TE content in CDS/Genes of a soft-masked reference genome

1/ Create a directory that will contain a file for each scaffold

mkdir fastafiles

cd fastafiles

perl multifasta2fastafiles [../infile_softmasked_refgenome.fasta]

(normally at this step you have a lot of files with a name: scaffold_name.fasta)

2/ Edit a little bit your gff

Extract the information you are interested in the gff (e.g. CDS, gene, etc…). Such a step is just a way to fix some issues related to the comment lines for instance.

awk '$3 == "CDS" {print $0}' [gff] > [gff.CDSonly]

If your geneID do not start by a “Name=” flag but by another flag (such as ID=), edit it:

sed 's/ID=/Name=/g' [gff.CDSonly] > [gff.CDSonly2]
(importantly, if you work on CDS, all exons need to have exactly the same ID, if not the script for the extraction will create a different file per exon)

***3/* Extract your sequences**

Create a list of all scaffolds containing at least a gene
awk '$3 == "CDS" {print $0}' [GFF_FILE] | awk '{print $1}' | sort | uniq > [GFF_FILE].scaffIDwithCDS

Then extract the sequences:

while read line; do python cutSeqGff.py fastafiles/$line.fasta [GFF_FILE] $line CDS; done < [GFF_FILE].scaffIDwithCDS

4/ for all CDS, compute the GC & TE content

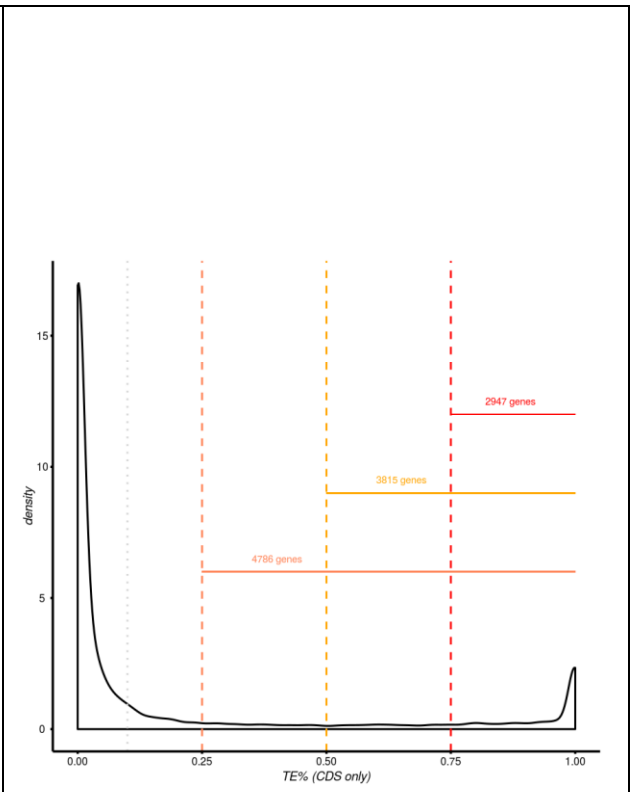
(the scripts also compute the TE content in GC, at least if a minimum of softmasked bases are present).

*for i in *; do perl GC-TEcontent_gene.pl ; done > summary_TE_GCcontent.txt*

5/ plot the output in R

Density

```
setwd("/home/thibaultleroy/Zosterops/gene_models/TE_CDS/CDS")
CDS=read.table("CDS_GC-TEcontents_sumstats.sed", h=T)
ggplot(CDS, aes(TErate))  +
  geom_density(lwd=1.05) + xlab("TE% (CDS only)")+
  theme_bw()+
  theme(panel.border = element_blank(), panel.grid.major = element_blank(), panel.grid.minor = element_blank(), axis.line = element_line(colour = "black"))+
  theme(axis.line = element_line(colour = 'black', size = 1.25), axis.ticks = element_line(colour = 'black', size = 1.25),
        axis.text.x = element_text(colour="black",size=12,angle=0,hjust=.5,vjust=.5,face="plain"),
        axis.text.y = element_text(colour="black",size=12,angle=0,hjust=.5,vjust=.5,face="plain"),
        axis.title.x = element_text(colour="black",size=14,angle=0,hjust=.5,vjust=.2,face="italic"),
        axis.title.y = element_text(colour="black",size=14,angle=90,hjust=.5,vjust=.5,face="italic"))+
  geom_vline(xintercept=0.1,col="lightgrey",lty=3,lwd=1)+
  geom_vline(xintercept=0.25,col="coral",lty=2,lwd=1)+
  geom_vline(xintercept=0.5,col="orange",lty=2,lwd=1)+
  geom_vline(xintercept=0.75,col="red",lty=2,lwd=1)+
  geom_segment(x=0.25,y=6,xend=1,yend=6,col="coral")+
  geom_segment(x=0.5,y=9,xend=1,yend=9,col="orange")+
  geom_segment(x=0.75,y=12,xend=1,yend=12,col="red")+
  annotate("text", x=0.4, y=6.5, label= "4786 genes",col="coral")+
  annotate("text", x=0.65, y=9.5, label= "3815 genes",col="orange")+
  annotate("text", x=0.87, y=12.5, label= "2947 genes",col="red")
```



Scatterplot length~TE% with loess (can takes some time to run)

Based on our discussion this morning, I just tried to add a plot to add also the length of the CDS, I only consider CDS with a length < 10 kb, the smooth can be adjusted with the span parameter.

```
ggplot(CDS)+
  geom_point(aes(x = TErate, y =ACGT),
size=0.1,color="lightblue") + ylim(0,10000)+
  geom_smooth(aes(x = TErate, y
=ACGT),method="loess", colour="black",span=0.75)+
  xlab("TE%")+ ylab("length (ACGT nucleotides only)")+
 theme(legend.key.size = unit(1.5,"line"))+
  theme_bw()+
  theme(panel.border = element_blank(),
panel.grid.major = element_blank(), panel.grid.minor =
element_blank(), axis.line = element_line(colour =
"black"))+
  theme(axis.line = element_line(colour = 'black', size =
1.25), axis.ticks = element_line(colour = 'black', size =
1.25),
        axis.text.x =
element_text(colour="black",size=12,angle=0,hjust=.5,vj
ust=.5,face="plain"),
        axis.text.y =
element_text(colour="black",size=12,angle=0,hjust=.5,vj
ust=.5,face="plain"),
        axis.title.x =
element_text(colour="black",size=14,angle=0,hjust=.5,vj
ust=.2,face="italic"),
        axis.title.y =
element_text(colour="black",size=14,angle=90,hjust=.5,
vjust=.5,face="italic"))
```