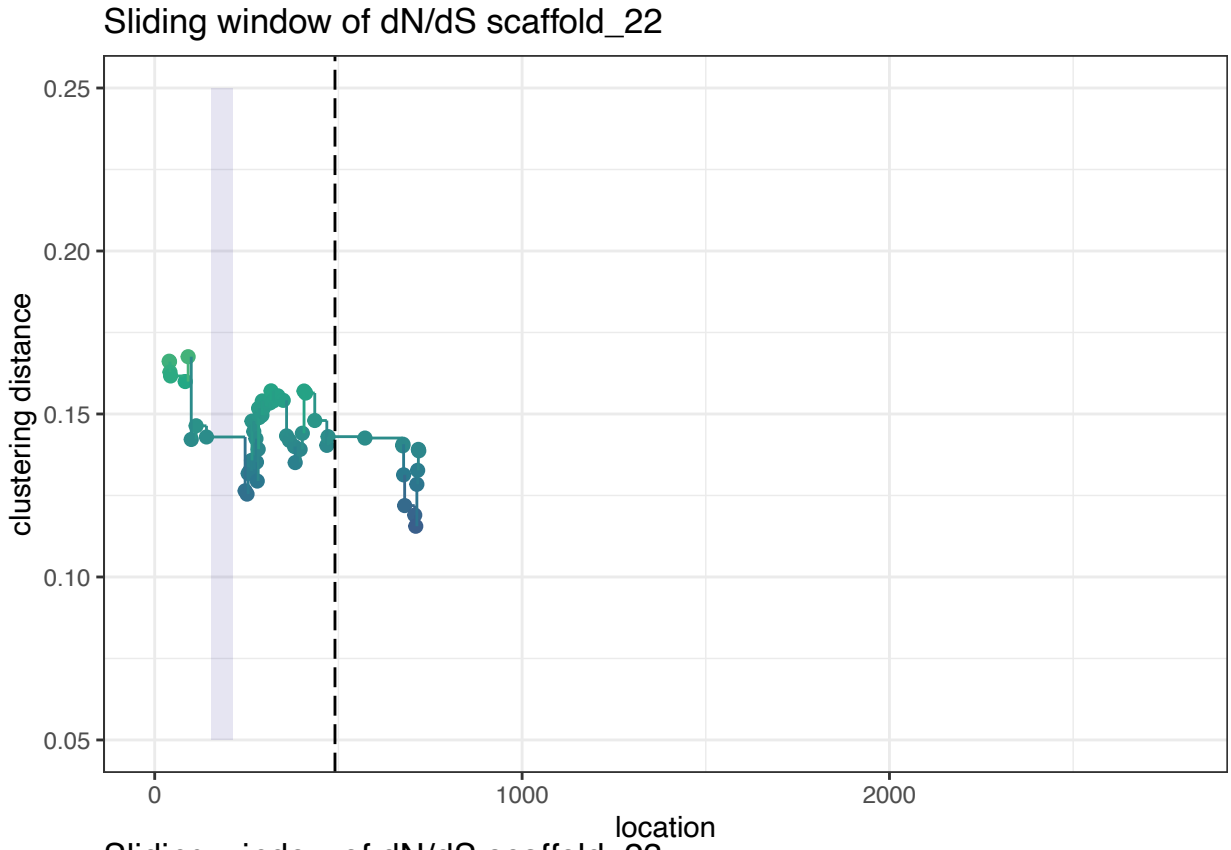

Sliding window of dN/dS scaffold_15
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

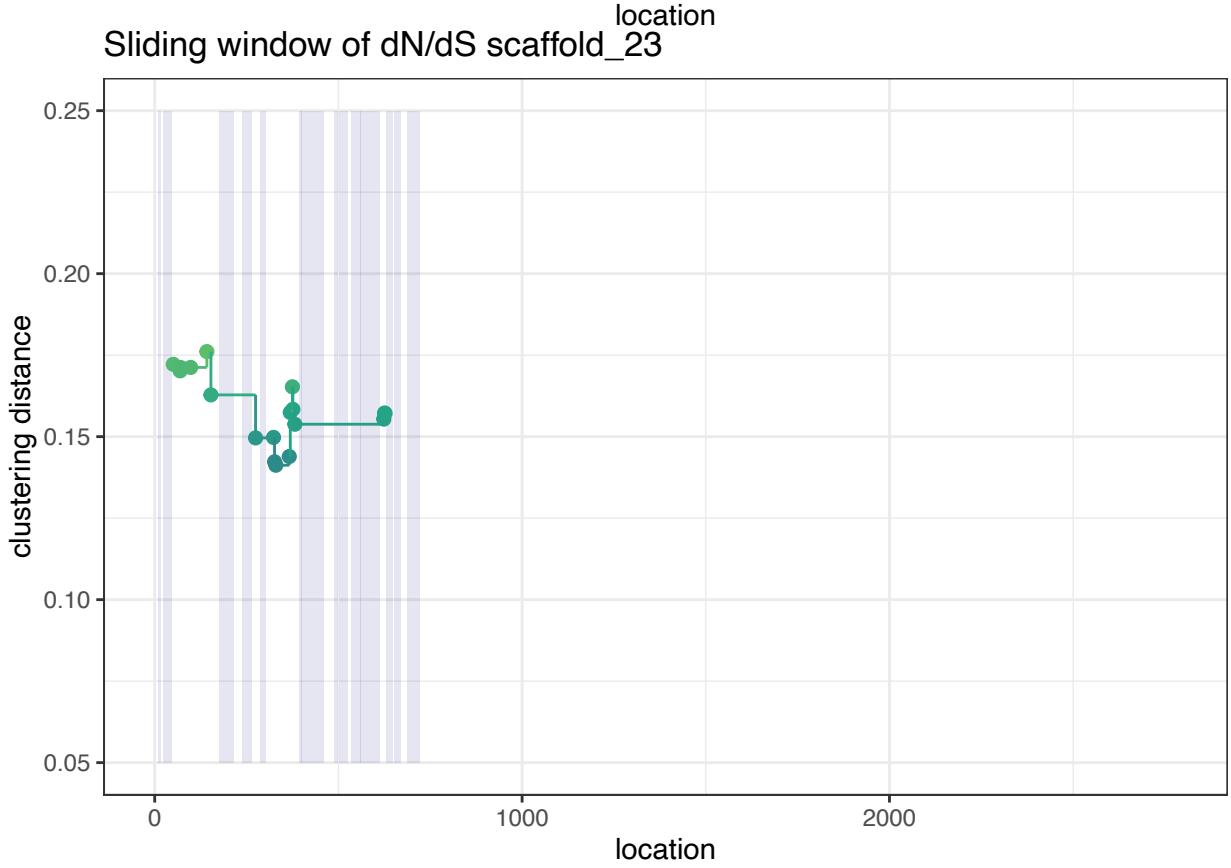

Sliding window of dN/dS scaffold_16
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

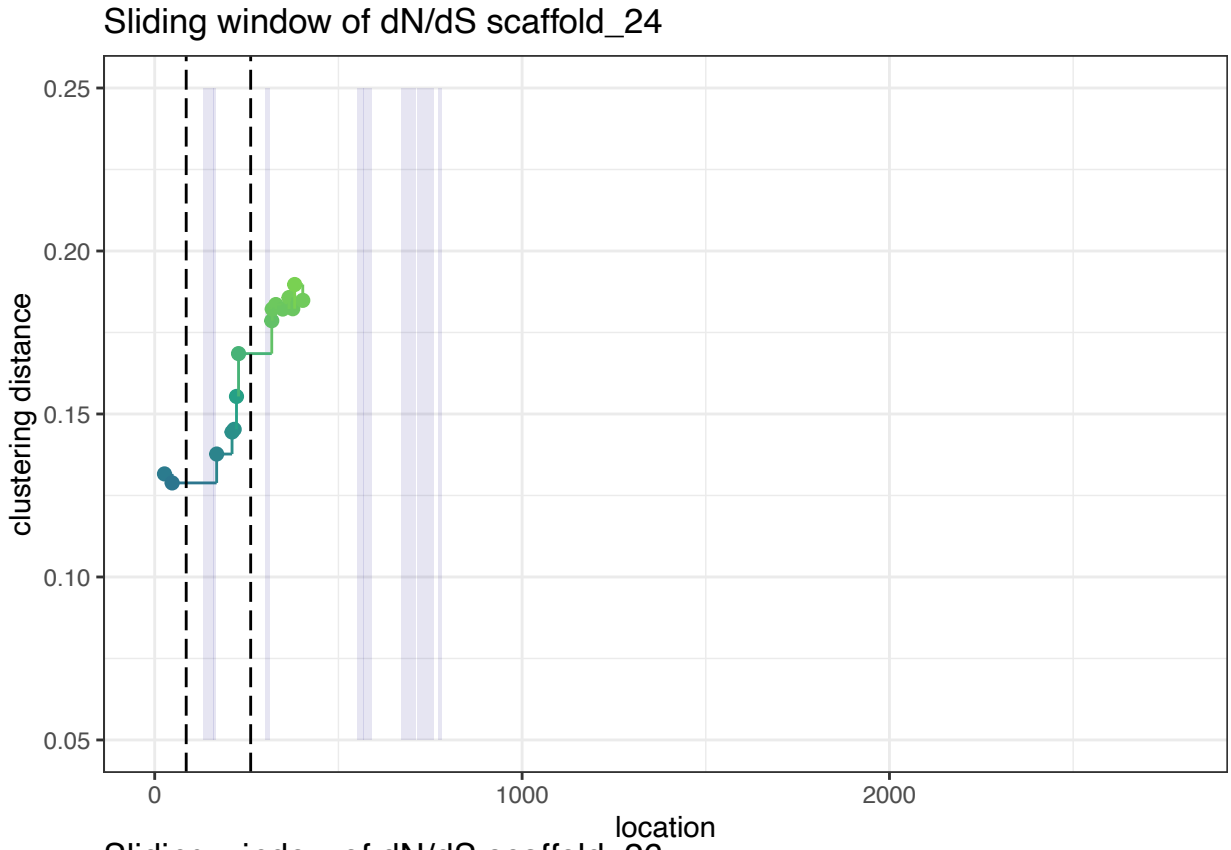

Sliding window of dN/dS scaffold_17
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

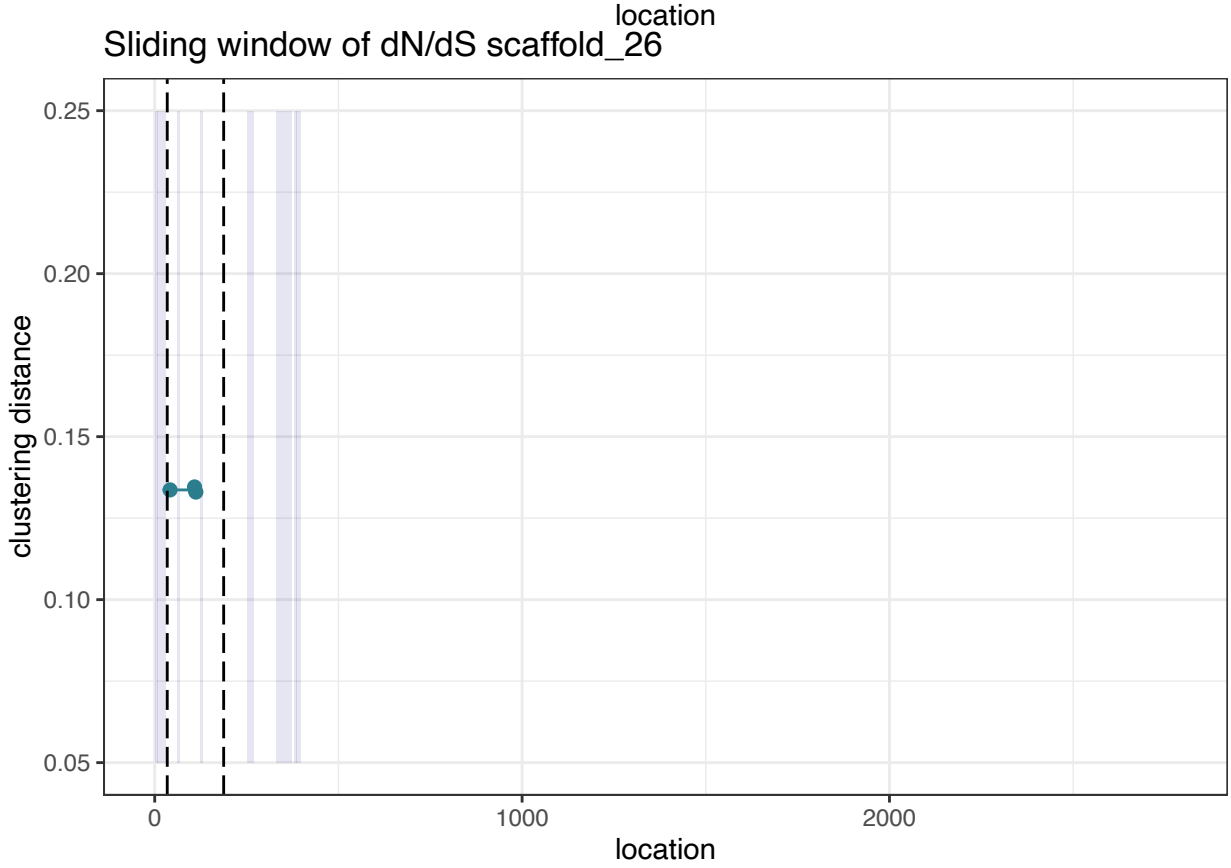

Sliding window of dN/dS scaffold_18
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

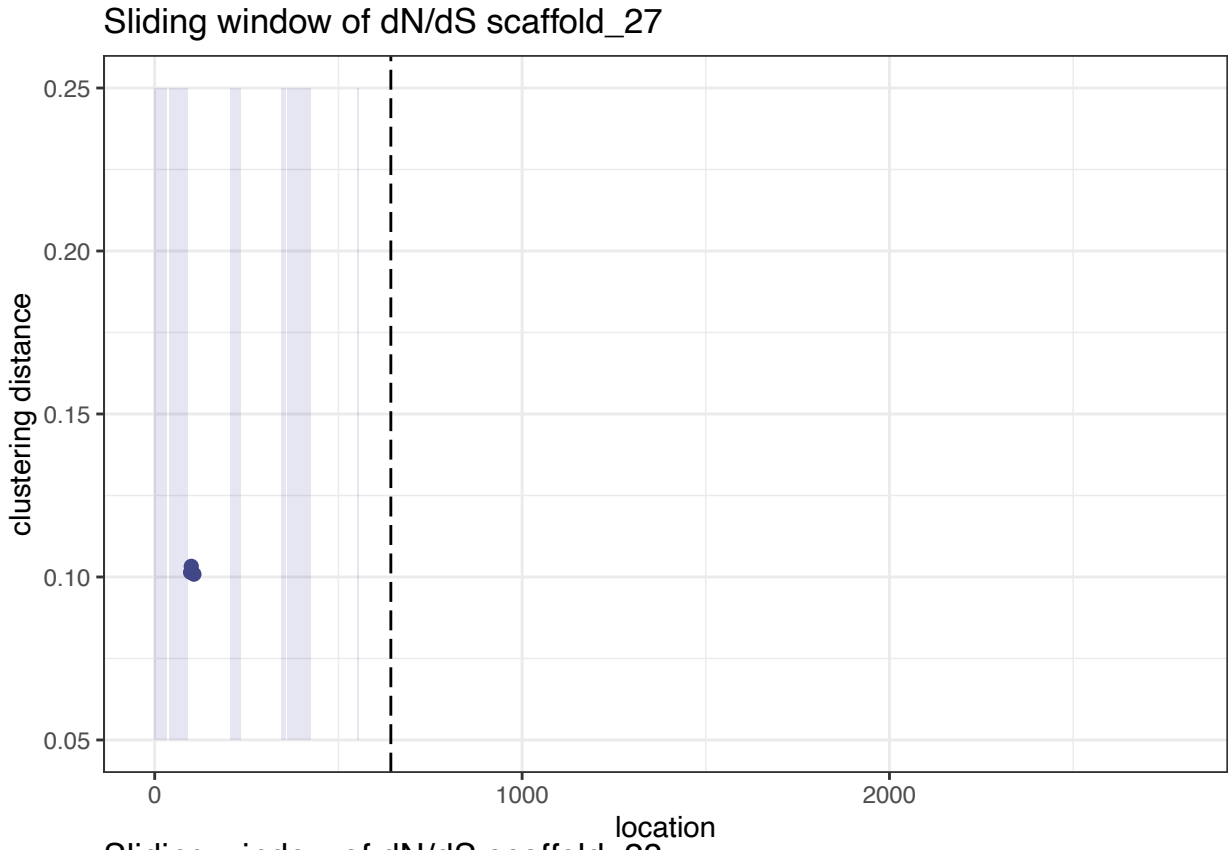

Sliding window of dN/dS scaffold_19
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

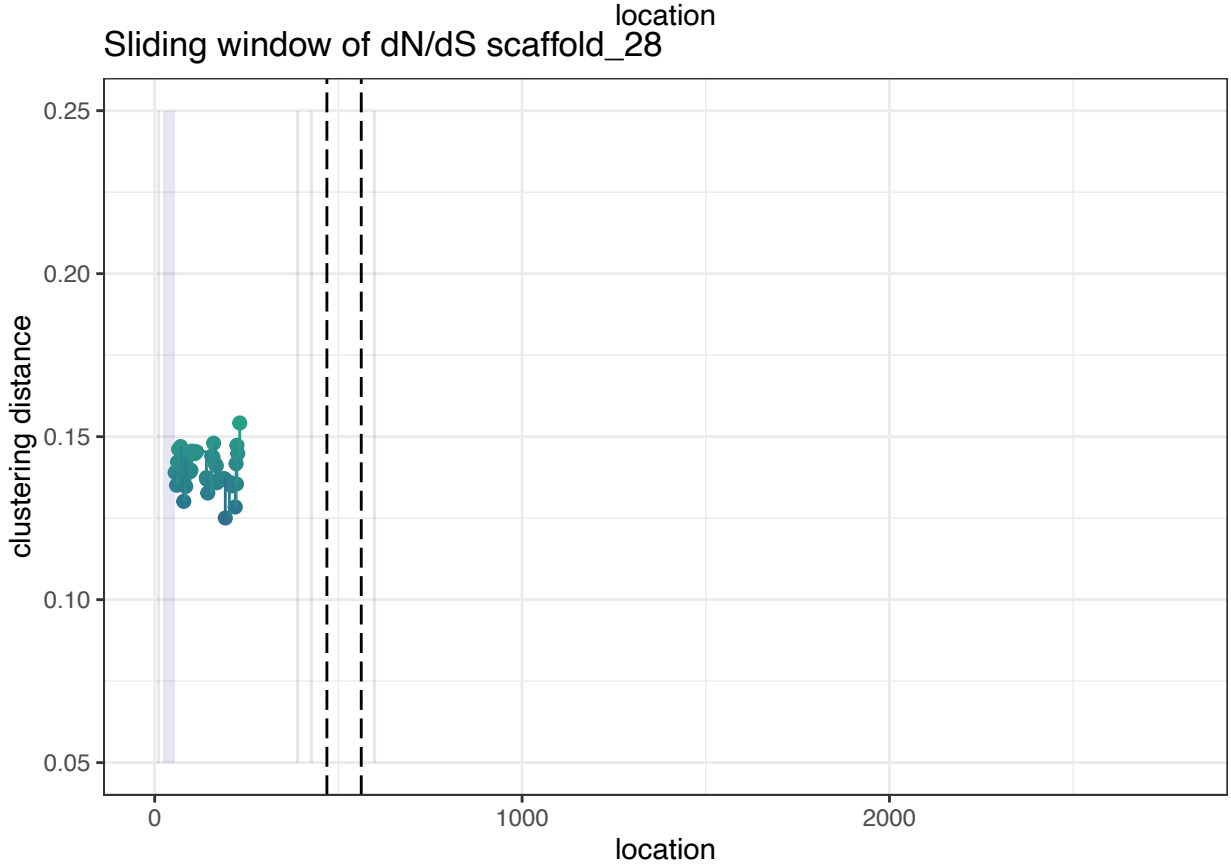

Sliding window of dN/dS scaffold_22
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

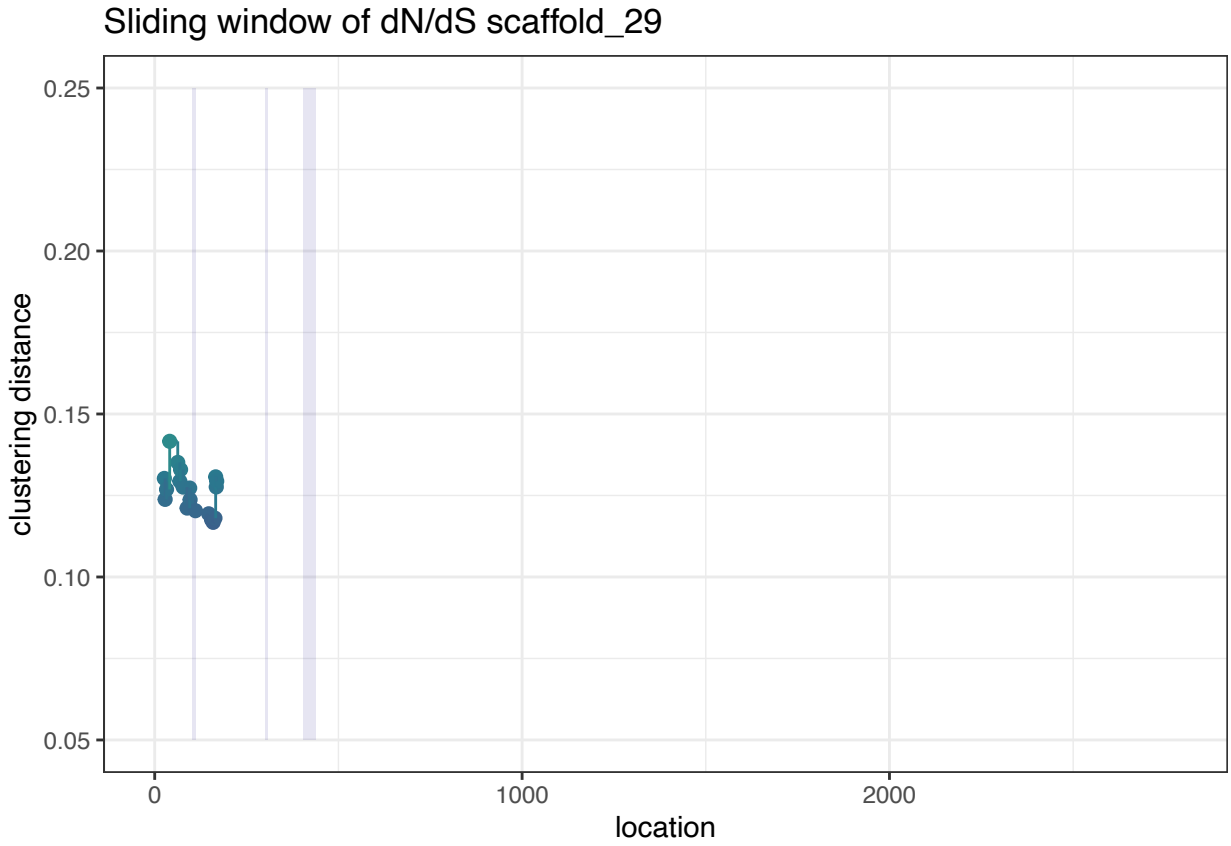

Sliding window of dN/dS scaffold_23
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location


Sliding window of dN/dS scaffold_24
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

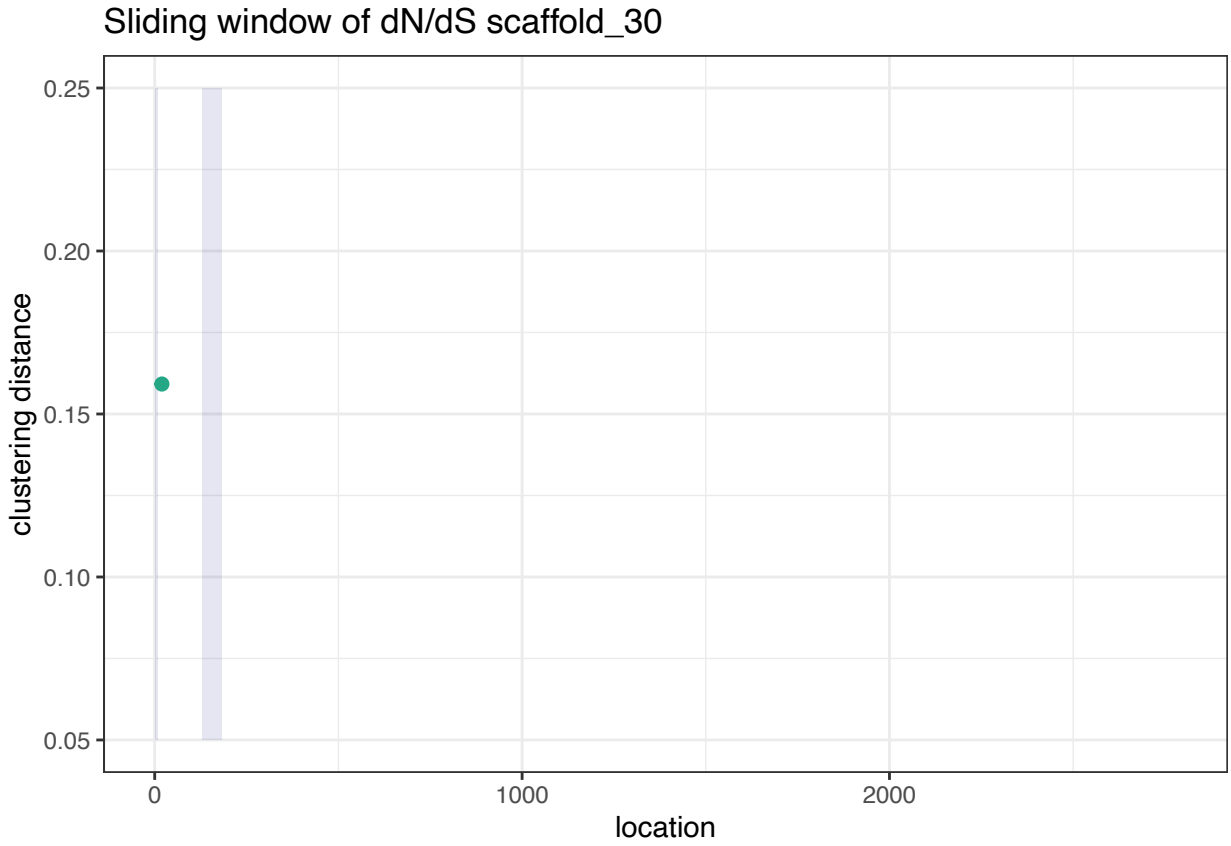

Sliding window of dN/dS scaffold_26
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location


Sliding window of dN/dS scaffold_27
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

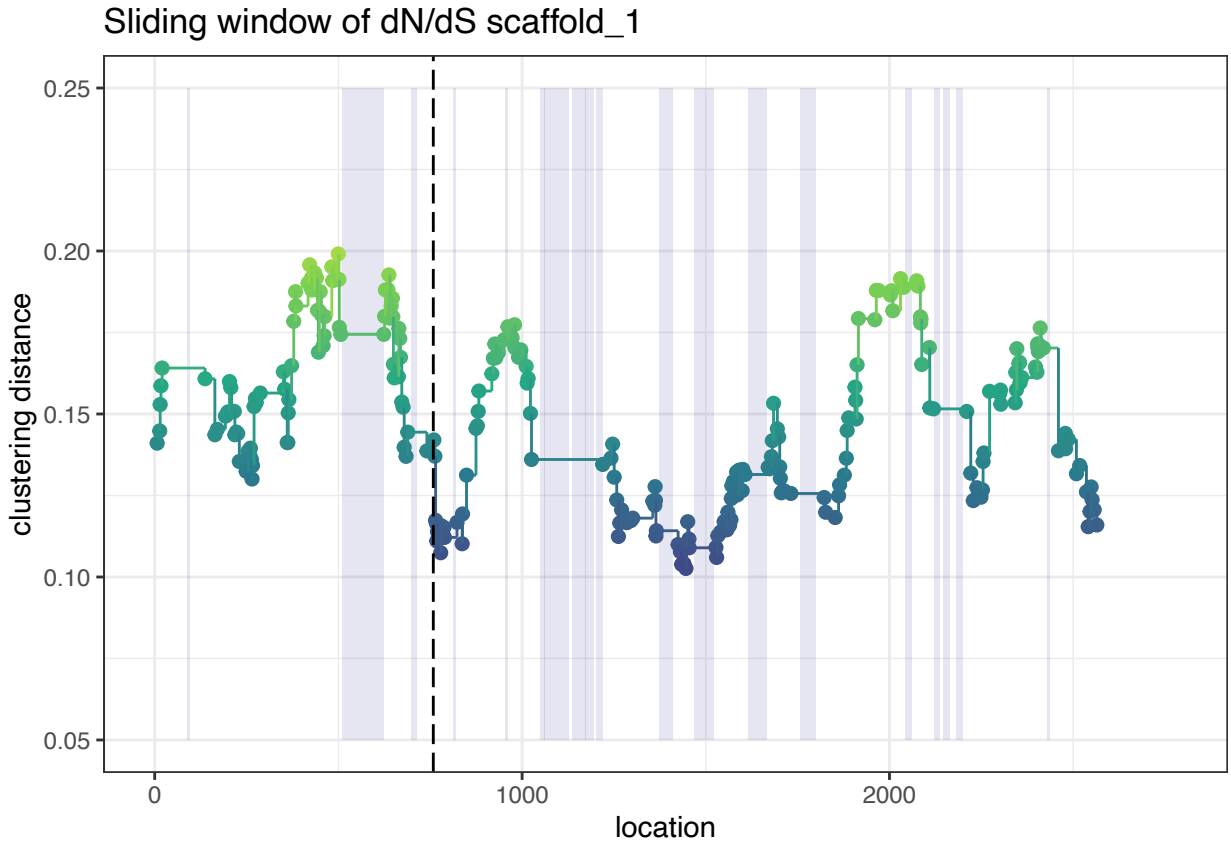

Sliding window of dN/dS scaffold_28
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location



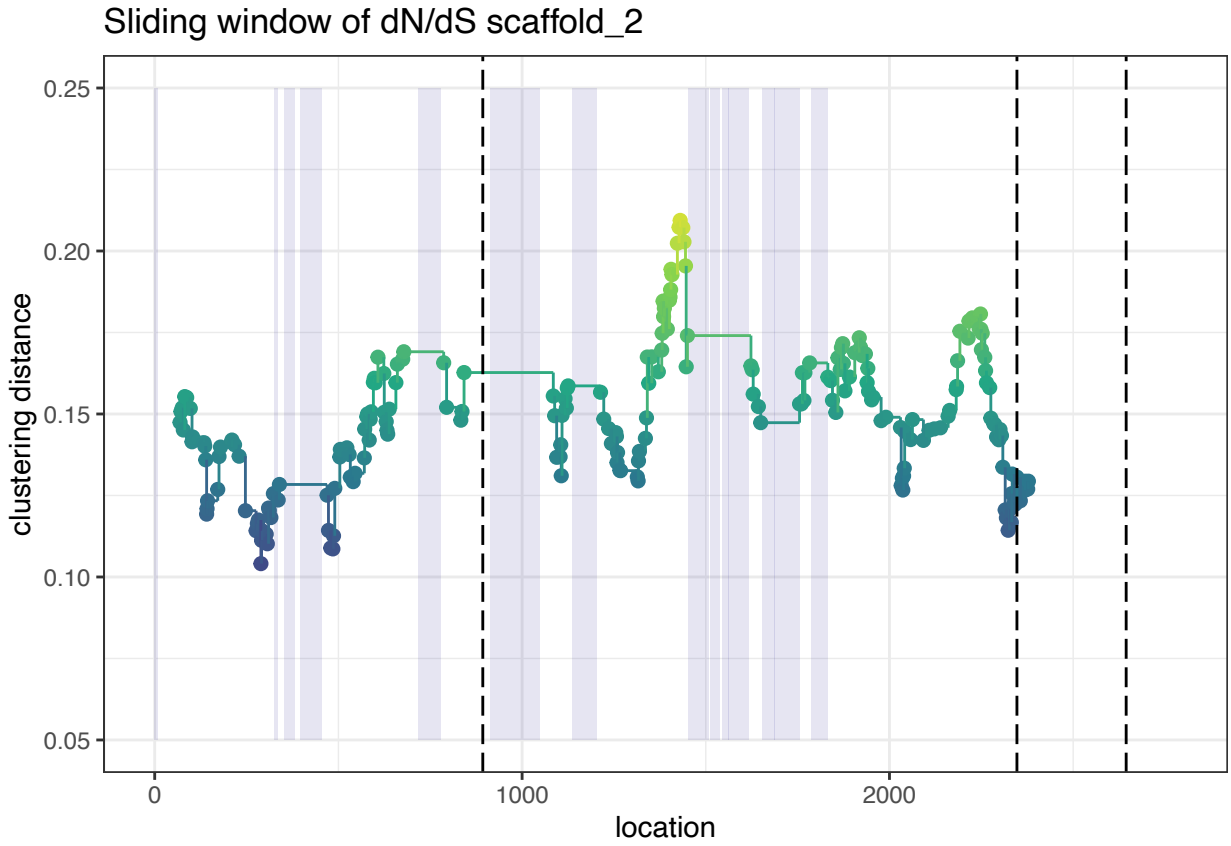

```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```


Sliding window of dN/dS scaffold_30
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

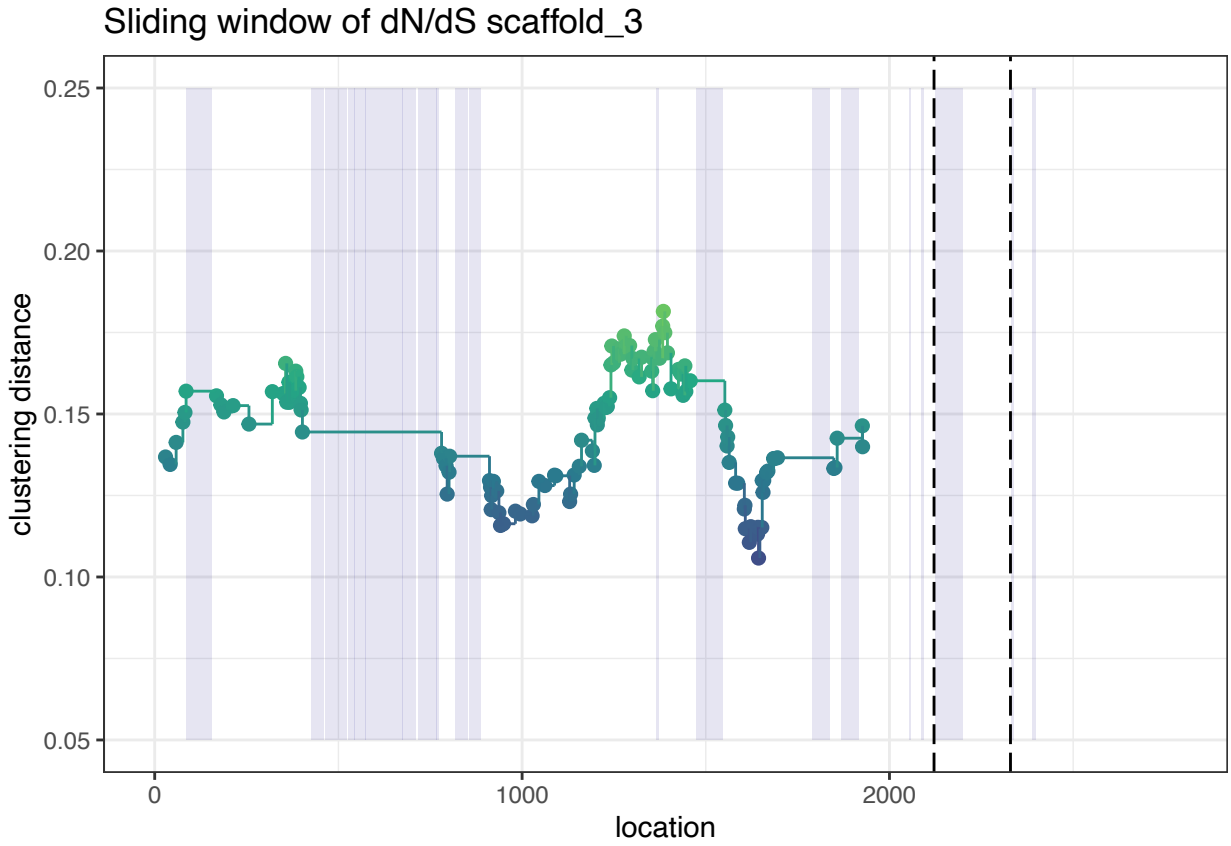
```
## $scaffold_1
```


Sliding window of dN/dS scaffold_1
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

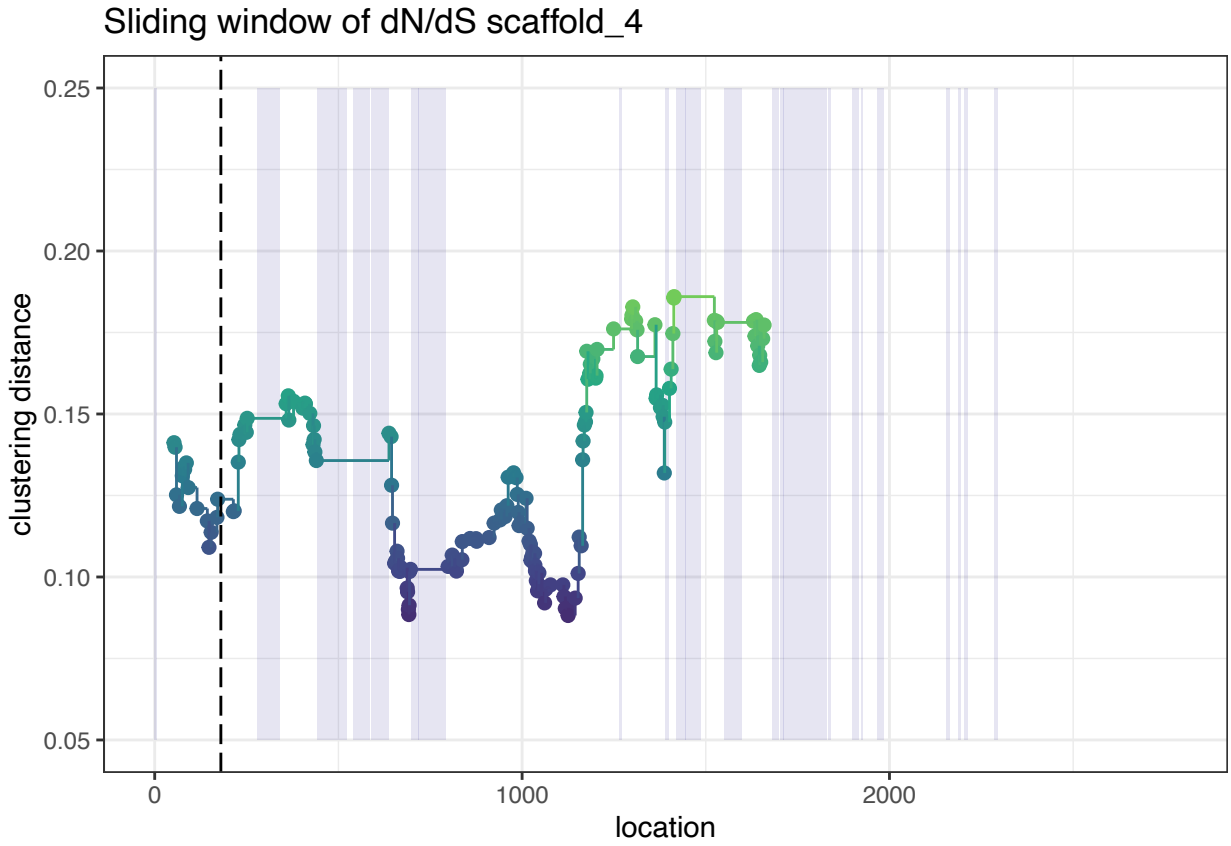
```
##
## $scaffold_2
```


Sliding window of dN/dS scaffold_2
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

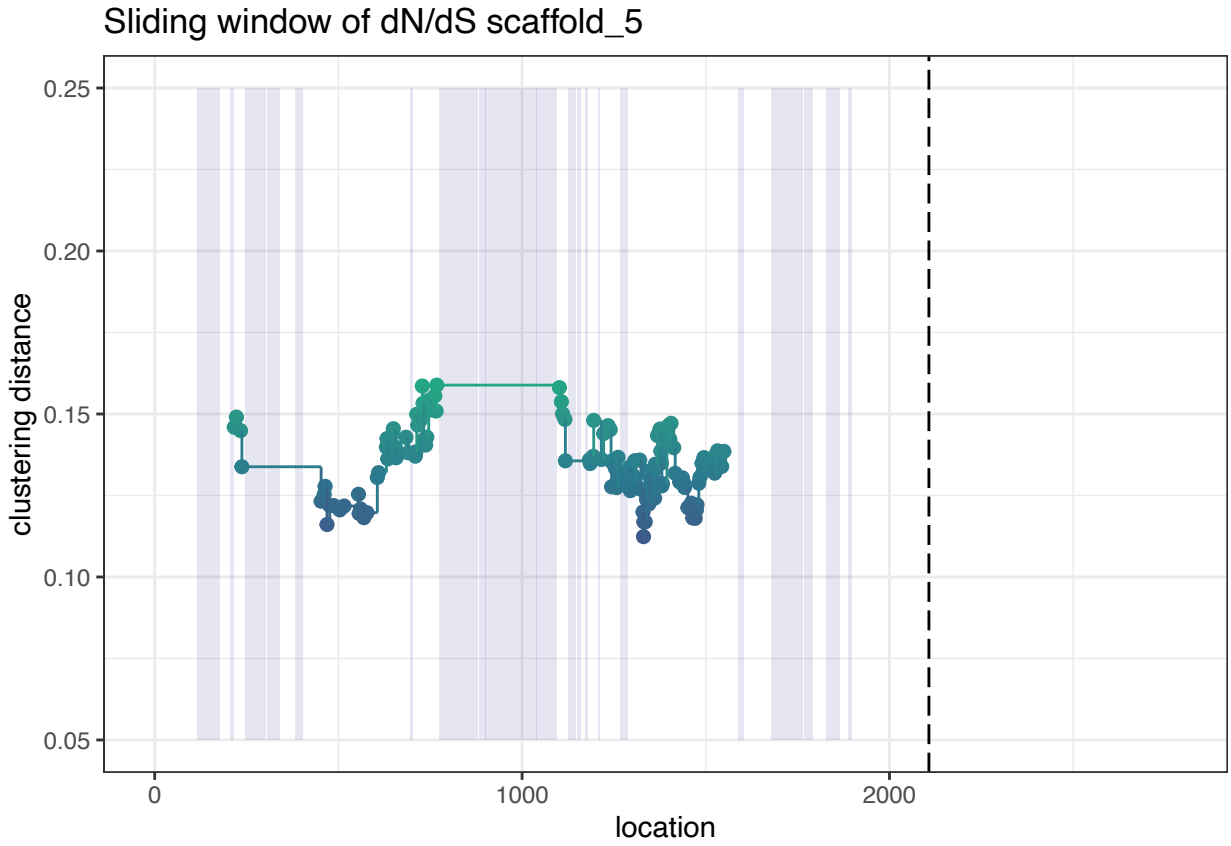
```
##
## $scaffold_3
```


Sliding window of dN/dS scaffold_3
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

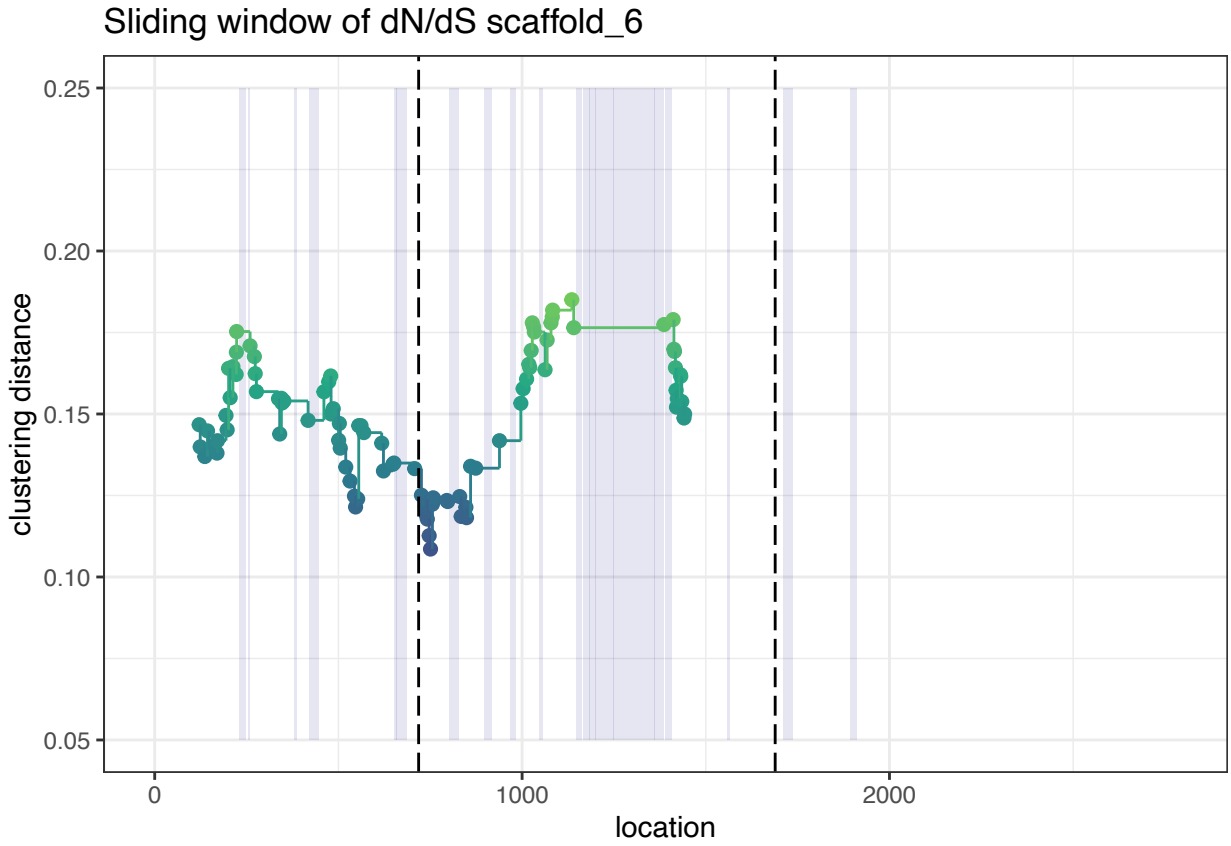
```
##
## $scaffold_4
```


Sliding window of dN/dS scaffold_4
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

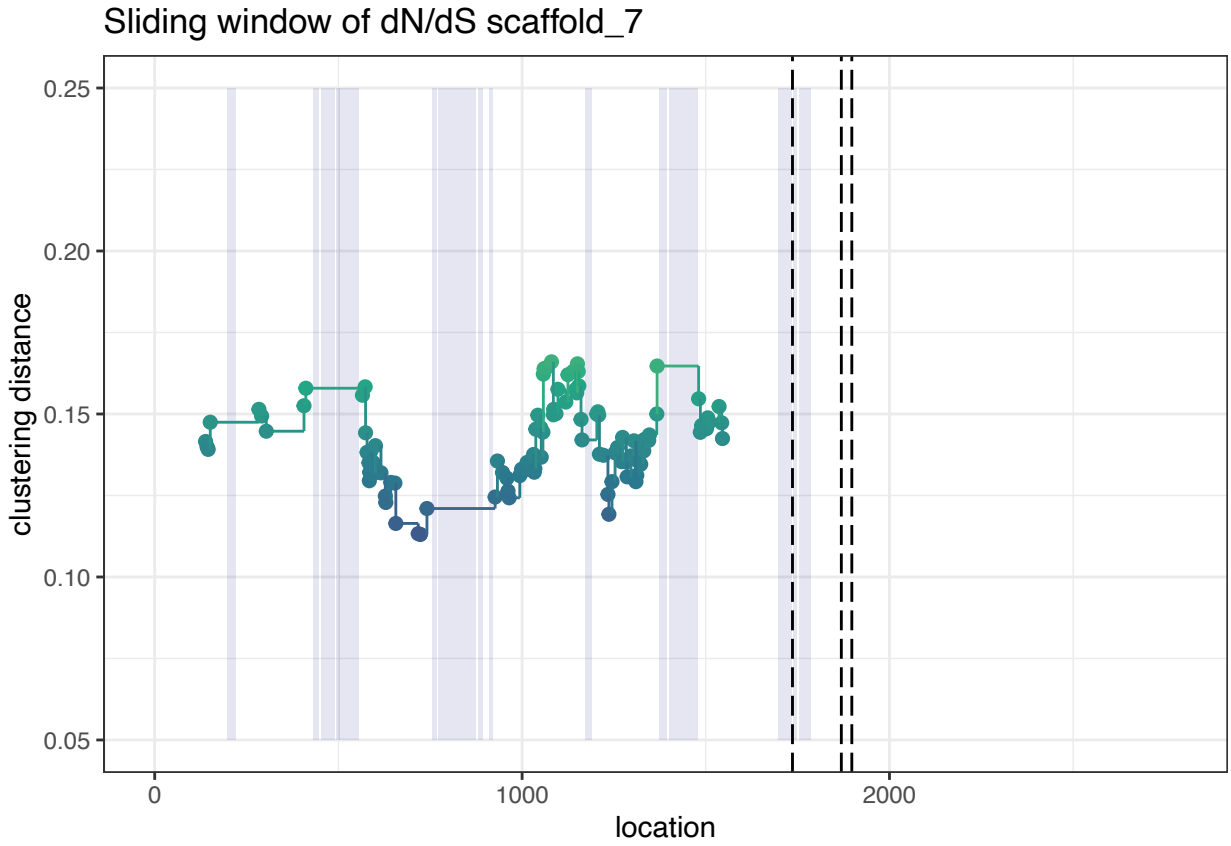
```
##
## $scaffold_5
```


Sliding window of dN/dS scaffold_5
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

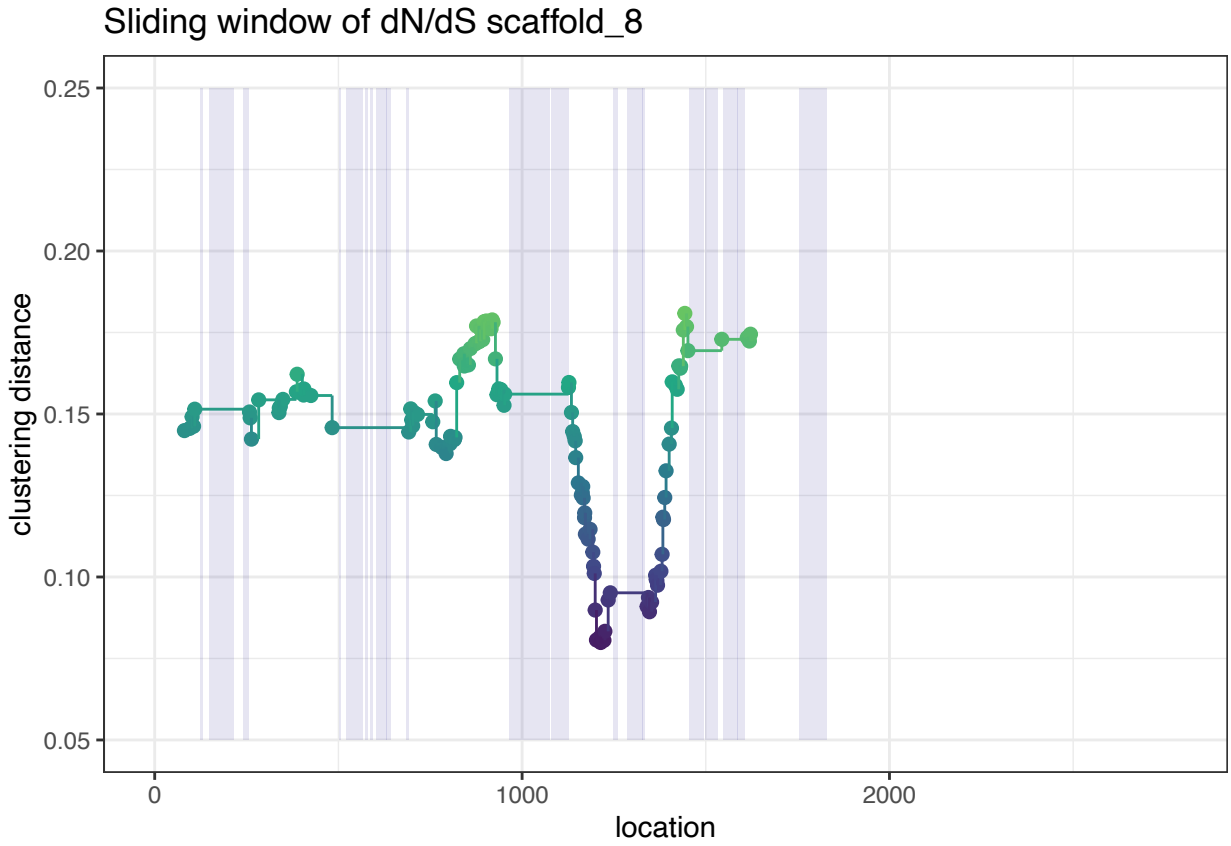
```
##
## $scaffold_6
```


Sliding window of dN/dS scaffold_6
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

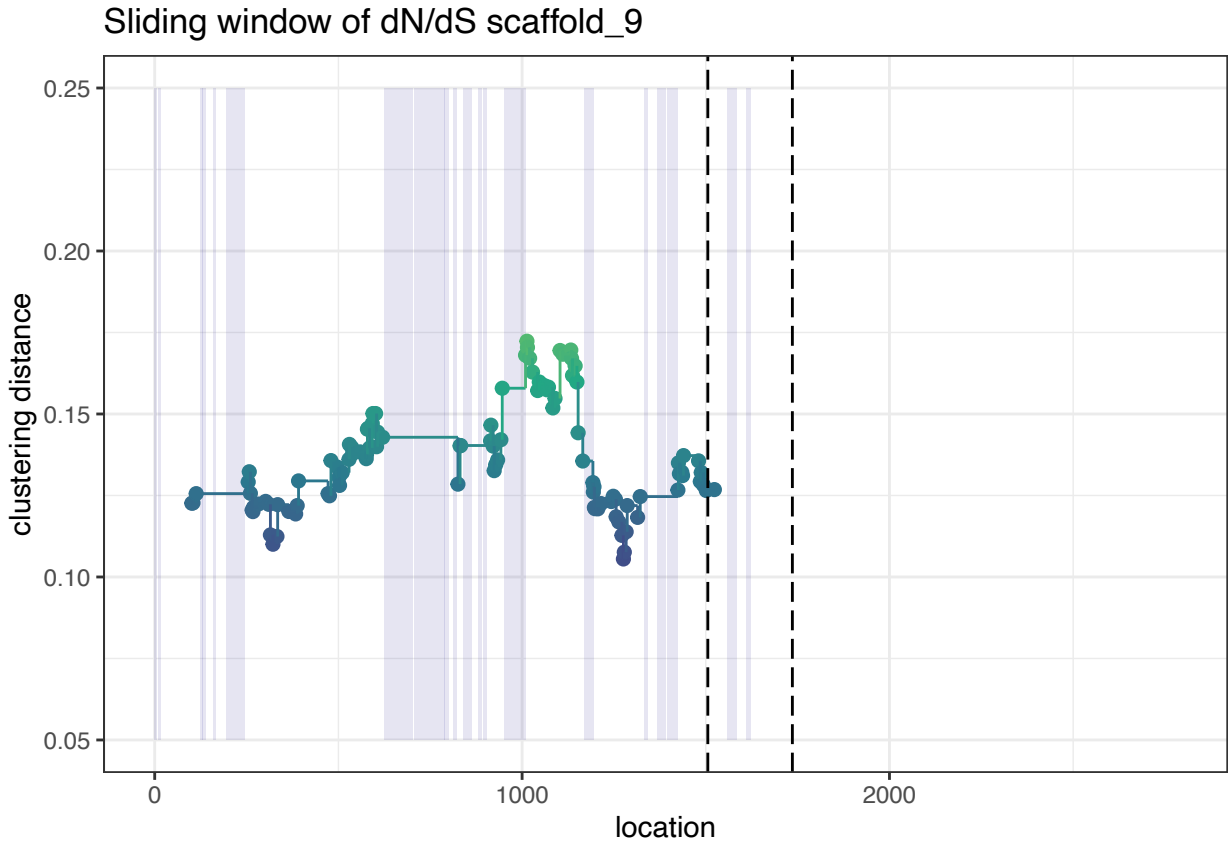
```
##
## $scaffold_7
```


Sliding window of dN/dS scaffold_7
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

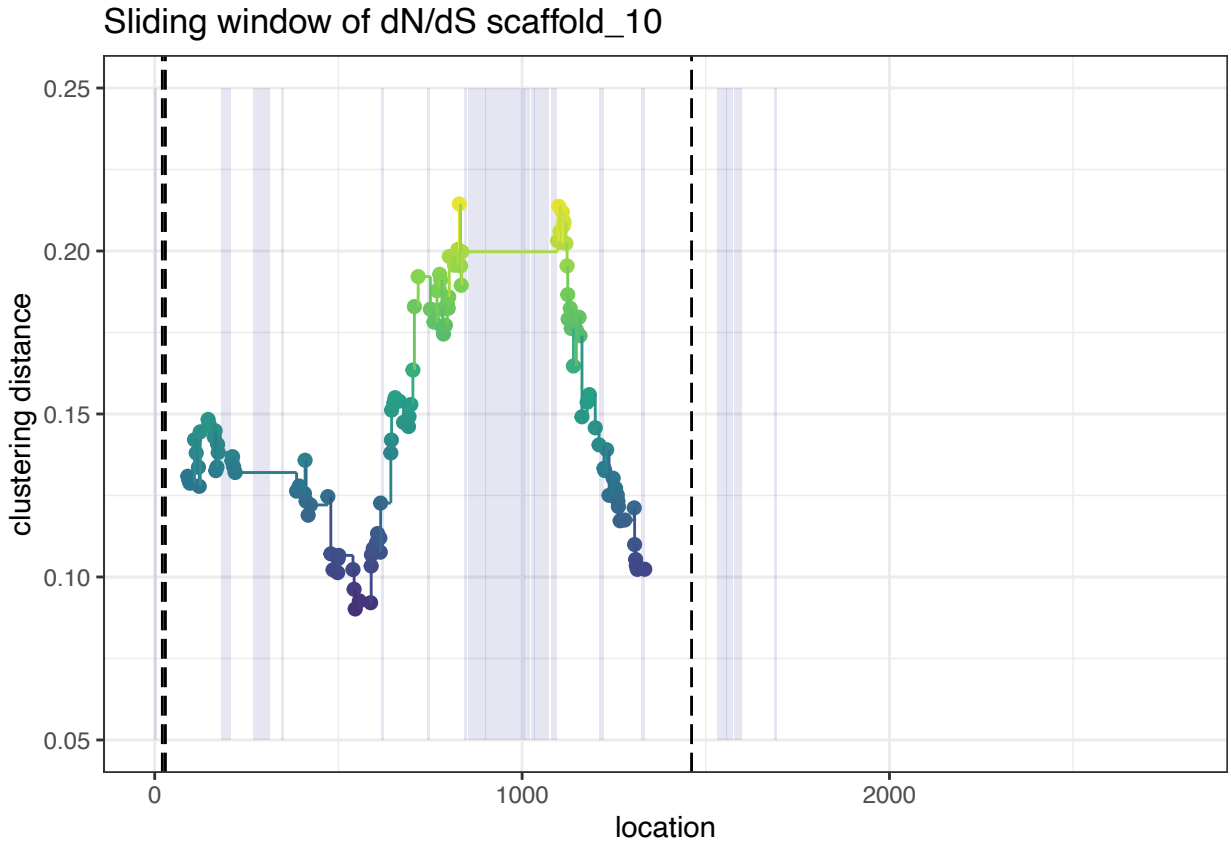
```
##
## $scaffold_8
```


Sliding window of dN/dS scaffold_8
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

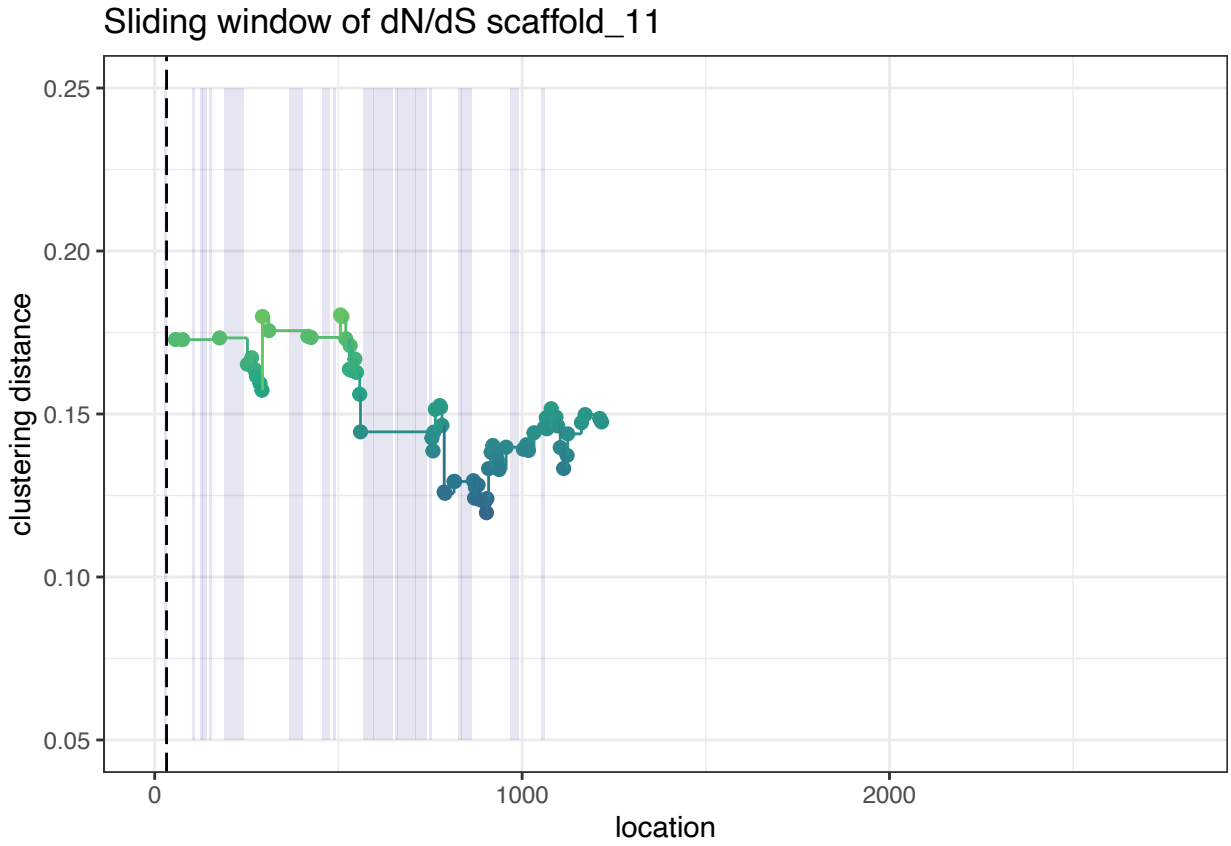
```
##
## $scaffold_9
```


Sliding window of dN/dS scaffold_9
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

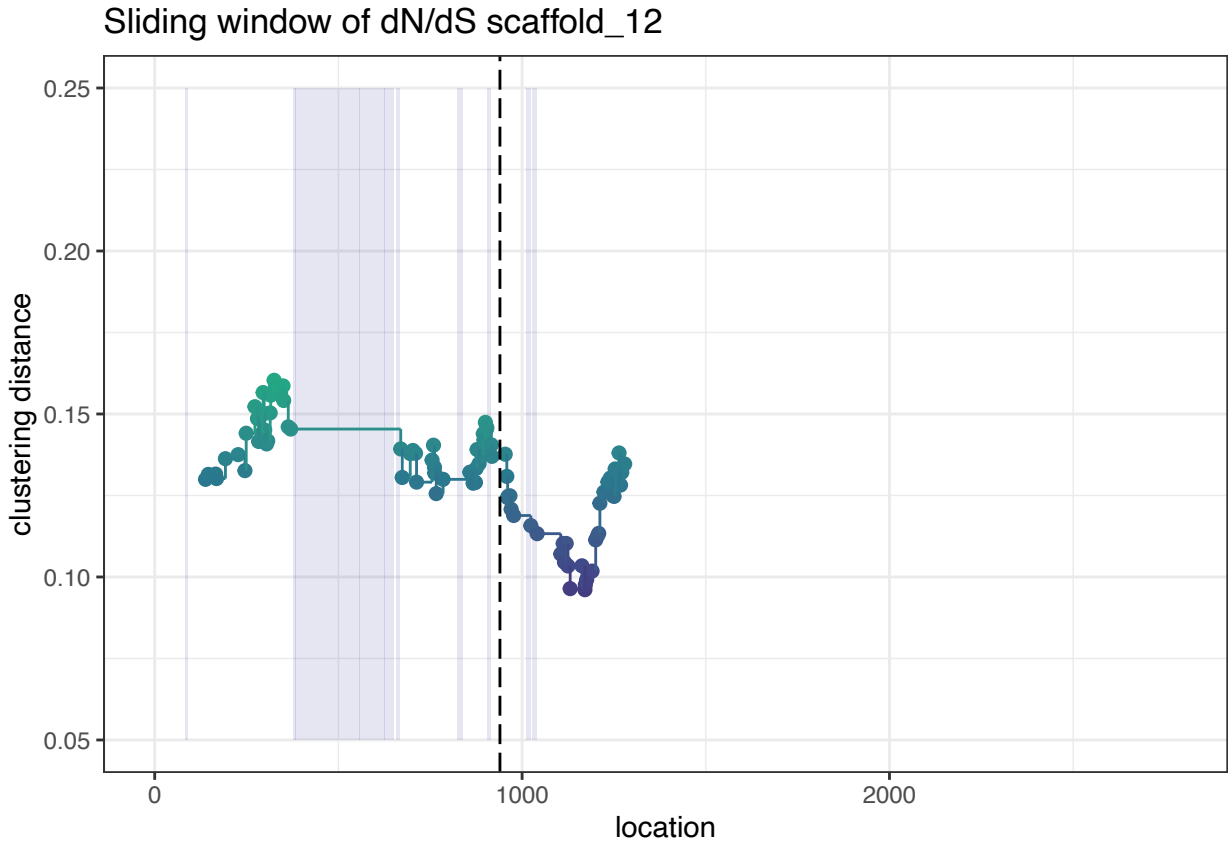
```
##
## $scaffold_10
```


Sliding window of dN/dS scaffold_10
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

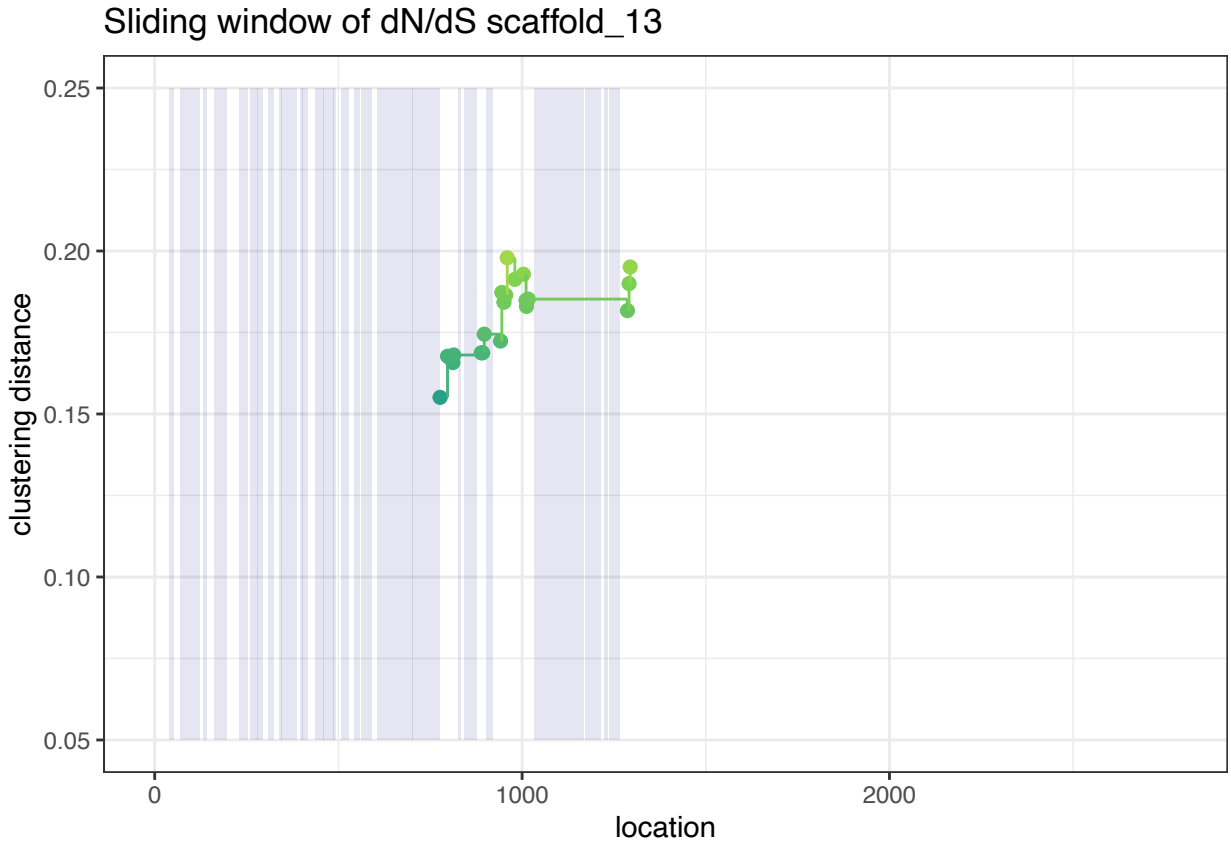
```
##
## $scaffold_11
```


Sliding window of dN/dS scaffold_11
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

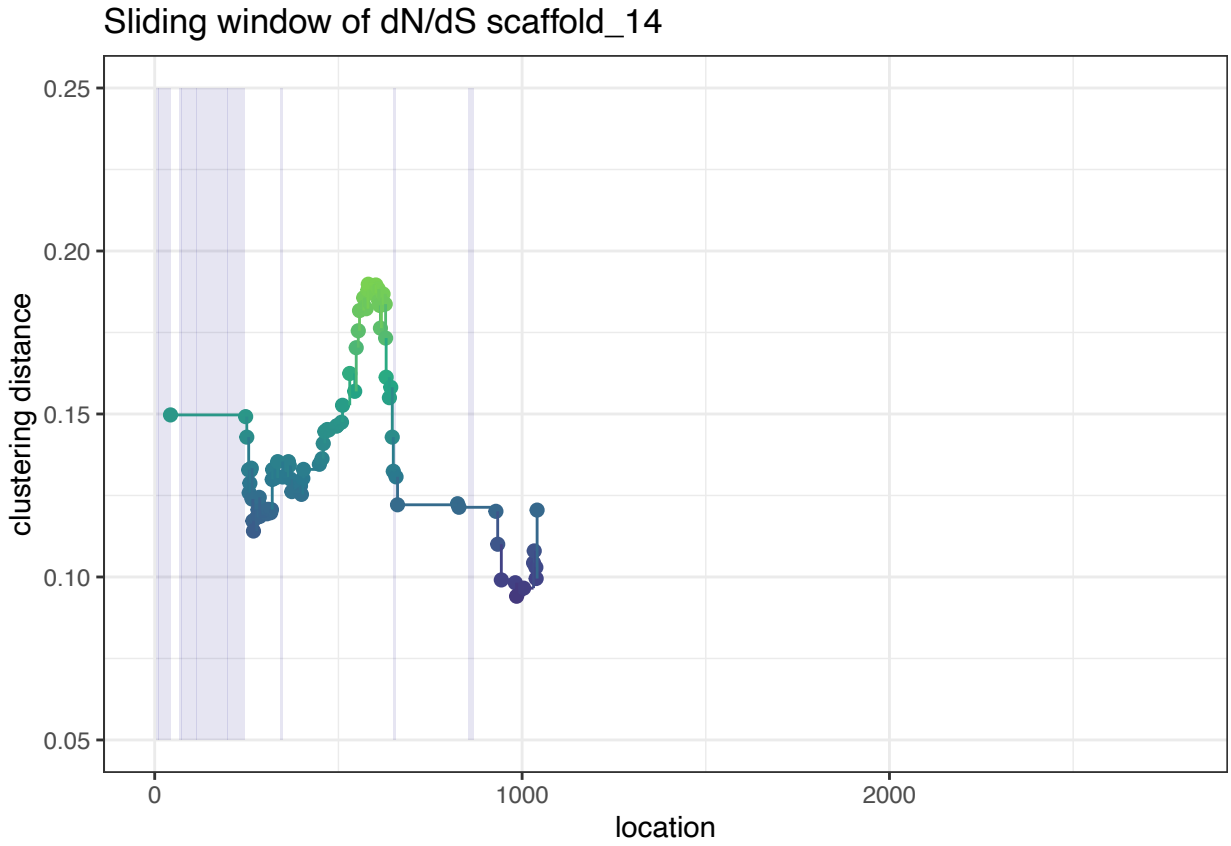
```
##
## $scaffold_12
```


Sliding window of dN/dS scaffold_12
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

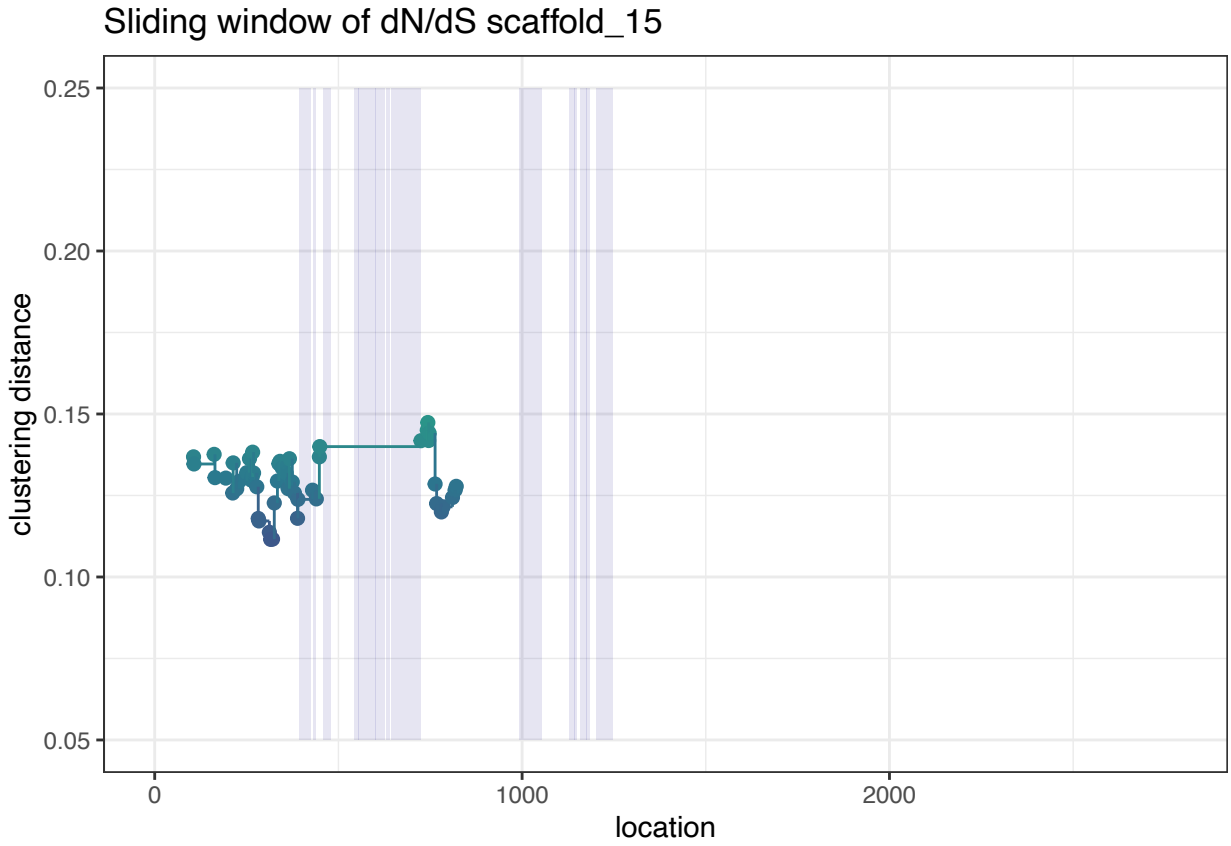
```
##
## $scaffold_13
```


Sliding window of dN/dS scaffold_13
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

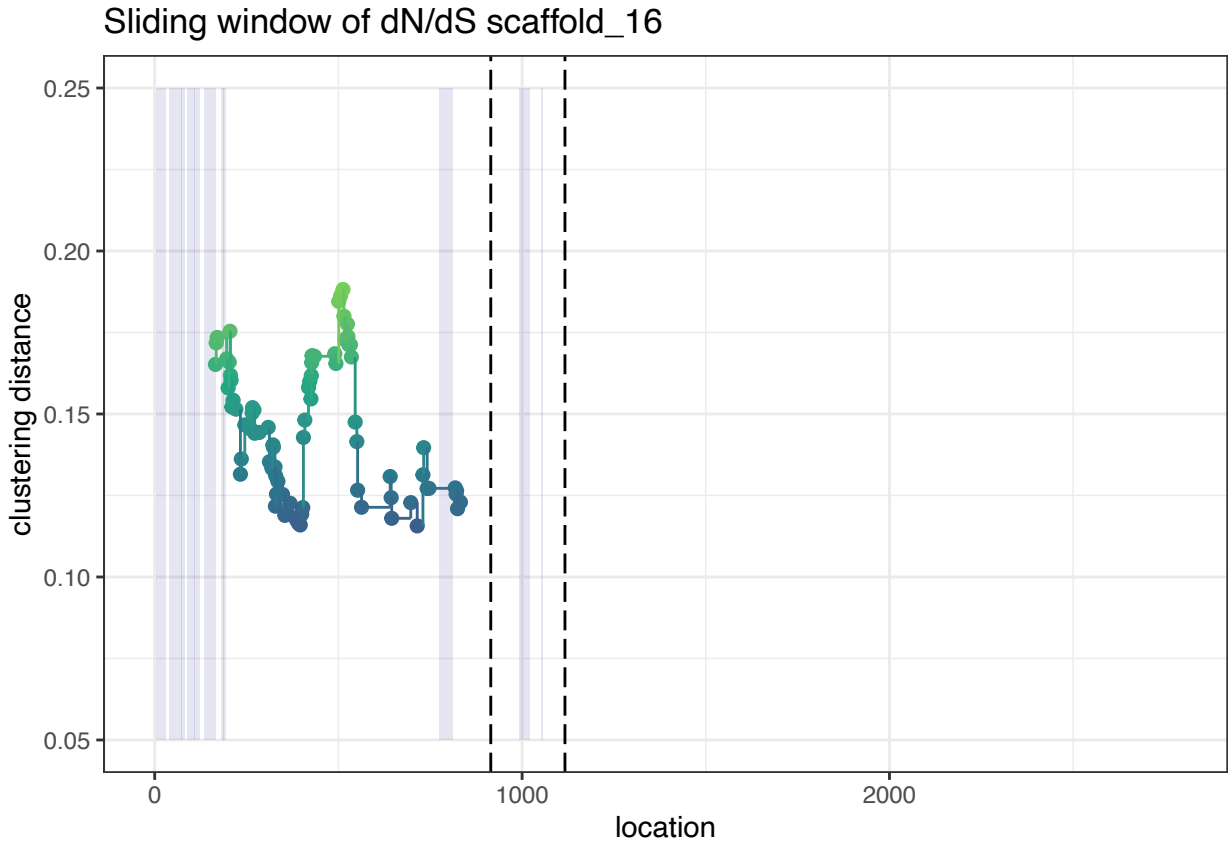
```
##
## $scaffold_14
```


Sliding window of dN/dS scaffold_14
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

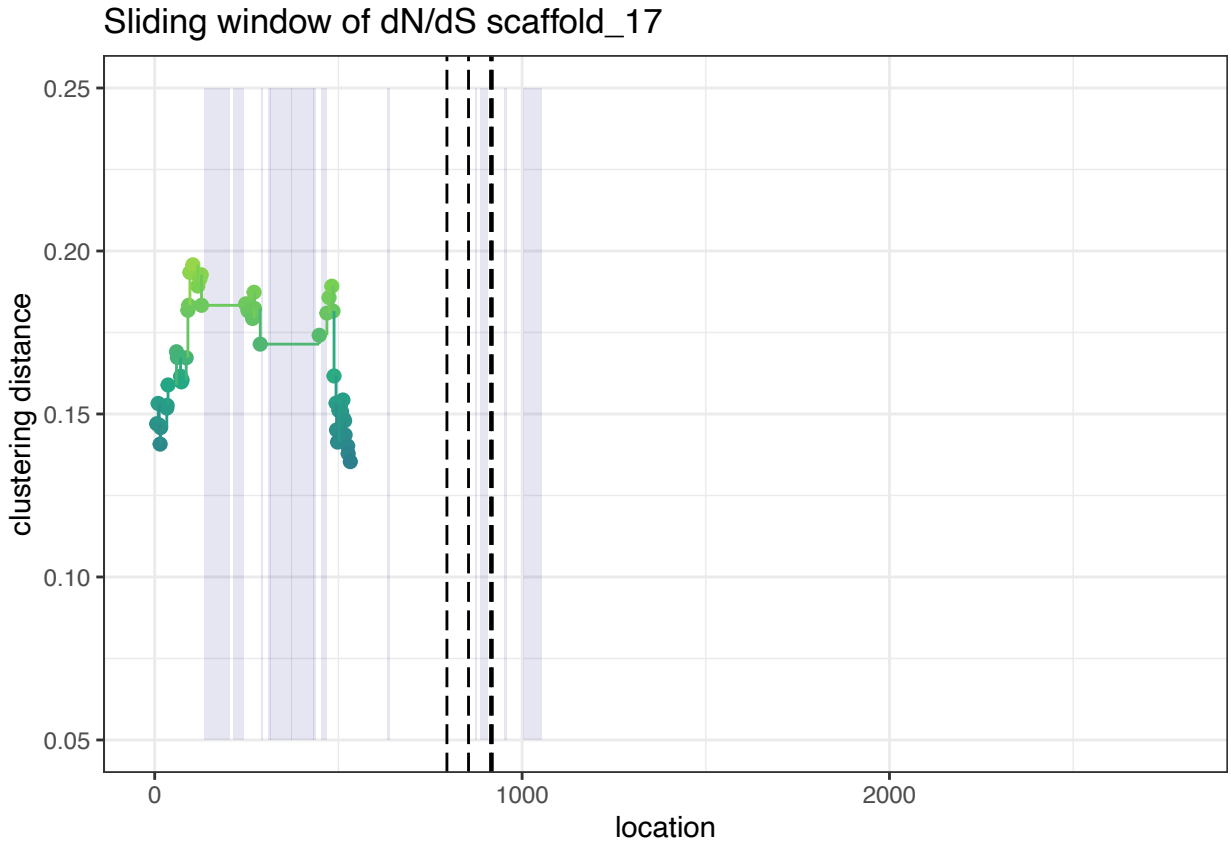
```
##
## $scaffold_15
```


Sliding window of dN/dS scaffold_15
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

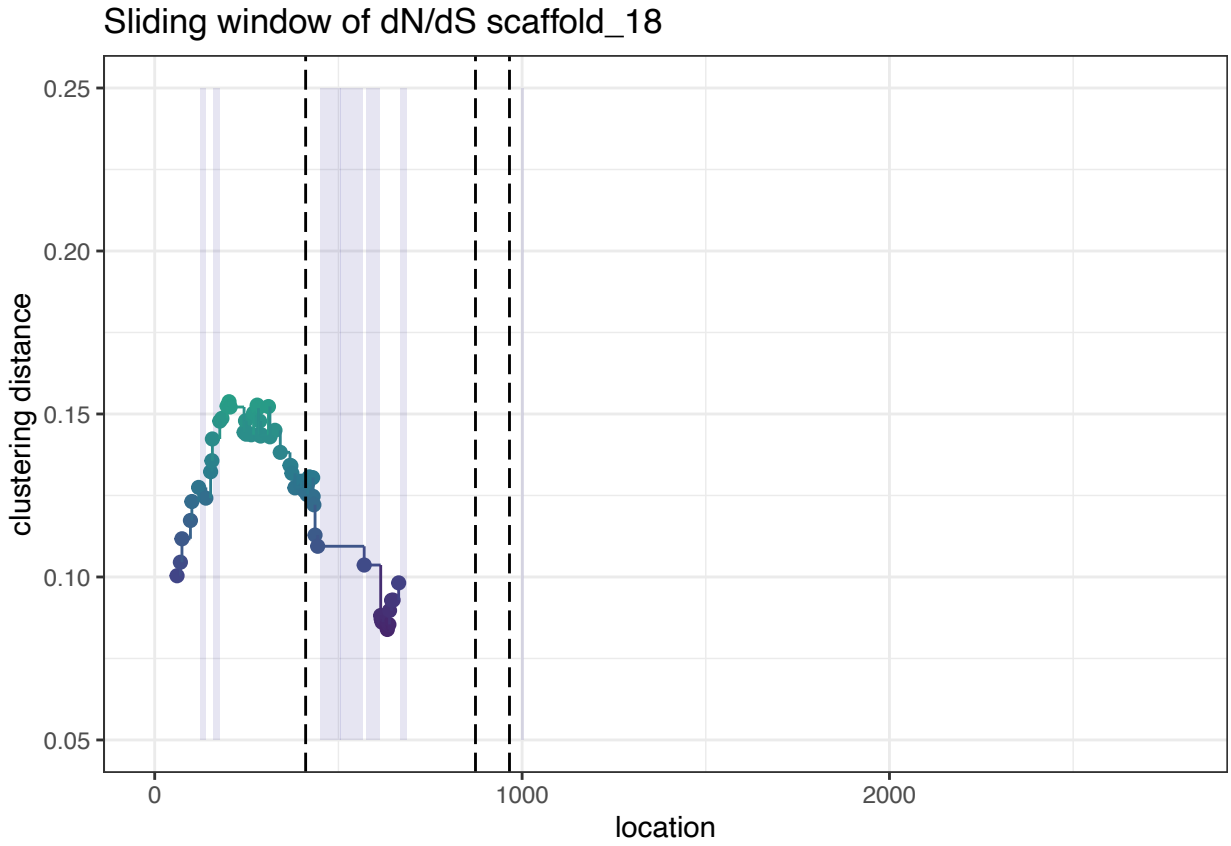
```
##
## $scaffold_16
```


Sliding window of dN/dS scaffold_16
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

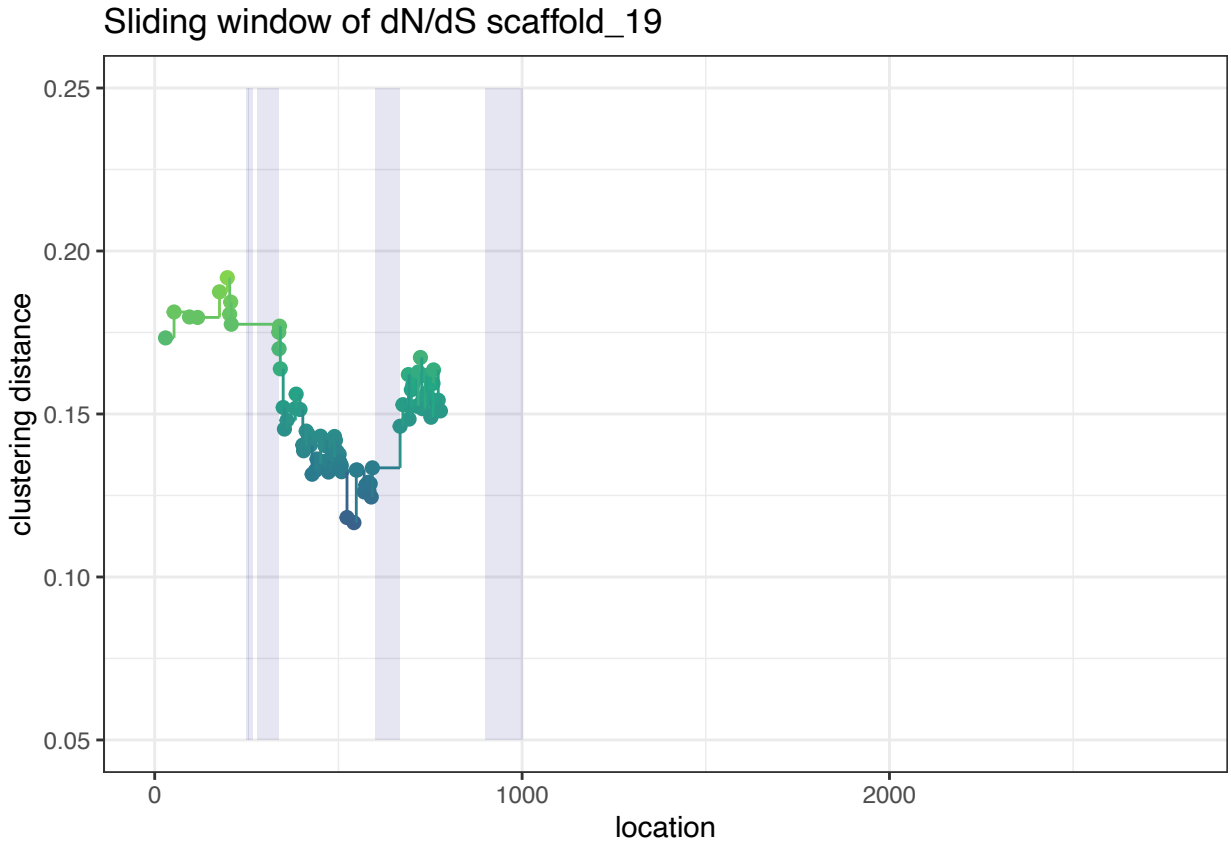
```
##
## $scaffold_17
```


Sliding window of dN/dS scaffold_17
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

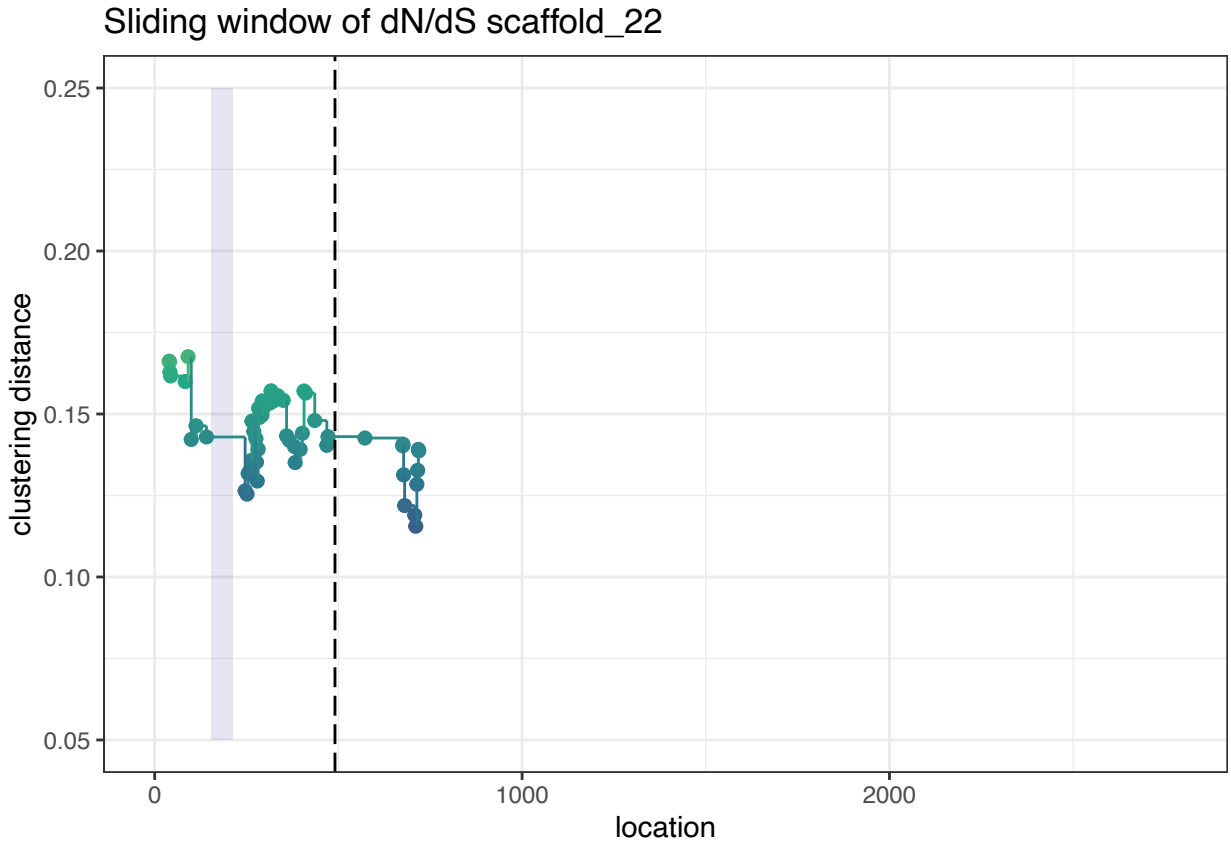
```
##
## $scaffold_18
```


Sliding window of dN/dS scaffold_18
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

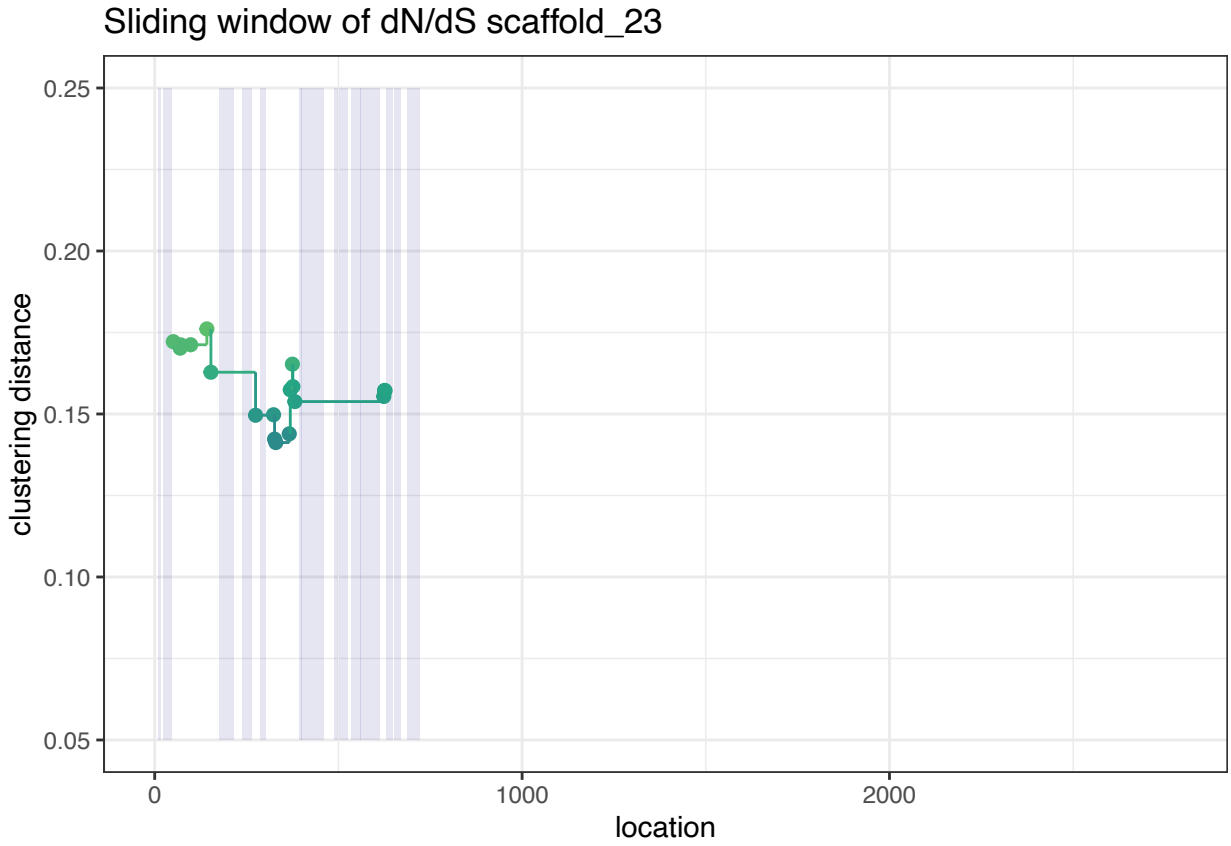
```
##
## $scaffold_19
```


Sliding window of dN/dS scaffold_19
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

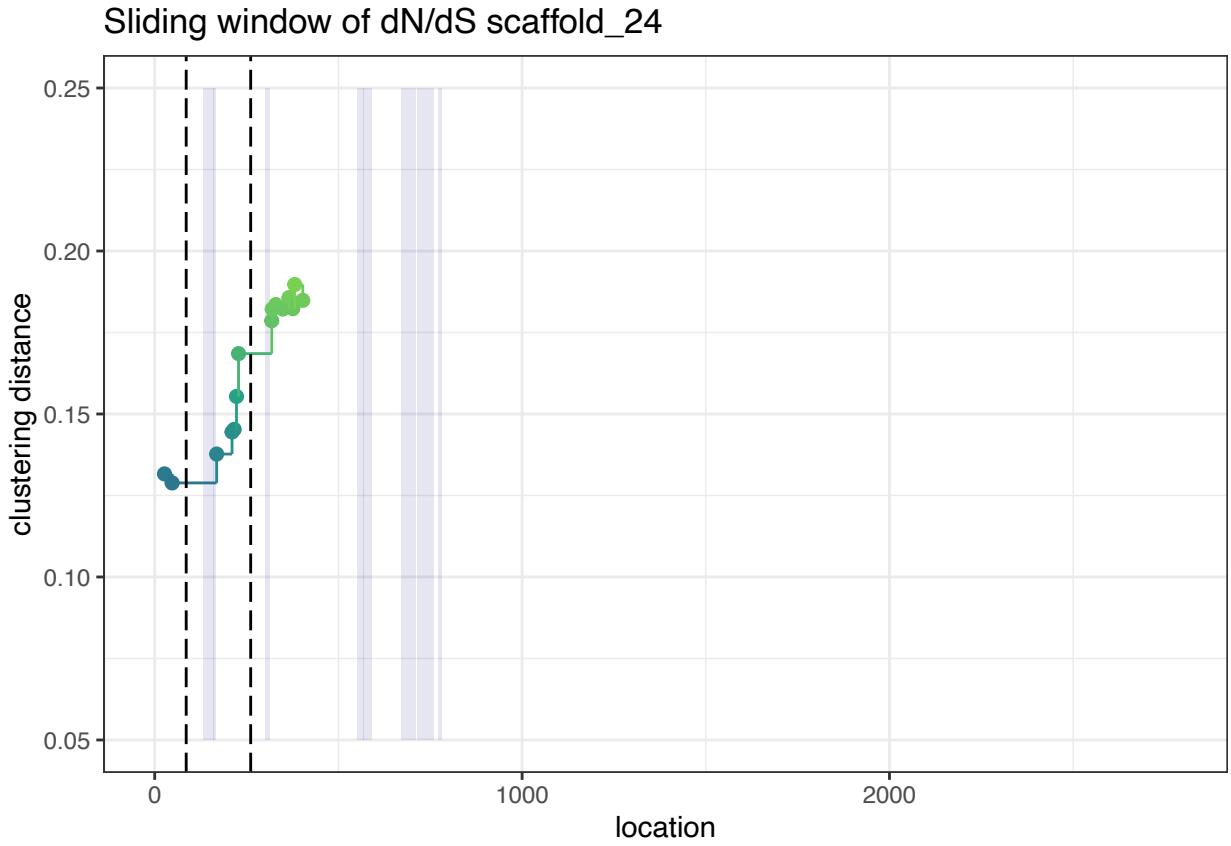
```
##
## $scaffold_22
```


Sliding window of dN/dS scaffold_22
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

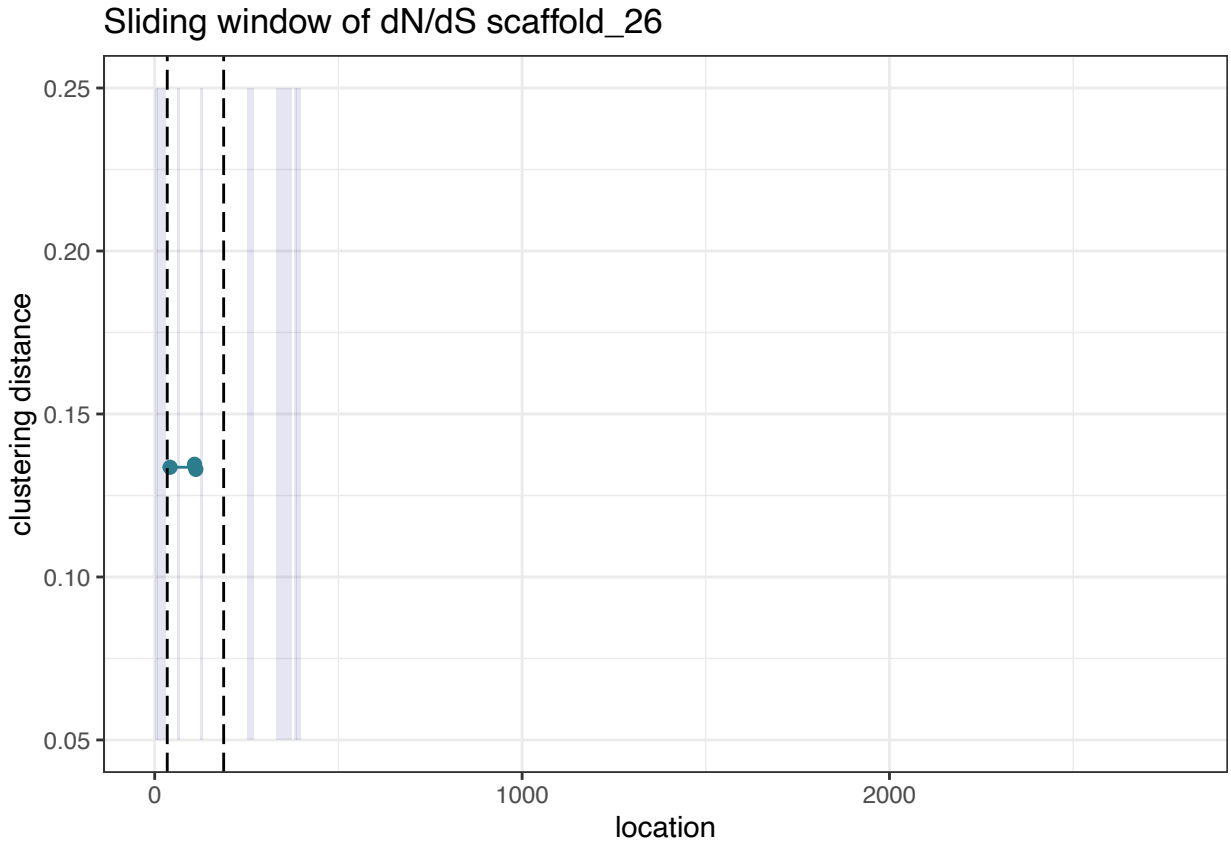
```
##
## $scaffold_23
```


Sliding window of dN/dS scaffold_23
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

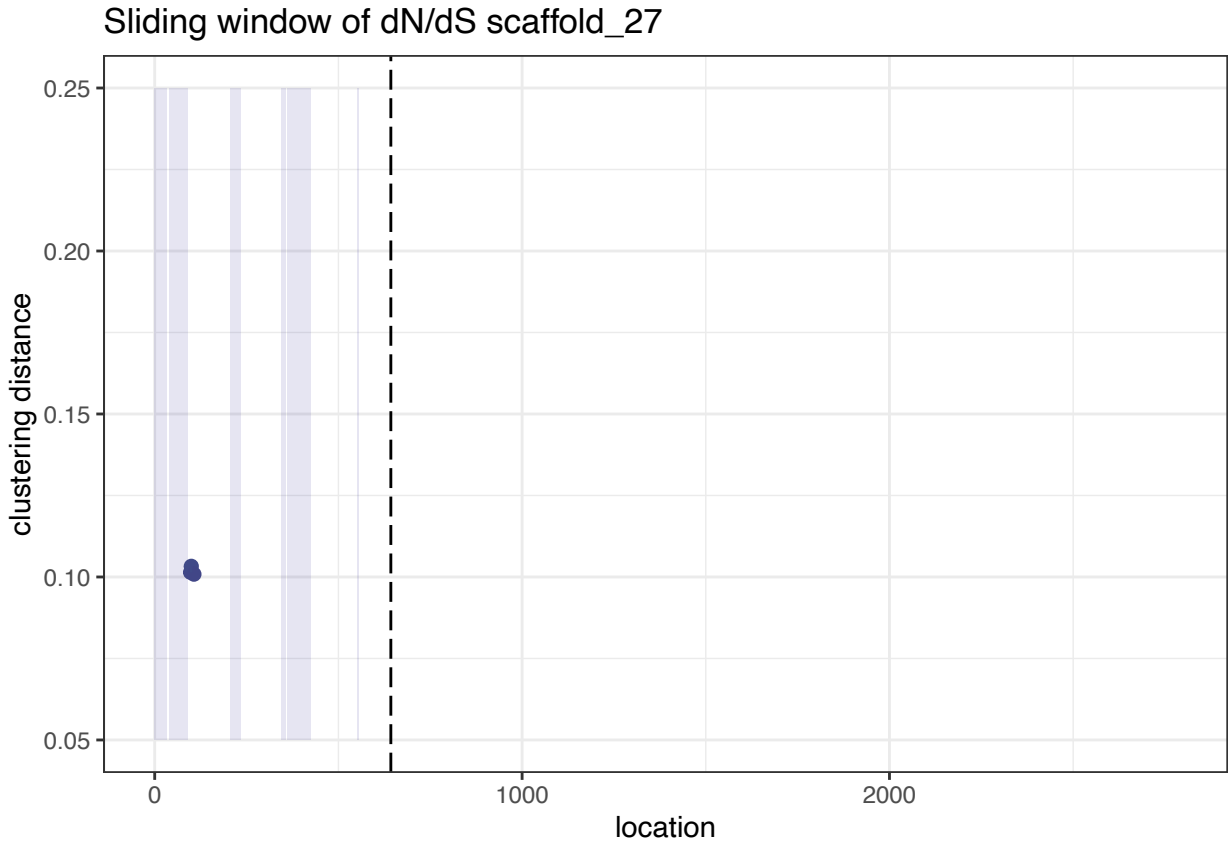
```
##
## $scaffold_24
```


Sliding window of dN/dS scaffold_24
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

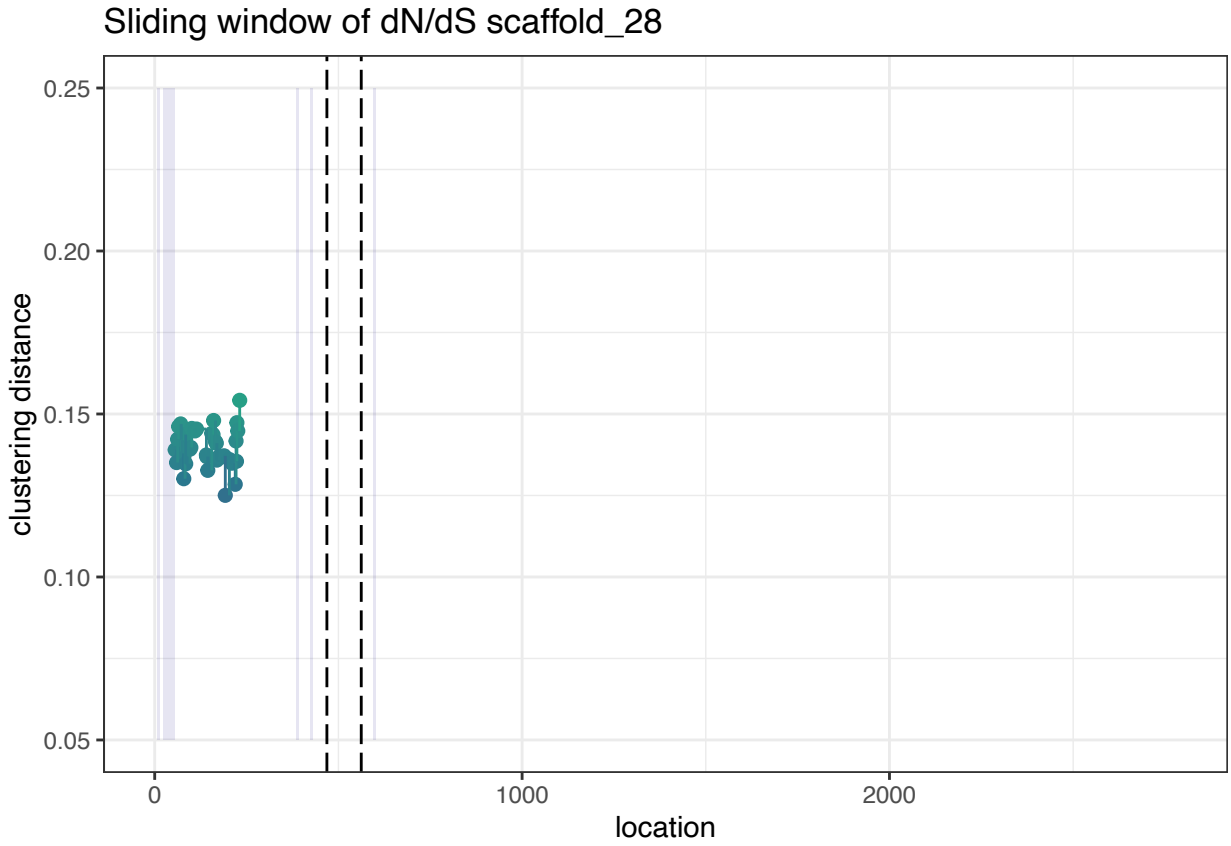
```
##
## $scaffold_26
```


Sliding window of dN/dS scaffold_26
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

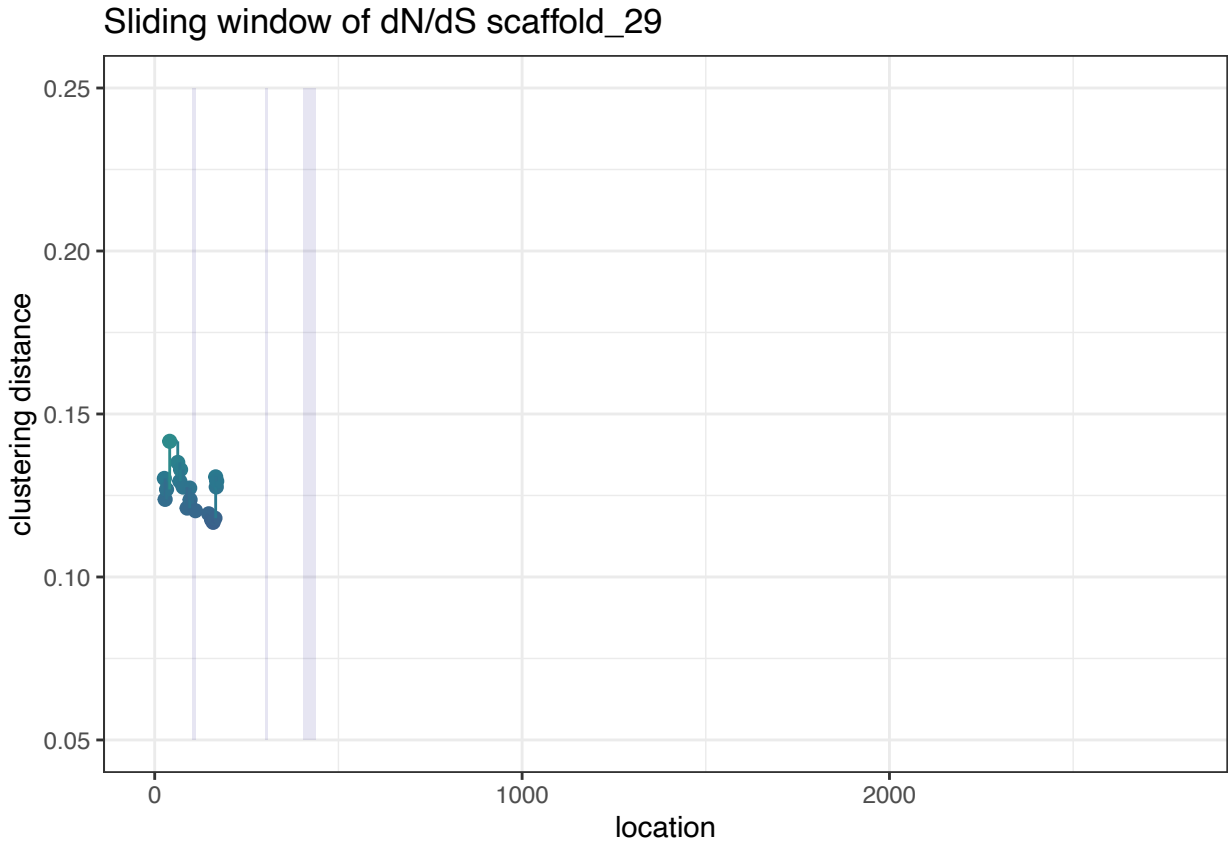
```
##
## $scaffold_27
```


Sliding window of dN/dS scaffold_27
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

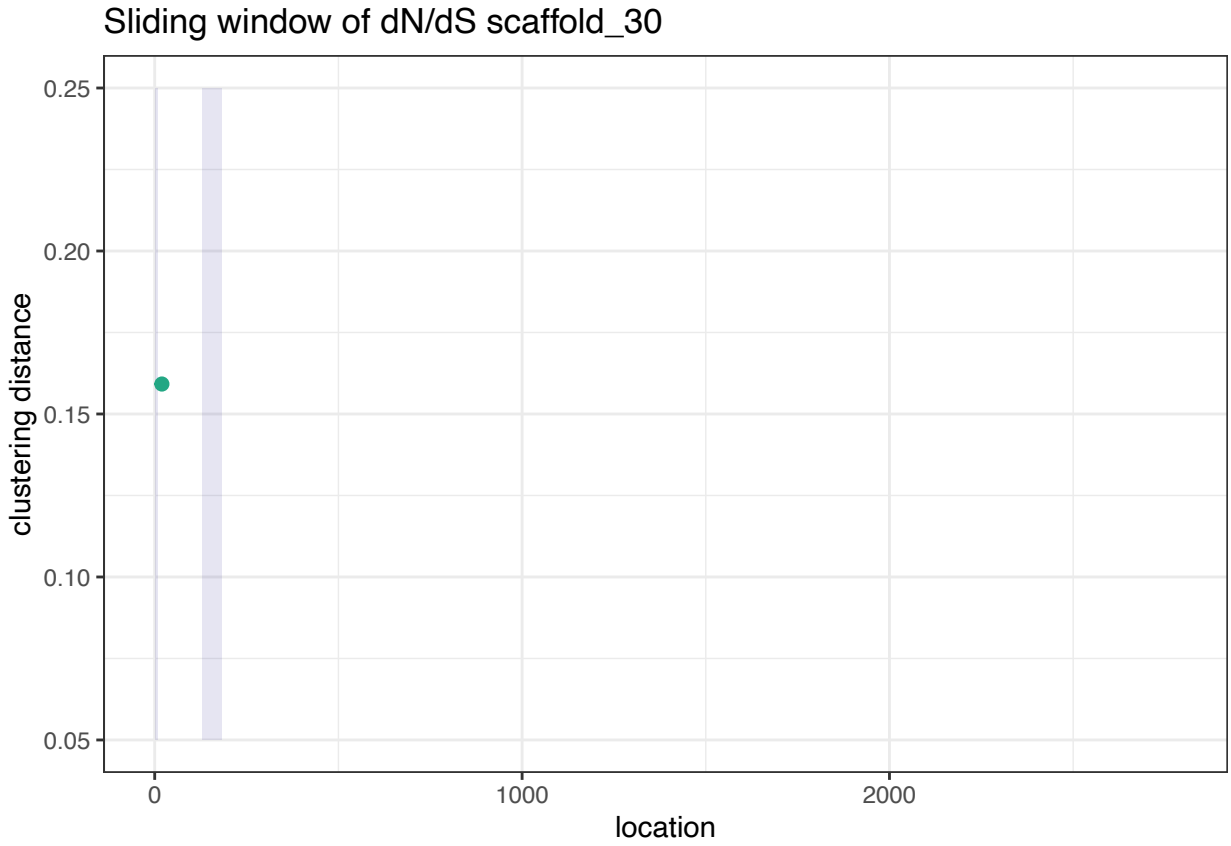
```
##
## $scaffold_28
```


Sliding window of dN/dS scaffold_28
clustering distance
0.25
0.20
0.15
0.10
0.05
0
1000
2000
location

```
##
## $scaffold_29
```




```
##
## $scaffold_30

## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

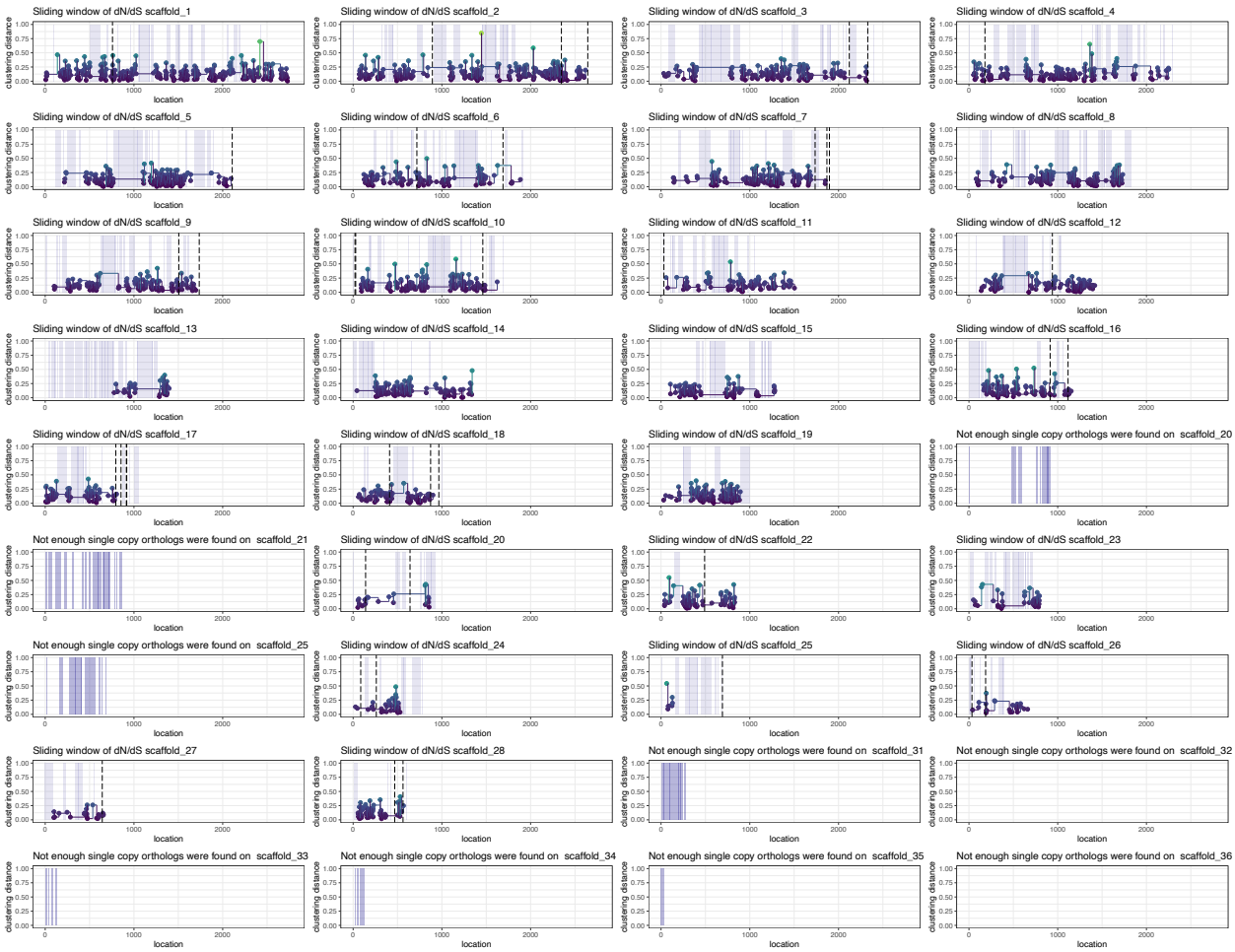


5.4.2 Plot of dN/dS values per ortholog, not on a sliding window.

```
logfile<-NULL
swindow<-NULL
foo.genes<-genes[rownames(ogs),]
plots<-list()
limite<-length(pe_genome[[1]])/1000
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2)
  {
    culo<-lrar[lrar$Name==SCAF,]
    foo.dnds<-ogs$dn_ds[foo.genes$scaffold==SCAF]
    foo.loc<-as.numeric(as.character(foo.genes$start[foo.genes$scaffold==SCAF]))/1000
        p<-ggplot(
          data.frame(dnds=foo.dnds,x=foo.loc),
          aes(x=x, y=dnds, color=dnds)) +
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0,ymax=1,alpha =.1,fill = spectru
          geom_point(size=2) +
          geom_step() +
          scale_color_continuous(type = "viridis", limits = range(0,1)) +
          xlim(c(0,limite)) + ylim(c(0,1)) +
          theme_bw() +
          labs (title=paste("Sliding window of dN/dS",SCAF),x="location",y="clustering distance")
    if (length(unlist(het[het[,1]==SCAF,]))!=0)
    {
          plots[[SCAF]]<- p + geom_vline(xintercept = as.numeric(as.character(het[het[,1]==SCAF,2]))/10
    }else{
```

```
    plots[[SCAF]]<-p + theme(legend.position = "none")
      }
    swindow<-NULL
}
```

```
pempty<-list()
for (SCAF in c("scaffold_20","scaffold_21","scaffold_25","scaffold_31","scaffold_32","scaffold_33","sca
  {
    culo<-lrar[lrar$Name==SCAF,]
    pempty[[SCAF]]<-ggplot(
          data.frame(mean_dist=c(5,5,5),x=c(0,1000,10000)),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(0,1))+theme_bw()+
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0,ymax=1,alpha = .3,fill = specti
          labs (title=paste("Not enough single copy orthologs were found on ",SCAF),x="location",y="clus
}
plots[[1]]/plots[[5]]/plots[[9]]/plots[[13]]/plots[[17]]/pempty[[2]]/pempty[[3]]/plots[[26]]/pempty[[6]]
plots[[2]]/plots[[6]]/plots[[10]]/plots[[14]]/plots[[18]]/plots[[20]]/plots[[23]]/plots[[27]]/pempty[[7]
plots[[3]]/plots[[7]]/plots[[11]]/plots[[15]]/plots[[19]]/plots[[21]]/plots[[24]]/pempty[[4]]/pempty[[8]
plots[[4]]/plots[[8]]/plots[[12]]/plots[[16]]/pempty[[1]]/plots[[22]]/plots[[25]]/pempty[[5]]/pempty[[9]
```

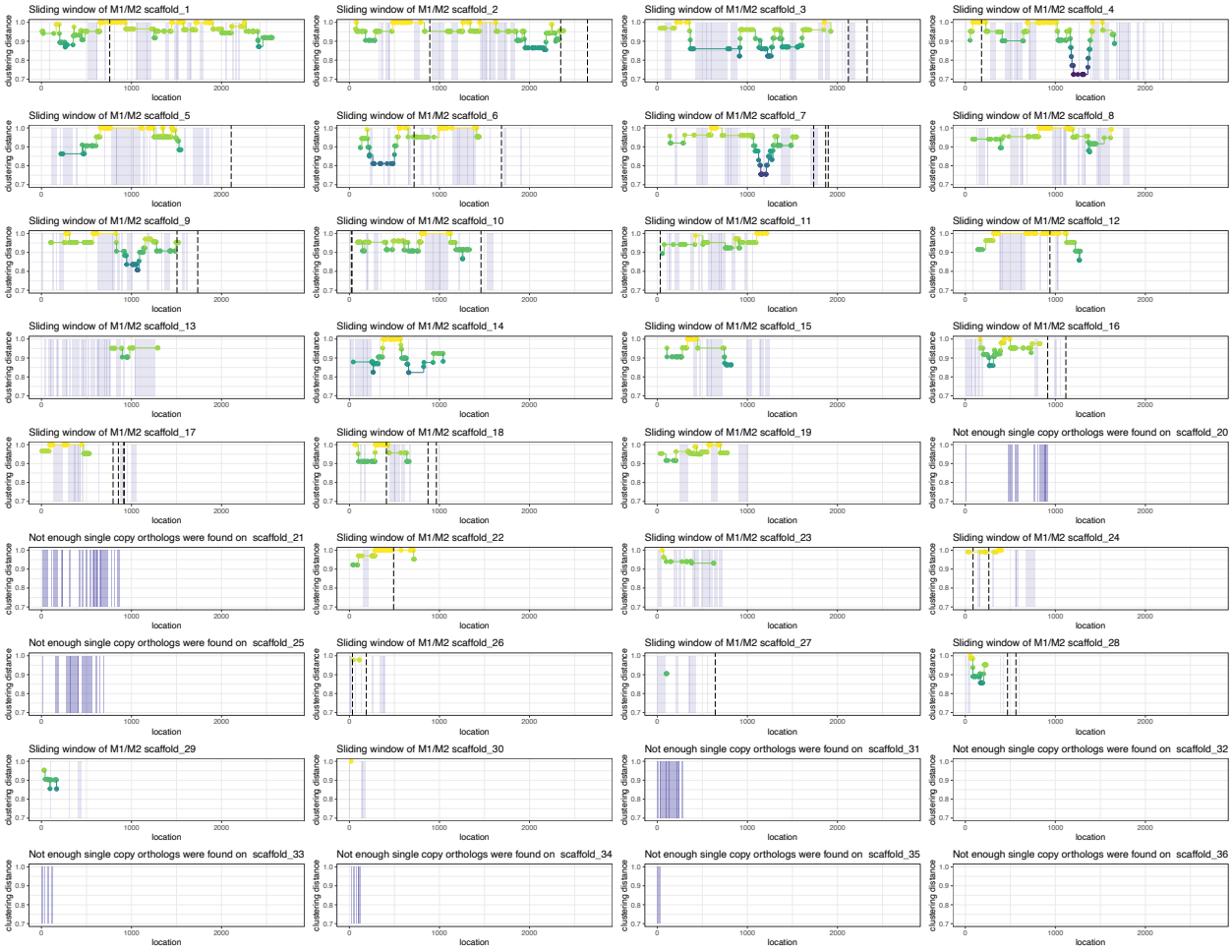


Average LRT M1/M2 (M1 = Almost Neutral, M2 = Positive selection) over a sliding window

```
logfile<-NULL
swindow<-NULL
```

```
step=20
foo.genes<-genes[rownames(ogs),]
plots<-list()
limite<-length(pe_genome[[1]])/1000
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2))
  {
    foo.dnds<-ogs$uno[foo.genes$scaffold==SCAF]
    foo.loc<-as.numeric(as.character(foo.genes$start[foo.genes$scaffold==SCAF]))/1000
    if(length(foo.dnds)>=21)
    {
      for (i in 1:(length(foo.dnds)-step))
        {
        swindow<-c(swindow,mean(foo.dnds[c(i:(i+step))]))
        }
        logfile<-rbind(logfile,cbind(swindow,foo.loc[1:length(swindow)],as.numeric(sapply(strsplit(SCAF
        culo<-lrar[lrar$Name==SCAF,]
        p<-ggplot(
          data.frame(mean_dnds=swindow,x=foo.loc[1:length(swindow)]),
          aes(x=x, y=mean_dnds, color=mean_dnds)) +
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0.7,ymax=1,alpha =.1,fill = spec
          geom_point(size=2) +
          geom_step() +
          scale_color_continuous(type = "viridis", limits = range(0.70,1)) +
          xlim(c(0,limite)) + ylim(c(0.7,1)) +
          theme_bw() +
          labs (title=paste("Sliding window of M1/M2",SCAF),x="location",y="clustering distance")
    if (length(unlist(het[het[,1]==SCAF,]))!=0)
    {
          plots[[SCAF]]<- p + geom_vline(xintercept = as.numeric(as.character(het[het[,1]==SCAF,2]))/10
    }else{
    plots[[SCAF]]<-p + theme(legend.position = "none")
      }
    }
    swindow<-NULL
}
logfile_m1m2<-logfile
```

```
pempty<-list()
for (SCAF in c("scaffold_20","scaffold_21","scaffold_25","scaffold_31","scaffold_32","scaffold_33","sca
  {
    culo<-lrar[lrar$Name==SCAF,]
    pempty[[SCAF]]<-ggplot(
          data.frame(mean_dist=c(5,5,5),x=c(0,1000,10000)),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(0.7,1))+theme_bw()+
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0.7,ymax=1,alpha = .3,fill = spe
          labs (title=paste("Not enough single copy orthologs were found on ",SCAF),x="location",y="clus
}
plots[[1]]/plots[[5]]/plots[[9]]/plots[[13]]/plots[[17]]/pempty[[2]]/pempty[[3]]/plots[[26]]/pempty[[6]]
plots[[2]]/plots[[6]]/plots[[10]]/plots[[14]]/plots[[18]]/plots[[20]]/plots[[23]]/plots[[27]]/pempty[[7]
plots[[3]]/plots[[7]]/plots[[11]]/plots[[15]]/plots[[19]]/plots[[21]]/plots[[24]]/pempty[[4]]/pempty[[8]
plots[[4]]/plots[[8]]/plots[[12]]/plots[[16]]/pempty[[1]]/plots[[22]]/plots[[25]]/pempty[[5]]/pempty[[9]

## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

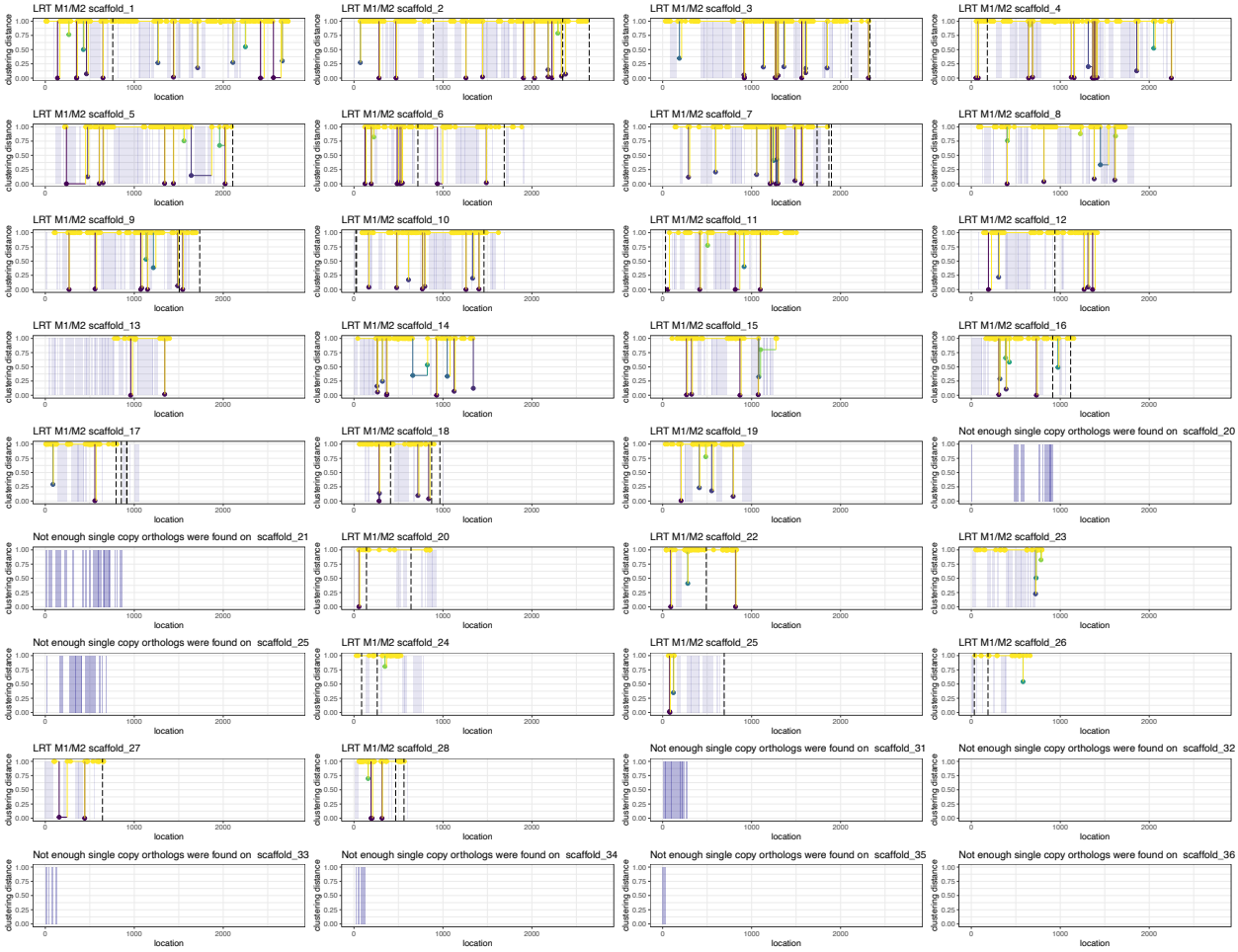


5.4.3 Orthologwise values for LRT M1/M2 (M1 = Almost Neutral, M2 = Positive selection)

```
logfile<-NULL
swindow<-NULL
foo.genes<-genes[rownames(ogs),]
plots<-list()
limite<-length(pe_genome[[1]])/1000
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2))
  {
    culo<-lrar[lrar$Name==SCAF,]
    foo.dnds<-ogs$uno[foo.genes$scaffold==SCAF]
    foo.loc<-as.numeric(as.character(foo.genes$start[foo.genes$scaffold==SCAF]))/1000
        p<-ggplot(
          data.frame(dnds=foo.dnds,x=foo.loc),
          aes(x=x, y=dnds, color=dnds)) +
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0,ymax=1,alpha =.1,fill = spectru
          geom_point(size=2) +
          geom_step() +
          scale_color_continuous(type = "viridis", limits = range(0,1)) +
          xlim(c(0,limite)) + ylim(c(0,1)) +
          theme_bw() +
          labs (title=paste("LRT M1/M2",SCAF),x="location",y="clustering distance")
    if (length(unlist(het[het[,1]==SCAF,]))!=0)
```

```
    {
          plots[[SCAF]]<- p + geom_vline(xintercept = as.numeric(as.character(het[het[,1]==SCAF,2]))/100
    }else{
    plots[[SCAF]]<-p + theme(legend.position = "none")
      }
    swindow<-NULL
}
```

```
pempty<-list()
for (SCAF in c("scaffold_20","scaffold_21","scaffold_25","scaffold_31","scaffold_32","scaffold_33","scaf
  {
    culo<-lrar[lrar$Name==SCAF,]
    pempty[[SCAF]]<-ggplot(
          data.frame(mean_dist=c(5,5,5),x=c(0,1000,10000)),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(0,1))+theme_bw()+
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0,ymax=1,alpha = .3,fill = specti
          labs (title=paste("Not enough single copy orthologs were found on ",SCAF),x="location",y="clus
}
plots[[1]]/plots[[5]]/plots[[9]]/plots[[13]]/plots[[17]]/pempty[[2]]/pempty[[3]]/plots[[26]]/pempty[[6]]
plots[[2]]/plots[[6]]/plots[[10]]/plots[[14]]/plots[[18]]/plots[[20]]/plots[[23]]/plots[[27]]/pempty[[7]
plots[[3]]/plots[[7]]/plots[[11]]/plots[[15]]/plots[[19]]/plots[[21]]/plots[[24]]/pempty[[4]]/pempty[[8]
plots[[4]]/plots[[8]]/plots[[12]]/plots[[16]]/pempty[[1]]/plots[[22]]/plots[[25]]/pempty[[5]]/pempty[[9]
```



5.4.4 Sliding window of LTR M7/M8

```
logfile<-NULL
swindow<-NULL
step=20
foo.genes<-genes[rownames(ogs),]
plots<-list()
limite<-length(pe_genome[[1]])/1000
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2)
  {
    foo.dnds<-ogs$dos[foo.genes$scaffold==SCAF]
    foo.loc<-as.numeric(as.character(foo.genes$start[foo.genes$scaffold==SCAF]))/1000
    if(length(foo.dnds)>=21)
    {
      for (i in 1:(length(foo.dnds)-step))
        {
        swindow<-c(swindow,mean(foo.dnds[c(i:(i+step))]))
        }
        logfile<-rbind(logfile,cbind(swindow,foo.loc[1:length(swindow)],as.numeric(sapply(strsplit(SCAF
        culo<-lrar[lrar$Name==SCAF,]
        p<-ggplot(
          data.frame(mean_dnds=swindow,x=foo.loc[1:length(swindow)]),
          aes(x=x, y=mean_dnds, color=mean_dnds)) +
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0.05,ymax=0.61,alpha =.1,fill = s
          geom_point(size=2) +
          geom_step() +
          scale_color_continuous(type = "viridis", limits = range(0.05,0.61)) +
          xlim(c(0,limite)) + ylim(c(0.05,0.61)) +
          theme_bw() +
          labs (title=paste("Sliding window of M1/M2",SCAF),x="location",y="clustering distance")
    if (length(unlist(het[het[,1]==SCAF,]))!=0)
    {
          plots[[SCAF]]<- p + geom_vline(xintercept = as.numeric(as.character(het[het[,1]==SCAF,2]))/100
    }else{
    plots[[SCAF]]<-p + theme(legend.position = "none")
      }
    }
    swindow<-NULL
}
logfile_m7m8<-logfile
```

```
pempty<-list()
for (SCAF in c("scaffold_20","scaffold_21","scaffold_25","scaffold_31","scaffold_32","scaffold_33","scaf
  {
    culo<-lrar[lrar$Name==SCAF,]
    pempty[[SCAF]]<-ggplot(
          data.frame(mean_dist=c(5,5,5),x=c(0,1000,10000)),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(0.7,1))+theme_bw()+
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0.05,ymax=0.61,alpha = .3,fill =
          labs (title=paste("Not enough single copy orthologs were found on ",SCAF),x="location",y="clus
}
plots[[1]]/plots[[5]]/plots[[9]]/plots[[13]]/plots[[17]]/pempty[[2]]/pempty[[3]]/plots[[26]]/pempty[[6]]
plots[[2]]/plots[[6]]/plots[[10]]/plots[[14]]/plots[[18]]/plots[[20]]/plots[[23]]/plots[[27]]/pempty[[7]
plots[[3]]/plots[[7]]/plots[[11]]/plots[[15]]/plots[[19]]/plots[[21]]/plots[[24]]/pempty[[4]]/pempty[[8]
```

```
plots[[4]]/plots[[8]]/plots[[12]]/plots[[16]]/pempty[[1]]/plots[[22]]/plots[[25]]/pempty[[5]]/pempty[[9
```

```
## Warning: Removed 28 rows containing missing values (geom_rect).
```

```
## Warning: Removed 13 rows containing missing values (geom_rect).
```

```
## Warning: Removed 4 rows containing missing values (geom_rect).
```

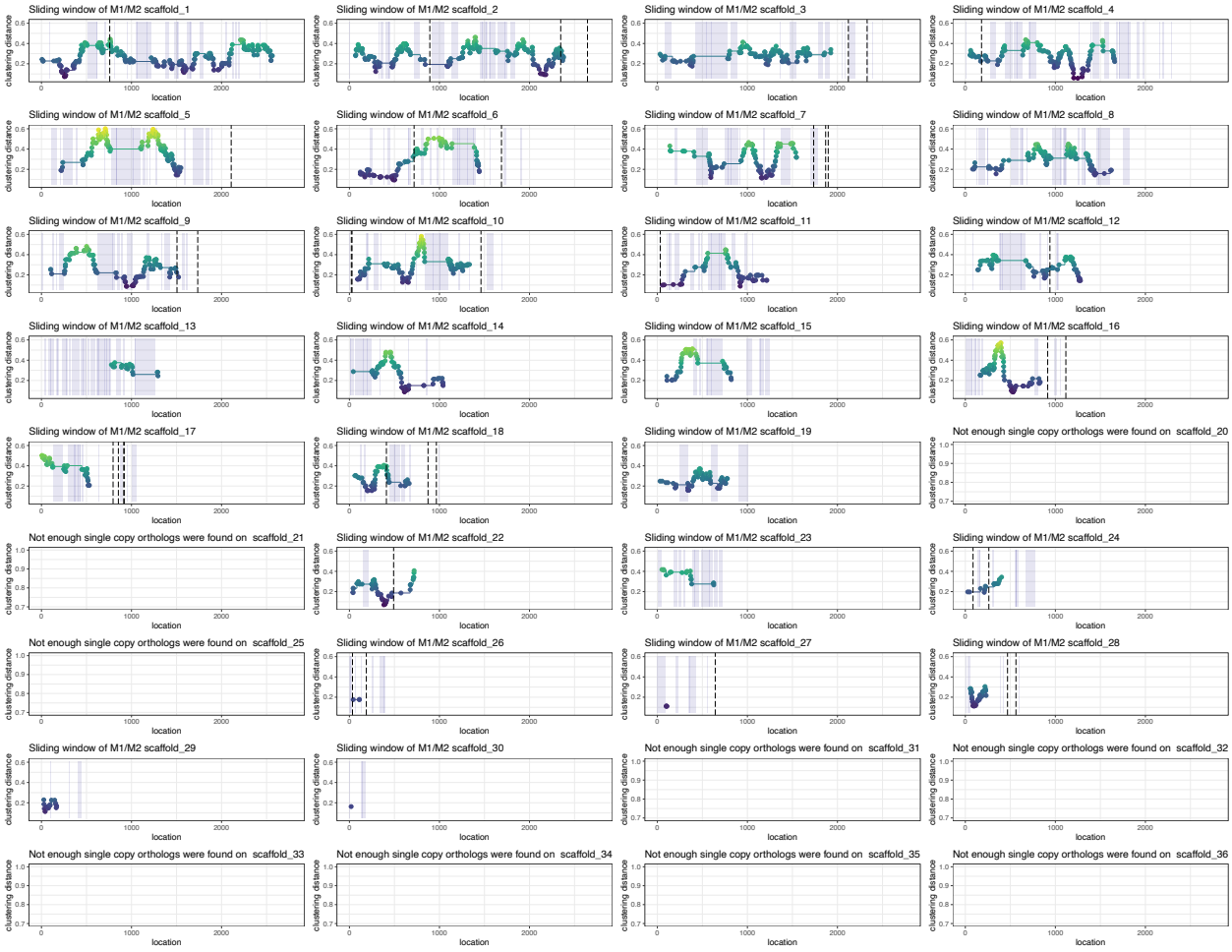
```
## geom_path: Each group consists of only one observation. Do you need to adjust
## the group aesthetic?
```

```
## Warning: Removed 6 rows containing missing values (geom_rect).
```

```
## Warning: Removed 14 rows containing missing values (geom_rect).
```

```
## Warning: Removed 1 rows containing missing values (geom_rect).
```

```
## Warning: Removed 10 rows containing missing values (geom_rect).
```

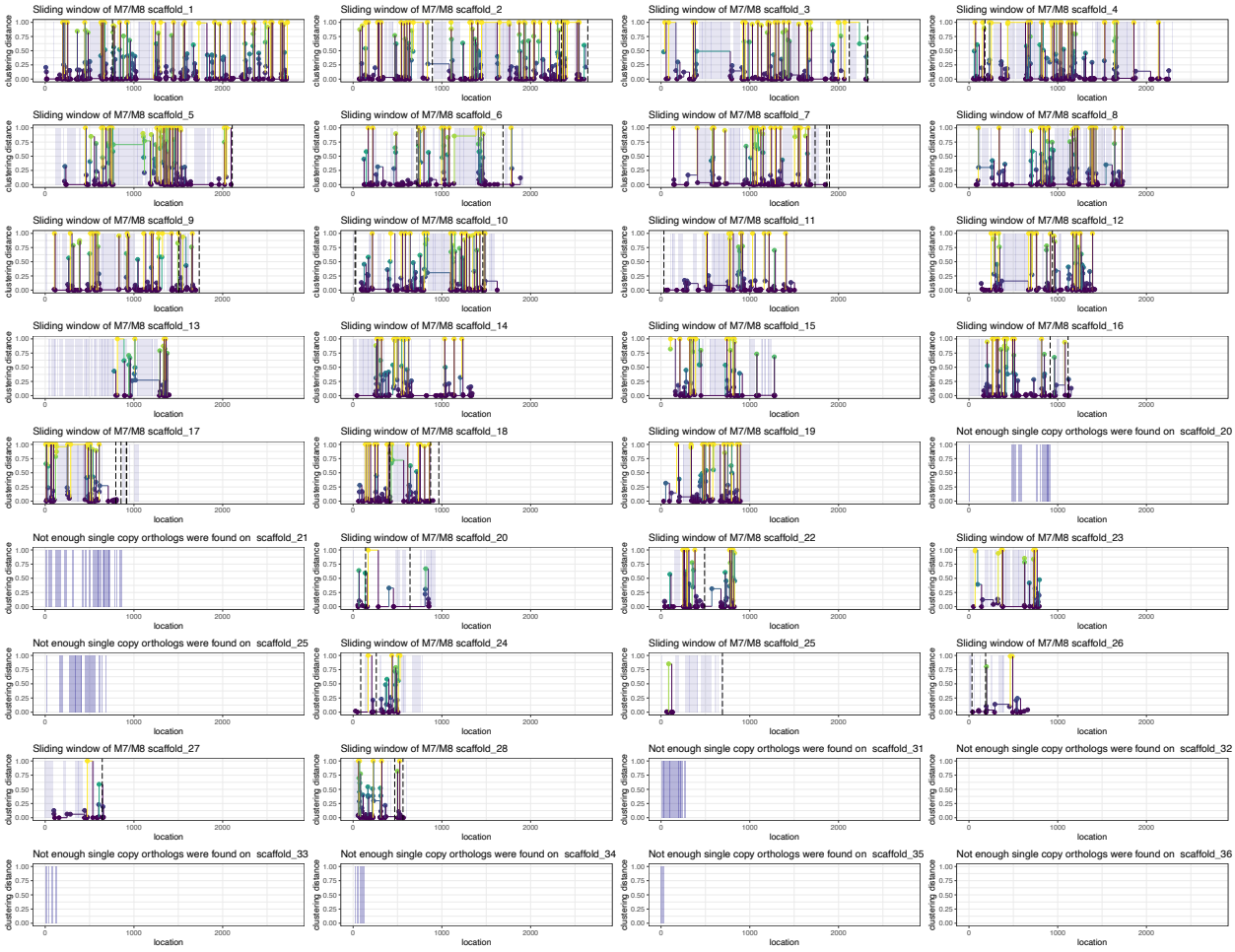


5.4.5 LRT M7/M8

```
logfile<-NULL
swindow<-NULL
foo.genes<-genes[rownames(ogs),]
plots<-list()
limite<-length(pe_genome[[1]])/1000
```

```
for (SCAF in levels(genes$scaffold)[order(as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2)
  {
    culo<-lrar[lrar$Name==SCAF,]
    foo.dnds<-ogs$dos[foo.genes$scaffold==SCAF]
    foo.loc<-as.numeric(as.character(foo.genes$start[foo.genes$scaffold==SCAF]))/1000
        p<-ggplot(
          data.frame(dnds=foo.dnds,x=foo.loc),
          aes(x=x, y=dnds, color=dnds)) +
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0,ymax=1,alpha =.1,fill = spectru
          geom_point(size=2) +
          geom_step() +
          scale_color_continuous(type = "viridis", limits = range(0,1)) +
          xlim(c(0,limite)) + ylim(c(0,1)) +
          theme_bw() +
          labs (title=paste("Sliding window of M7/M8",SCAF),x="location",y="clustering distance")
    if (length(unlist(het[het[,1]==SCAF,]))!=0)
    {
          plots[[SCAF]]<- p + geom_vline(xintercept = as.numeric(as.character(het[het[,1]==SCAF,2]))/100
    }else{
    plots[[SCAF]]<-p+ theme(legend.position = "none")
      }
    swindow<-NULL
}
```

```
pempty<-list()
for (SCAF in c("scaffold_20","scaffold_21","scaffold_25","scaffold_31","scaffold_32","scaffold_33","scaf
  {
    culo<-lrar[lrar$Name==SCAF,]
    pempty[[SCAF]]<-ggplot(
          data.frame(mean_dist=c(5,5,5),x=c(0,1000,10000)),
          aes(x=x, y=mean_dist, color=mean_dist)) + xlim(c(0,limite)) + ylim(c(0,1))+theme_bw()+
          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0,ymax=1,alpha = .3,fill = spectr
          labs (title=paste("Not enough single copy orthologs were found on ",SCAF),x="location",y="clus
}
plots[[1]]/plots[[5]]/plots[[9]]/plots[[13]]/plots[[17]]/pempty[[2]]/pempty[[3]]/plots[[26]]/pempty[[6]]
plots[[2]]/plots[[6]]/plots[[10]]/plots[[14]]/plots[[18]]/plots[[20]]/plots[[23]]/plots[[27]]/pempty[[7]
plots[[3]]/plots[[7]]/plots[[11]]/plots[[15]]/plots[[19]]/plots[[21]]/plots[[24]]/pempty[[4]]/pempty[[8]
plots[[4]]/plots[[8]]/plots[[12]]/plots[[16]]/pempty[[1]]/plots[[22]]/plots[[25]]/pempty[[5]]/pempty[[9]
```



Circos plot distance and dn/ds

```
ranges_phylodist<-makeGRangesFromDataFrame(data.frame(chr=paste("scaffold_",logfile_distances[,3],sep="
                        keep.extra.columns=TRUE,
                        ignore.strand=FALSE,
                        seqinfo=NULL,
                        seqnames.field="chr",
                        start.field="start",
                        end.field="end",
                        strand.field="strand",
                        starts.in.df.are.0based=FALSE)

ranges_dnds<-makeGRangesFromDataFrame(data.frame(chr=paste("scaffold_",logfile_dnds[,3],sep=""),start=lo
                        keep.extra.columns=TRUE,
                        ignore.strand=FALSE,
                        seqinfo=NULL,
                        seqnames.field="chr",
                        start.field="start",
                        end.field="end",
                        strand.field="strand",
                        starts.in.df.are.0based=FALSE)
ranges_m1m2<-makeGRangesFromDataFrame(data.frame(chr=paste("scaffold_",logfile_m1m2[,3],sep=""),start=lo
                        keep.extra.columns=TRUE,
                        ignore.strand=FALSE,
```

```
                          seqinfo=NULL,
                          seqnames.field="chr",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
ranges_m7m8<-makeGRangesFromDataFrame(data.frame(chr=paste("scaffold_",logfile_m7m8[,3],sep=""),start=lo
                          keep.extra.columns=TRUE,
                          ignore.strand=FALSE,
                          seqinfo=NULL,
                          seqnames.field="chr",
                          start.field="start",
                          end.field="end",
                          strand.field="strand",
                          starts.in.df.are.0based=FALSE)
```

```
shared.ogs<-strsplit(all.ogs$V5,split=", ")

shared.ogs<-melt(shared.ogs)

shared.ogs<-shared.ogs[grep("Pyrenodesmiaerodens",shared.ogs$value),]

shared.ogs$value<-as.character(shared.ogs$value)
shared.ogs$value<-gsub("-T1","",shared.ogs$value)
shared.ogs<-shared.ogs[shared.ogs[,2]%in%shared.ogs[,2][duplicated(shared.ogs[,2])],]
foo_uno<-unlist(strsplit(genes.all[shared.ogs$value,1],":"))
dim(foo_uno)<-c(2,length(foo_uno)/2)
foo_uno<-t(foo_uno)
shared.ogs<-cbind(shared.ogs,scaf=foo_uno[,1],start=sapply(strsplit(foo_uno[,2],"-"),`[`,1),end=sapply(s
#shared.ogs$scaf<-as.numeric(gsub("scaffold_","",shared.ogs$scaf))
```

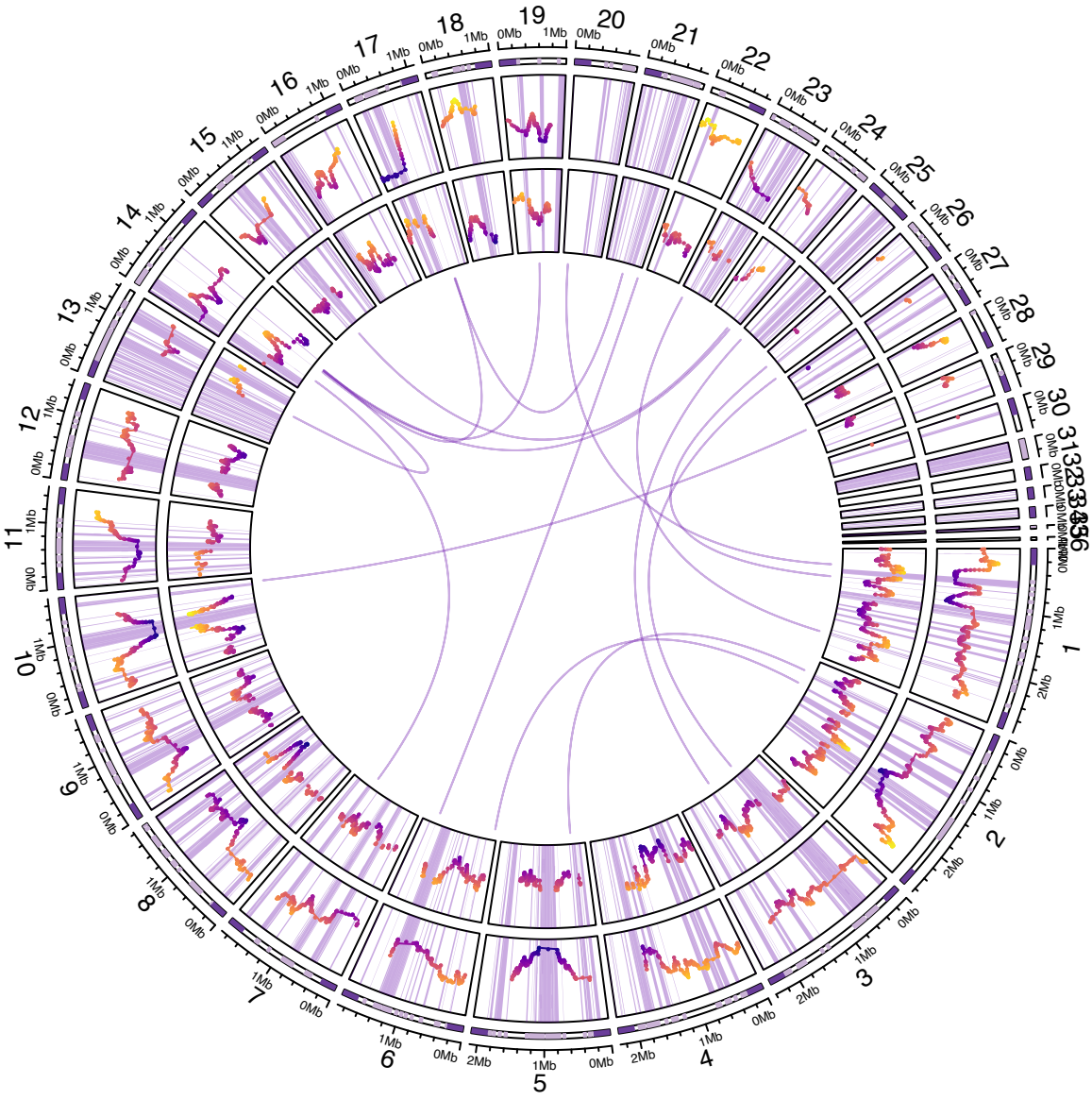
```
base.for.circos<-function(GENOME,LINKS,LOST=lost_genes)
{
require(circlize)
#------------------
# Initialize
#------------------
circos.genomicInitialize(data.frame(names=names(GENOME),start=0,end=sapply(GENOME,length)),sector.names
circos.genomicTrack(ylim=c(0,1), track.height=0.01,bg.col=NA,bg.border=TRUE,cell.padding=c(0,0,0,0))
# First track LRAR
for (i in names(GENOME))
{
  foo<-lrar_flanks[lrar_flanks@seqnames==i,]
    if (length(foo)>0)
  {
    circos.genomicRect(cbind(foo@ranges@start,foo@ranges@start+foo@ranges@width), sector.index = i,ytop
  }
    foo<-telomere_locs[telomere_locs@seqnames==i,]
    if (length(foo)>0)
  {
    circos.genomicRect(cbind(foo@ranges@start,foo@ranges@start+foo@ranges@width), sector.index = i,ytop
    }
}
#
```

```
circos.genomicTrack(ylim=c(2.5,12.5), track.height=0.15,bg.col=NA,bg.border=TRUE,cell.padding=c(0,0,0,0
# First track
foo2<-as.data.frame(ranges_phylodist)
foo2<-cbind(foo2,
              color=spectrum[round(30*(foo2$dist-min(foo2$dist))/(max(foo2$dist)-min(foo2$dist)))+1])
for (i in names(GENOME))
{
  foo<-lrar[lrar$Name==i,]
  foo3<-foo2[foo2$seqnames==i,]
  foo3$end<-c(foo3$start[-1],foo3$start[dim(foo3)[1]])
  if (dim(foo)[1]>0)
  {
    circos.genomicRect(foo[,c(2,3)], sector.index = i,ytop = 12.5, ybottom = 2.5, col=c("#6001A655"), bo
    if (dim(foo3)[1]>0)
  {
    circos.genomicPoints(foo3, value=foo3$dist, col = foo3$color, bg = foo3$color, count_by = "number",
    circos.segments(foo3$start,foo3$dist,foo3$end,c(foo3$dist[-1],foo3$dist[dim(foo3)[1]]), col = foo3$c
    }
  }
}
#
circos.genomicTrack(ylim=c(0.05,0.25), track.height=0.15,bg.col=NA,bg.border=TRUE,cell.padding=c(0,0,0,0
# First track LRAR
foo2<-as.data.frame(ranges_dnds)
foo2<-cbind(foo2,
              color=spectrum[round(30*(foo2$dist-min(foo2$dist))/(max(foo2$dist)-min(foo2$dist)))+1])
for (i in names(GENOME))
{
  foo<-lrar[lrar$Name==i,]
  foo3<-foo2[foo2$seqnames==i,]

  if (dim(foo)[1]>0)
  {
    circos.genomicRect(foo[,c(2,3)], sector.index = i,ytop = 0.25, ybottom = 0.05, col=c("#6001A655"), b

    if (dim(foo3)[1]>0)
  {
      circos.genomicPoints(foo3, value=foo3$dist, col = foo3$color, bg = foo3$color, count_by = "number"
    circos.segments(foo3$start,foo3$dist,foo3$end,c(foo3$dist[-1],foo3$dist[dim(foo3)[1]]), col = foo3$c
    }
  }
}
# third track links orthogroups
for (OG in levels(factor(LINKS$L1)))
{
  foo4<-LINKS[LINKS$L1==OG,]
  extent<-dim(foo4)[1]
  for(i in c(1:(extent-1)))
  {
    for (j in c((i+1):extent))
    {
      if(foo4$scaf[i]!=foo4$scaf[j])
        {
```

```
        circos.link(foo4$scaf[i],c(as.numeric(foo4$start[i]),as.numeric(foo4$end[i])),foo4$scaf[j],c(as
      }
    }
  }
}
  }

base.for.circos(pe_genome,shared.ogs,lost_genes)
```

```
## Note: 2 points are out of plotting region in sector 'scaffold_1', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_2', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_3', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_4', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_5', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_6', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_7', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_8', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_9', track
## '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_10',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_11',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_12',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_14',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_15',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_16',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_17',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_18',
## track '2'.
## Note: 2 points are out of plotting region in sector 'scaffold_22',
## track '2'.
```

```
## Note: 2 points are out of plotting region in sector 'scaffold_24',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_25',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_26',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_27',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_28',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_29',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_32',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_33',
## track '2'.

## Note: 2 points are out of plotting region in sector 'scaffold_34',
## track '2'.
```



5.4.6 Is phylogenetic discordance correlated to the proliferation of transposable elements in the neighbouring LRARs?

```
ranges_results2<-NULL
for (REGION in c("general","telomeric","LRAR"))
  {
  if (REGION=="general")
    {ranges2use<-non_telomeric_regions
  }else if (REGION=="telomeric")
    {ranges2use<-telomere_locs
  }else if (REGION=="LRAR")
  {ranges2use<-lrar_flanks
    }
  for (TYPE in c("distance","dnds","m1m2","m7m8"))
  {
   if (REGION=="general"&TYPE=="distance")
```

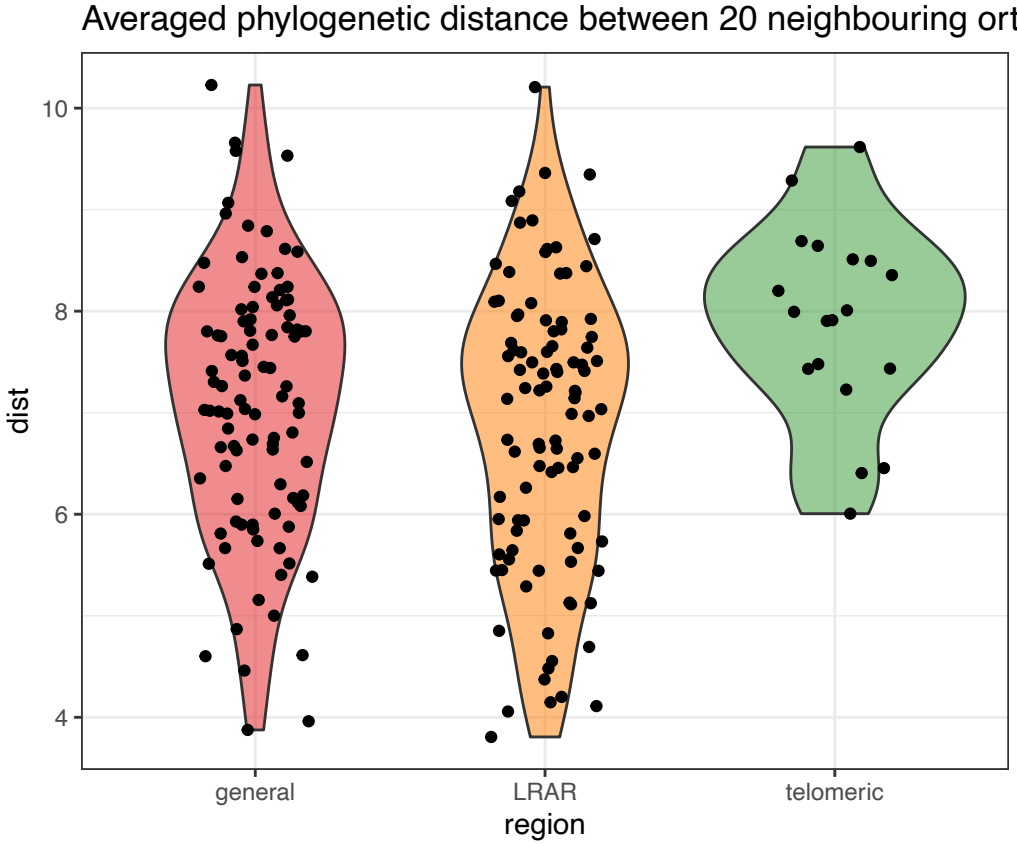
```
    {data2use<-subsetByOverlaps(ranges_phylodist,non_telomeric_regions)
  }else if (REGION=="telomeric"&TYPE=="distance")
    {data2use<-subsetByOverlaps(ranges_phylodist,telomere_locs)
  }else if (REGION=="LRAR"&TYPE=="distance")
  {data2use<-subsetByOverlaps(ranges_phylodist,lrar_flanks)
    } else if (REGION=="general"&TYPE=="dnds")
    {data2use<-subsetByOverlaps(ranges_dnds,non_telomeric_regions)
  }else if (REGION=="telomeric"&TYPE=="dnds")
    {data2use<-subsetByOverlaps(ranges_dnds,telomere_locs)
  }else if (REGION=="LRAR"&TYPE=="dnds")
  {data2use<-subsetByOverlaps(ranges_dnds,lrar_flanks)
    } else    if (REGION=="general"&TYPE=="m1m2")
    {data2use<-subsetByOverlaps(ranges_m1m2,non_telomeric_regions)
  }else if (REGION=="telomeric"&TYPE=="m1m2")
    {data2use<-subsetByOverlaps(ranges_m1m2,telomere_locs)
  }else if (REGION=="LRAR"&TYPE=="m1m2")
  {data2use<-subsetByOverlaps(ranges_m1m2,lrar_flanks)
    } else    if (REGION=="general"&TYPE=="m7m8")
    {data2use<-subsetByOverlaps(ranges_m7m8,non_telomeric_regions)
  }else if (REGION=="telomeric"&TYPE=="m7m8")
    {data2use<-subsetByOverlaps(ranges_m7m8,telomere_locs)
  }else if (REGION=="LRAR"&TYPE=="m7m8")
  {data2use<-subsetByOverlaps(ranges_m7m8,lrar_flanks)
    }
for (I in 1:length(ranges2use))
{
if (length(data2use$dist[attr(findOverlaps(data2use,ranges2use[I]),"from")])>0)
{
  ranges_results2<-rbind(ranges_results2,cbind(REGION,TYPE,I,mean(  data2use$dist[attr(findOverlaps(data
}
    }
  }
}
```

```
## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_20, scaffold_21, scaffold_25, scaffold_32, scaffold_33, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).

## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_20, scaffold_21, scaffold_25, scaffold_32, scaffold_33, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).

## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24
##   - in 'y': scaffold_20, scaffold_21, scaffold_25, scaffold_32, scaffold_33, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).

## Warning in .Seqinfo.mergexy(x, y): Each of the 2 combined objects has sequence levels not in the othe
##   - in 'x': scaffold_23, scaffold_24
```

```
##   - in 'y': scaffold_20, scaffold_21, scaffold_25, scaffold_32, scaffold_33, scaffold_34
##   Make sure to always combine/compare objects based on the same reference
##   genome (use suppressWarnings() to suppress this warning).
```

```
ranges_results2<-data.frame(region=factor(ranges_results2[,1]),feature=factor(ranges_results2[,2]),loca
```

```
ggplot(ranges_results2[ranges_results2$feature=="distance",],aes(x=region,y=dist,fill=region,z=feature)
```

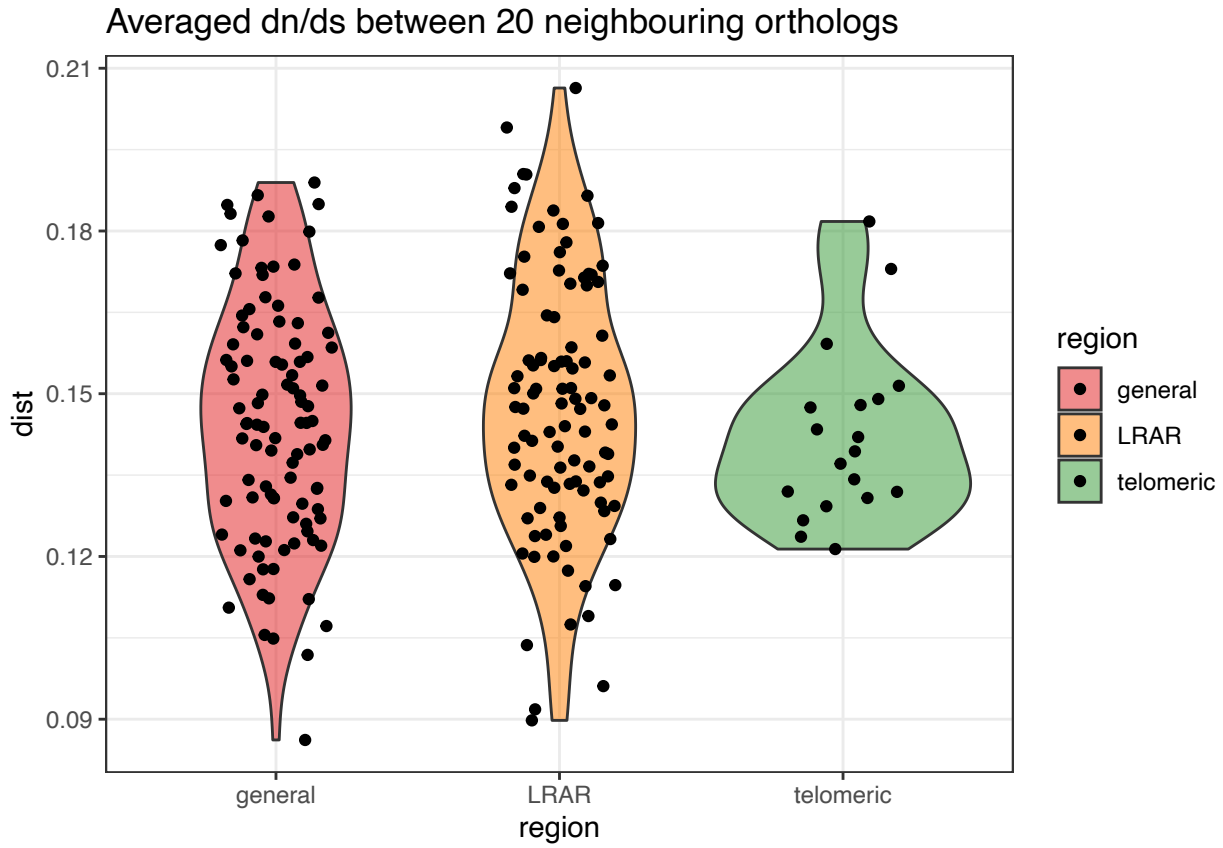


5.4.6.1 phylogenetic distance

```
pairwise.wilcox.test(ranges_results2[ranges_results2$feature=="distance","dist"],ranges_results2[ranges
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results2[ranges_results2$feature == "distance", "dist"] and ranges_results2[ranges_resu
##
##           general LRAR
## LRAR      0.2640  -
## telomeric 0.0104  0.0058
##
## P value adjustment method: BH
```

```
ggplot(ranges_results2[ranges_results2$feature=="dnds",],aes(x=region,y=dist,fill=region,z=feature))+ge
```

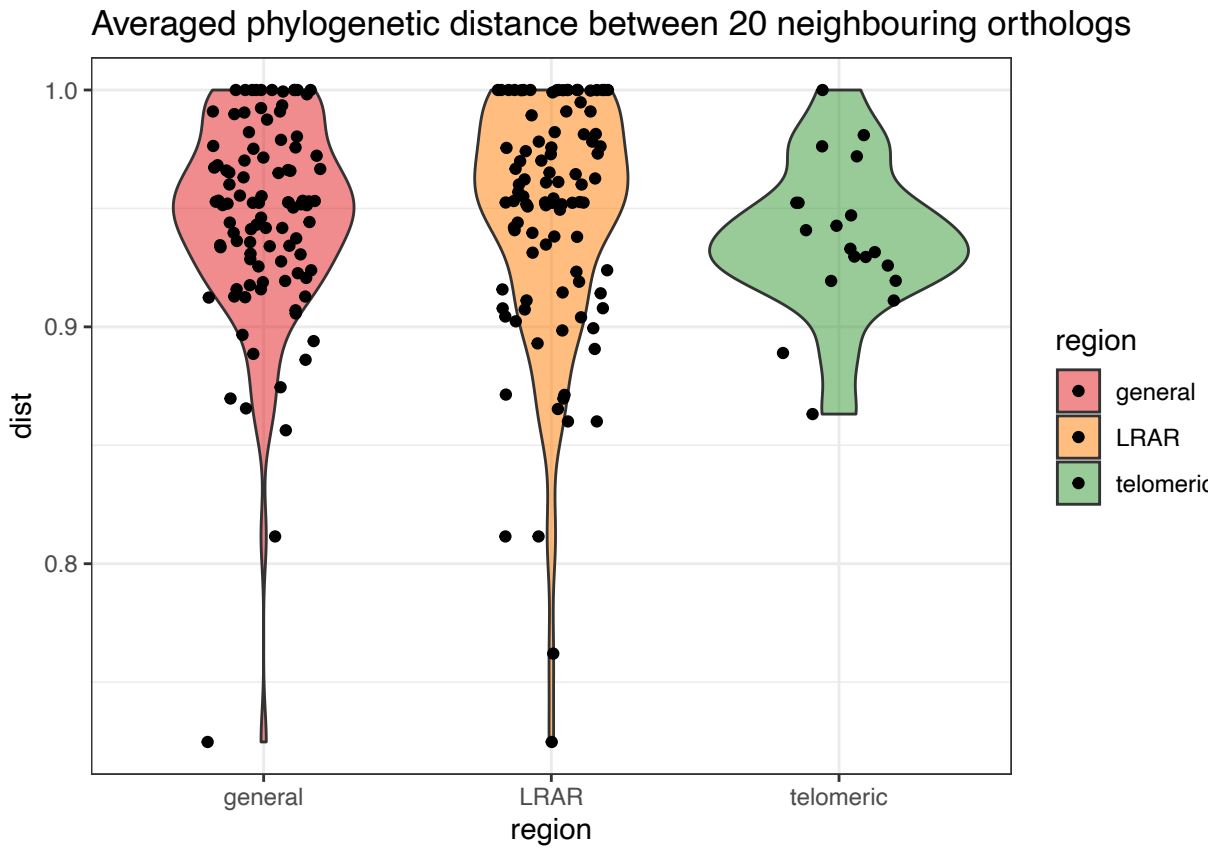


5.4.6.2 dn/ds

```
pairwise.wilcox.test(ranges_results2[ranges_results2$feature=="dnds","dist"],ranges_results2[ranges_resu
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results2[ranges_results2$feature == "dnds", "dist"] and ranges_results2[ranges_results
##
##           general LRAR
## LRAR      0.54    -
## telomeric 0.59    0.54
##
## P value adjustment method: BH
```

```
ggplot(ranges_results2[ranges_results2$feature=="m1m2",],aes(x=region,y=dist,fill=region,z=feature))+ge
```



5.4.6.3 m1/m2

```
pairwise.wilcox.test(ranges_results2[ranges_results2$feature=="m1m2","dist"],ranges_results2[ranges_resu
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results2[ranges_results2$feature == "m1m2", "dist"] and ranges_results2[ranges_results
##
##           general LRAR
## LRAR      0.46    -
## telomeric 0.24    0.24
##
## P value adjustment method: BH
```

```
ggplot(ranges_results2[ranges_results2$feature=="m7m8",],aes(x=region,y=dist,fill=region,z=feature))+ge
```

5.4.6.4 m7/m8

```
pairwise.wilcox.test(ranges_results2[ranges_results2$feature=="m7m8","dist"],ranges_results2[ranges_resu
```

```
##
##  Pairwise comparisons using Wilcoxon rank sum test with continuity correction
##
## data:  ranges_results2[ranges_results2$feature == "m7m8", "dist"] and ranges_results2[ranges_results
##
##           general LRAR
## LRAR      0.329   -
## telomeric 0.057   0.034
##
## P value adjustment method: BH
```

5.4.7 Is phylogenetic discordance correlated to the proliferation of transposable elements in the neighbouring LRARs?

```
ranges_lrar<-cbind(tabulate_rep_lrar[,1:4],mean_dist=0)
ranges_lrar$Start<-ranges_lrar$Start-50000
ranges_lrar$End<-ranges_lrar$End+50000
ranges_lrar$Start[ranges_lrar$Start<0]<-0
for (i in 1:dim(ranges_lrar)[1])
{
  ranges_lrar[i,"mean_dist"]<-mean(
    logfile_distances[logfile_distances[,3]==sapply(strsplit(ranges_lrar[i,1],"_"),`[`,2)&
                logfile_distances[,2]>=ranges_lrar[i,2]/1000&logfile_distances[,2]<=ranges_lrar[i,3]/100
              ,1])
}
```

```
#anova(lm(ranges_lrar$mean_dist~tabulate_rep_lrar[,14]*(tabulate_rep_lrar[,13]+tabulate_rep_lrar[,11])))
foo_reg<-cbind(tabulate_rep_lrar,mean_dist = ranges_lrar$mean_dist)
fit<-lm(mean_dist~`DNA/hAT-Ac`+`DNA/MULE-MuDR`+`Low_complexity`+`LTR/Copia`+`LTR/Gypsy`+`LTR/Ngaro`+`RC/
summary(fit)
```

```
##
## Call:
## lm(formula = mean_dist ~ `DNA/hAT-Ac` + `DNA/MULE-MuDR` + Low_complexity +
##     `LTR/Copia` + `LTR/Gypsy` + `LTR/Ngaro` + `RC/Helitron` +
##     Simple_repeat + Unknown + Unspecified, data = foo_reg)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.0039 -0.9938  0.0921  1.0380  3.8602
##
## Coefficients:
##                   Estimate Std. Error t value Pr(>|t|)
## (Intercept)      7.193e+00  1.837e-01  39.149  < 2e-16 ***
## `DNA/hAT-Ac`    -2.126e+03  2.976e+03  -0.714    0.476
## `DNA/MULE-MuDR` -1.123e+04  7.146e+03  -1.572    0.117
## Low_complexity   3.818e+02  1.665e+03   0.229    0.819
## `LTR/Copia`      3.172e+02  6.876e+02   0.461    0.645
## `LTR/Gypsy`     -8.054e+02  6.298e+02  -1.279    0.202
## `LTR/Ngaro`      7.217e+02  1.576e+03   0.458    0.647
## `RC/Helitron`   -2.917e+02  1.165e+03  -0.250    0.803
## Simple_repeat   -1.644e+02  5.668e+02  -0.290    0.772
## Unknown         -1.219e+03  2.909e+02  -4.191 3.75e-05 ***
## Unspecified      2.286e+03  2.175e+03   1.051    0.294
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.349 on 274 degrees of freedom
##   (250 observations deleted due to missingness)
## Multiple R-squared:  0.09912,    Adjusted R-squared:  0.06624
## F-statistic: 3.015 on 10 and 274 DF,  p-value: 0.00124
```

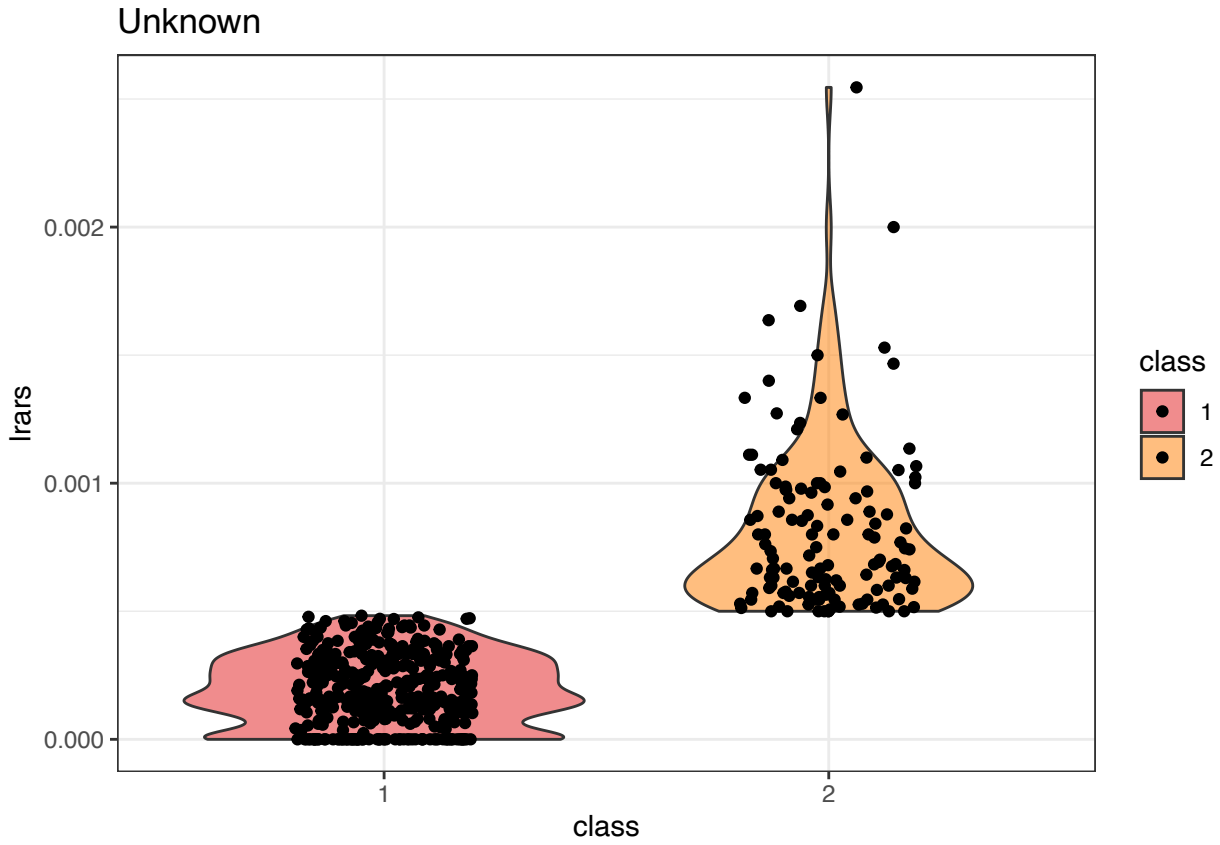
```
anova(fit)
```

```
## Analysis of Variance Table
##
## Response: mean_dist
##                  Df Sum Sq Mean Sq F value    Pr(>F)
## `DNA/hAT-Ac`      1   1.02  1.0184  0.5595  0.455090
## `DNA/MULE-MuDR`   1   4.04  4.0374  2.2182  0.137538
## Low_complexity    1   0.19  0.1856  0.1020  0.749703
## `LTR/Copia`       1   0.09  0.0888  0.0488  0.825368
## `LTR/Gypsy`       1  12.55 12.5467  6.8935  0.009137 **
## `LTR/Ngaro`       1   0.77  0.7731  0.4247  0.515127
## `RC/Helitron`     1   2.88  2.8796  1.5821  0.209526
## Simple_repeat     1   0.30  0.2991  0.1643  0.685504
## Unknown           1  31.03 31.0327 17.0501 4.838e-05 ***
## Unspecified       1   2.01  2.0093  1.1039  0.294330
## Residuals       274 498.71  1.8201
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

5.4.8 Unknown transposable elements (Chiefly ....)

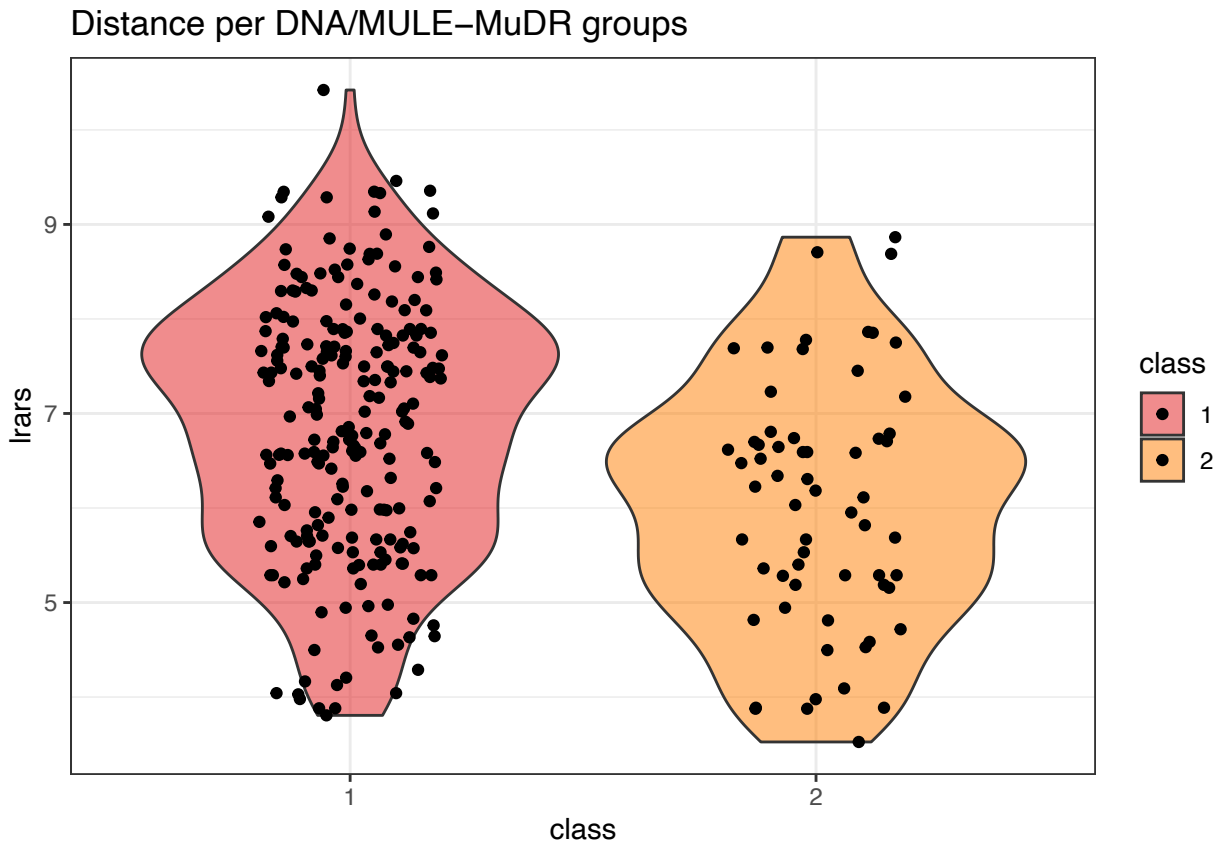
```
library(mclust)
grupos<-Mclust(tabulate_rep_lrar$Unknown,1:10)
ggplot(data.frame(lrars=tabulate_rep_lrar$Unknown,class=as.factor(grupos$classification)),aes(x=class,y
```




```
ggplot(data.frame(lrars=ranges_lrar$mean_dist,class=as.factor(grupos$classification)),aes(x=class,y=lra
```

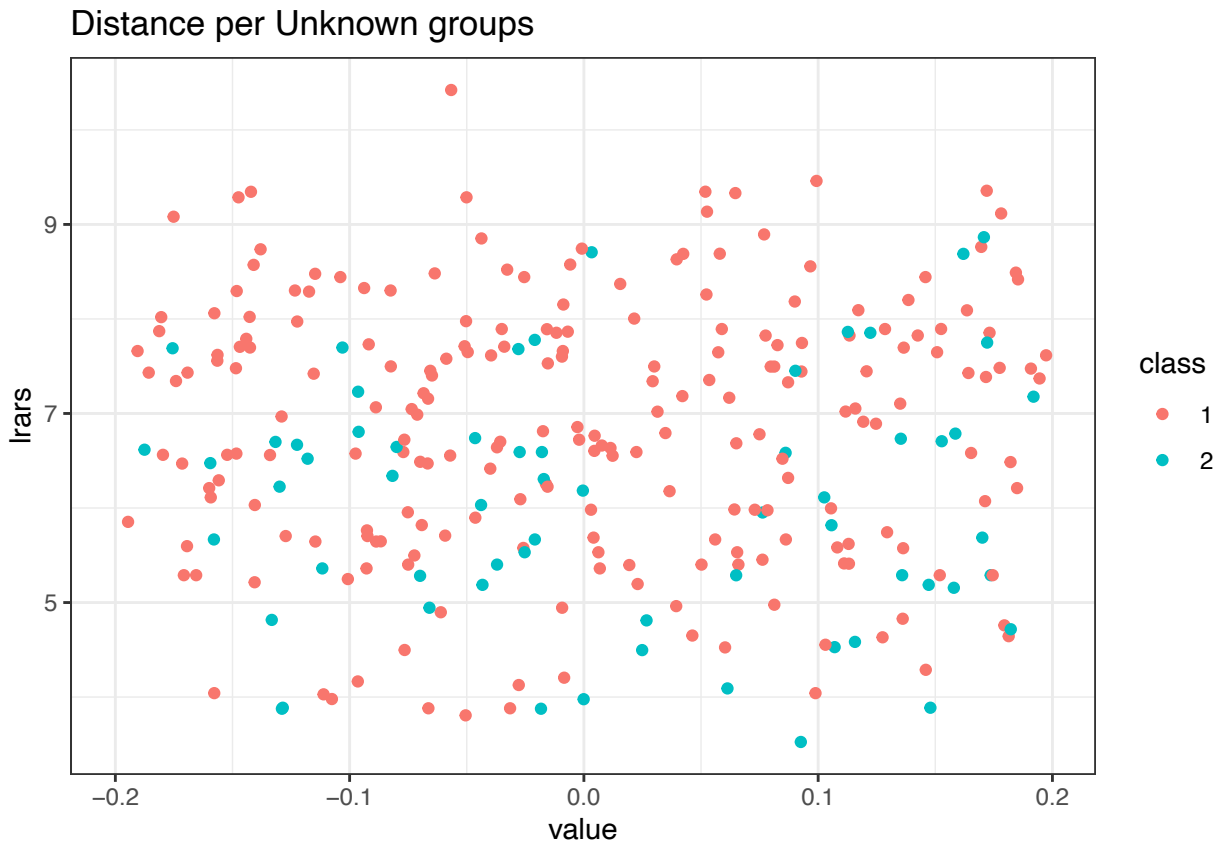
```
## Warning: Removed 250 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 250 rows containing missing values (geom_point).
```



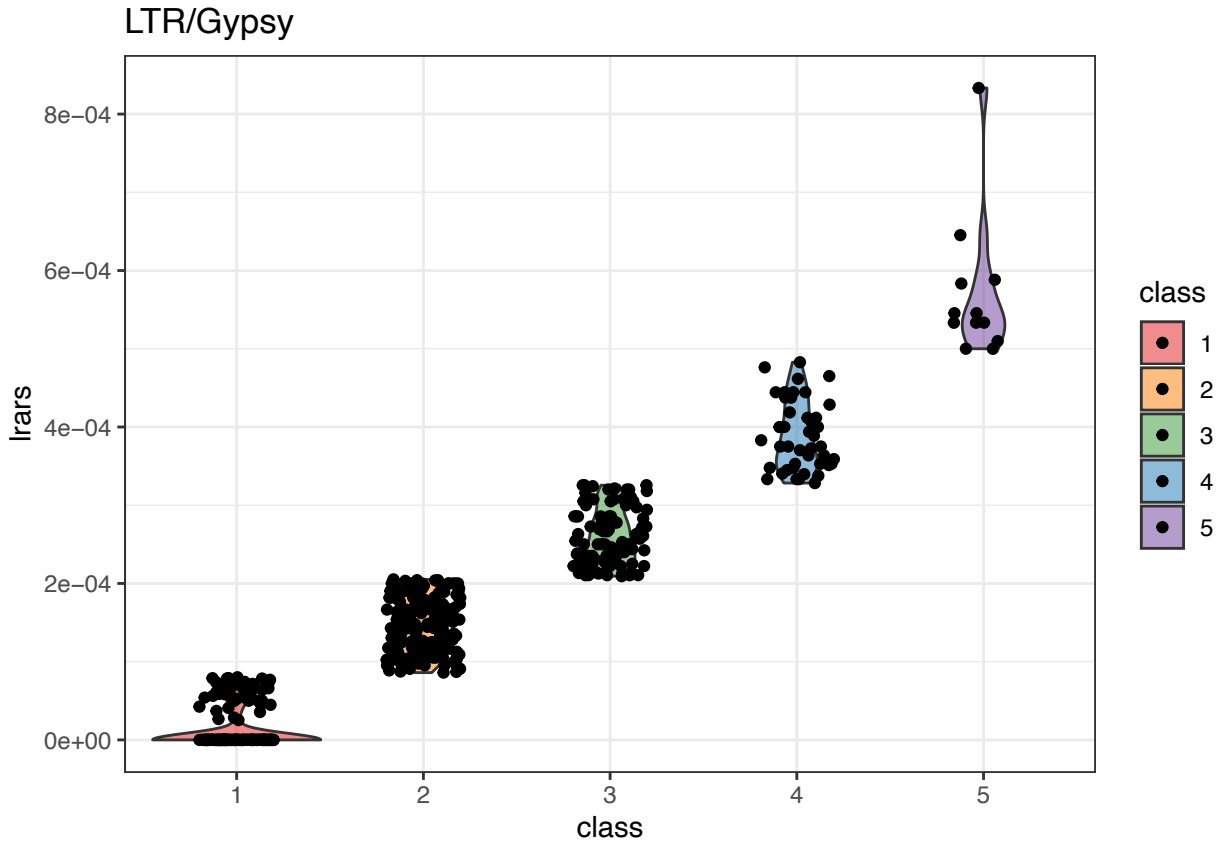

```
ggplot(data.frame(lrars=ranges_lrar$mean_dist,value=tabulate_rep_lrar$Unknown,class=as.factor(grupos$cla
```

```
## Warning: Removed 250 rows containing missing values (geom_point).
```



5.4.9 LTR Gipsy

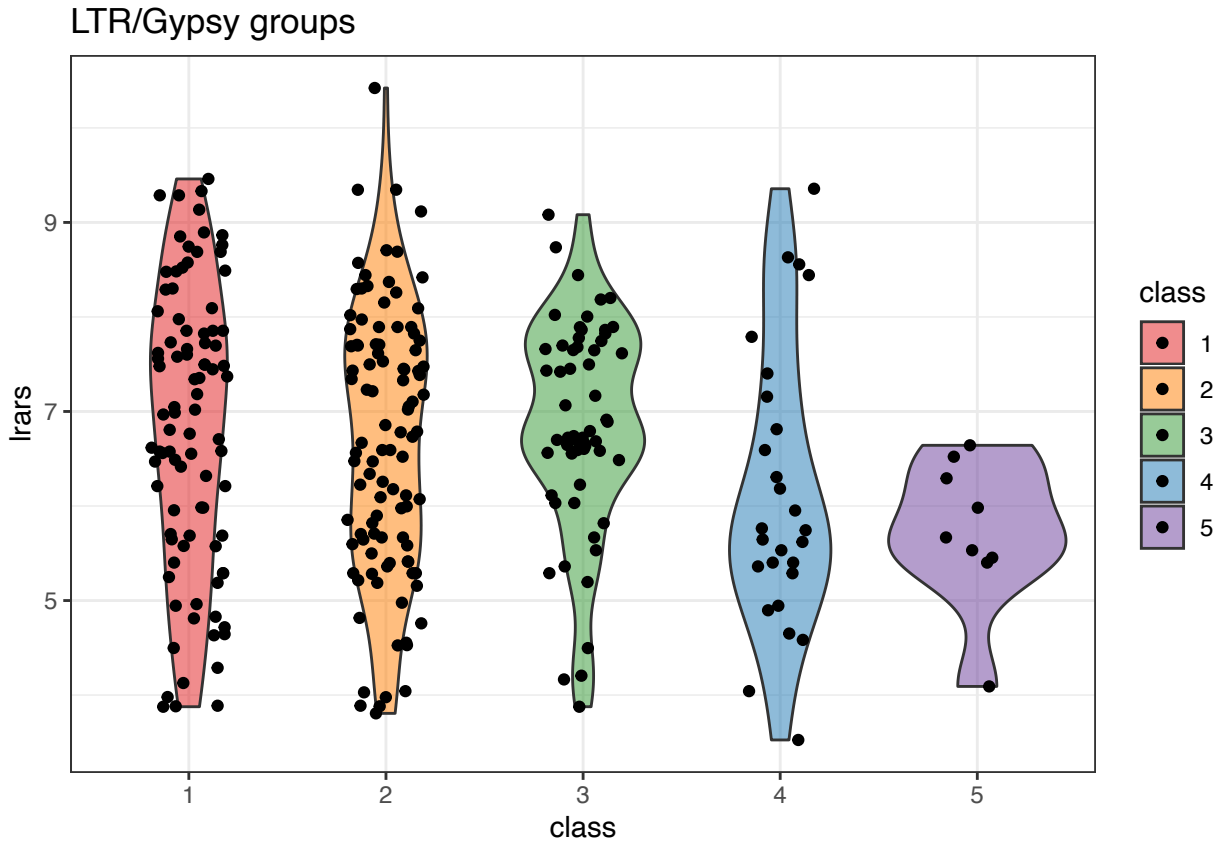
```
grupos<-Mclust(tabulate_rep_lrar$`LTR/Gypsy`,1:6)
ggplot(data.frame(lrars=tabulate_rep_lrar$`LTR/Gypsy`,class=as.factor(grupos$classification)),aes(x=cla
```




```
ggplot(data.frame(lrars=ranges_lrar$mean_dist,class=as.factor(grupos$classification)),aes(x=class,y=lra
```

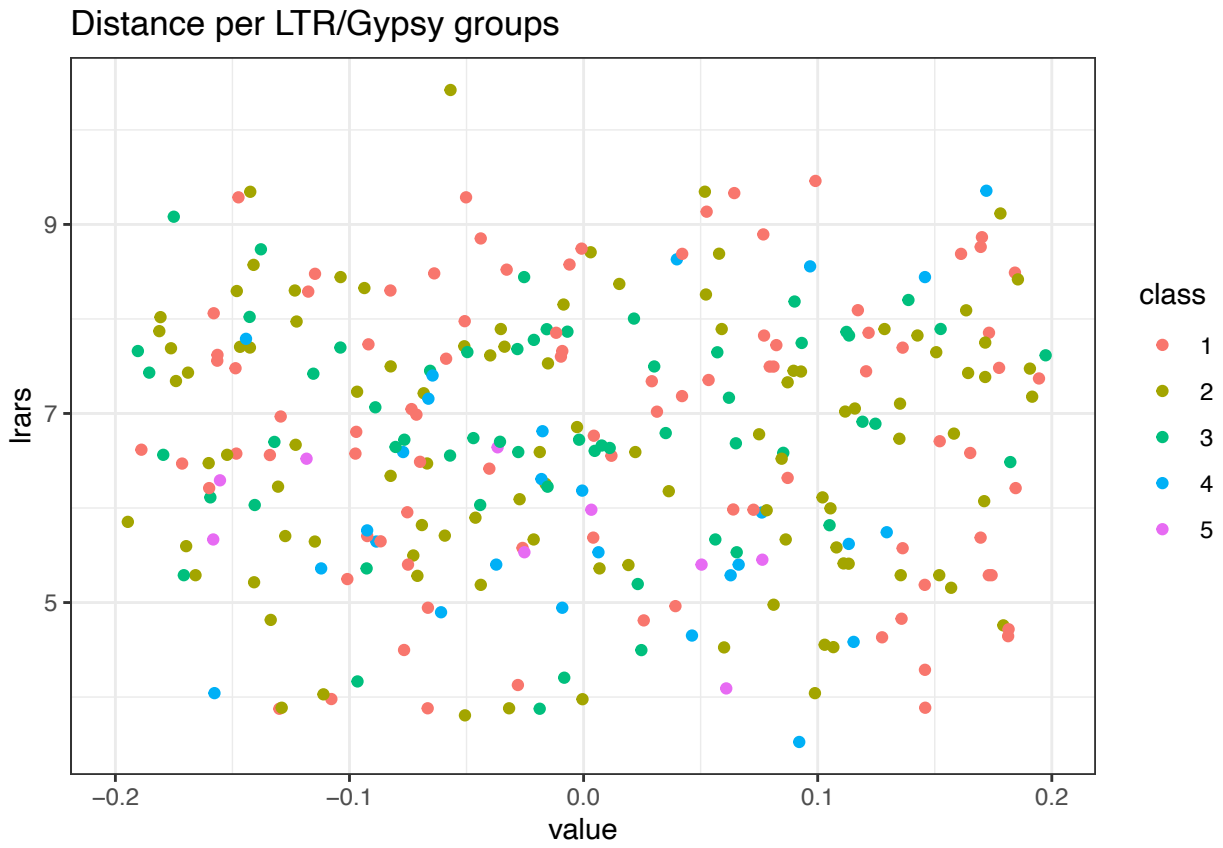
```
## Warning: Removed 250 rows containing non-finite values (stat_ydensity).
```

```
## Warning: Removed 250 rows containing missing values (geom_point).
```




```
ggplot(data.frame(lrars=ranges_lrar$mean_dist,value=tabulate_rep_lrar$`LTR/Gypsy`,class=as.factor(grupos
```

```
## Warning: Removed 250 rows containing missing values (geom_point).
```




```
#library(mclust)
#clust_steps<-Mclust(logfile,k=2:10)
#colnames(clust_steps$data)[3]<-"scaf"
```

```
#for (SCAF in as.numeric(sapply(strsplit(levels(genes$scaffold),"_"),`[`,2)))
#  {
#   culo<-lrar[lrar$Name==paste("scaffold_",SCAF,sep=""),]
#   p<-ggplot(
#          data.frame(cbind(clust_steps$data[clust_steps$data[,3]==SCAF,],classification=clust_steps$cl
#          aes(x=foo.loc, y=swindow, color=classification)) +
#          annotate("rect",xmin=culo$Start/1000,xmax=culo$End/1000,ymin=0,ymax=15,alpha = .1,fill = spe
#          geom_point(size=2) +
#          geom_step() +
#          scale_color_continuous(type = "viridis") +
#          theme_bw() +
#          labs (title=paste("Sliding window of clustering distances across",SCAF),x="location",y="clus
#print(p)
#}
```