

# FEL results preliminary

sadie

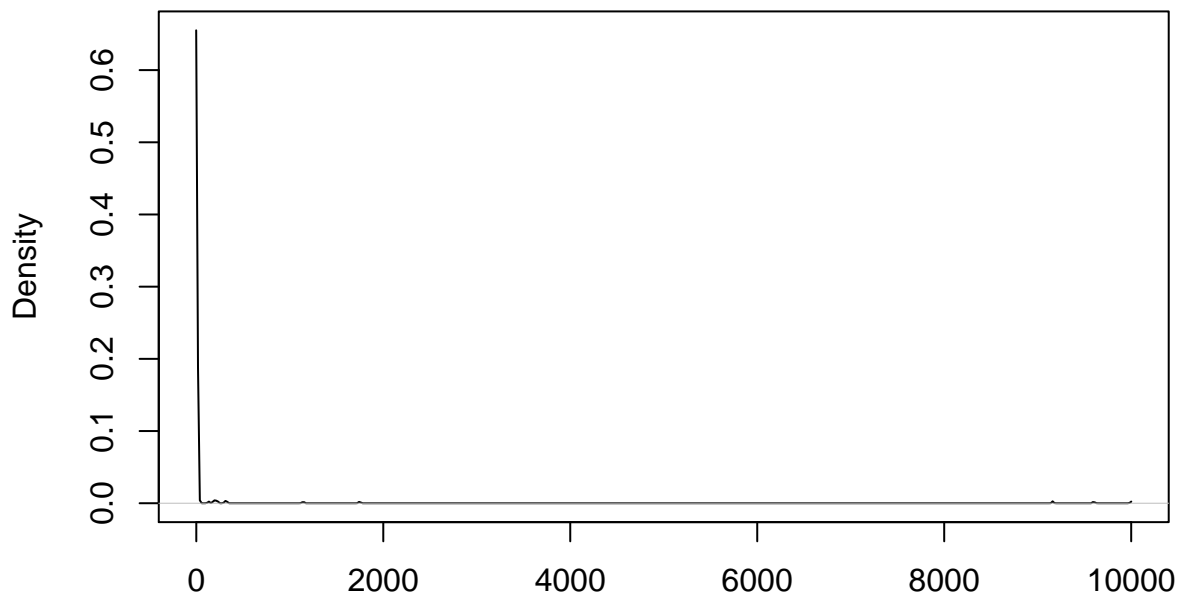
3/4/2020

```
filepath <- read_json("~/bin/mtDNA_redo/data/FEL/zeiformes-nd1-align-dna.fas.FEL.json") #read in json
heads <- filepath$MLE$headers %>% unlist() %>% .[c(TRUE,FALSE)] #get headers and ignore header descrip
#get MLE contents and make them a data frame
temp <- filepath$MLE$content$`0` %>% unlist %>% matrix(ncol = 6, byrow = TRUE) %>% as.data.frame()
#make the headers the variable names
names(temp) <- heads
```

## kernel density and plots?

```
d_alpha <- density(temp$alpha, kernel = "gaussian")
d_alpha %>% plot()
```

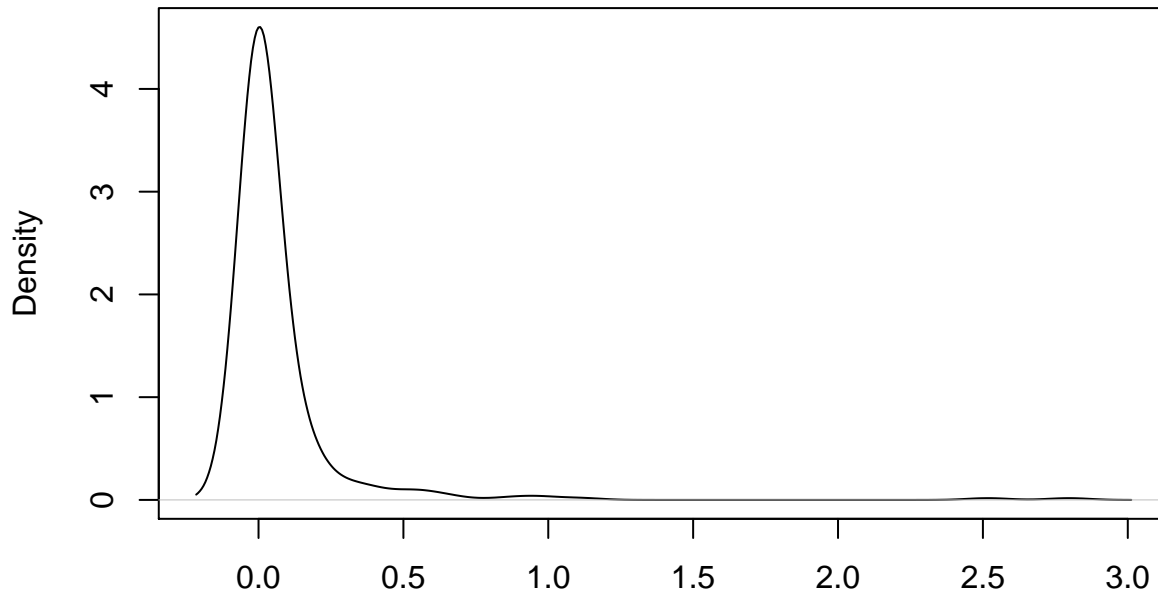
**density.default(x = temp\$alpha, kernel = "gaussian")**



N = 325 Bandwidth = 0.4233

```
d_beta <- density(temp$beta)
d_beta %>% plot()
```

## density.default(x = temp\$beta)



N = 325 Bandwidth = 0.07172

```
#uses entropy library to calculate Kullback-Leibler divergence (KL) as it is needed to do the JSD
library("entropy")
```

```
KL.plugin(freqs1 = d_alpha$x, freqs2 = d_beta$x)
```

```
## Warning in KL.plugin(freqs1 = d_alpha$x, freqs2 = d_beta$x): Vanishing value(s)
## in argument freqs2!
```

```
## Warning in log(freqs1/freqs2): NaNs produced
```

```
## [1] NaN
```

```
#actually, lets try the way this site does it: https://enterotype.embl.de/enterotypes.html
```

```
JSD<- function(x,y) sqrt(0.5 * KLD(x, (x+y)/2) + 0.5 * KLD(y, (x+y)/2))
```

```
KLD <- function(x,y) sum(x * log(x/y))
```

```
dist.JSD <- function(inMatrix, pseudocount=0.000001, ...) {
  KLD <- function(x,y) sum(x *log(x/y))
  JSD<- function(x,y) sqrt(0.5 * KLD(x, (x+y)/2) + 0.5 * KLD(y, (x+y)/2))
  matrixColSize <- length(colnames(inMatrix))
  matrixRowSize <- length(rownames(inMatrix))
  colnames <- colnames(inMatrix)
  resultsMatrix <- matrix(0, matrixColSize, matrixColSize)
```

```
inMatrix = apply(inMatrix,1:2,function(x) ifelse (x==0,pseudocount,x))
```

```
  for(i in 1:matrixColSize) {
    for(j in 1:matrixColSize) {
      resultsMatrix[i,j]=JSD(as.vector(inMatrix[,i]),
```

```

        as.vector(inMatrix[,j]))
    }
}
colnames -> colnames(resultsMatrix) -> rownames(resultsMatrix)
as.dist(resultsMatrix)->resultsMatrix
attr(resultsMatrix, "method") <- "dist"
return(resultsMatrix)
}

d.temp <- dist.JSD(temp)

d.temp

##              alpha          beta alpha=beta          LRT      p-value
## beta          108.088230
## alpha=beta    107.509626    5.177179
## LRT           106.375193    21.272359    18.908560
## p-value       108.137680    4.851234    6.477034    21.339475
## Total branch length  5.825413 103.860582 103.276212 102.195915 103.926349

```