```
##>> Script to analyse Fig. 3 using the material found in https://github.com/djhwueng/kappapcm
##>> Repository last accessed on Oct. 23, last commit ref. 4ada256.
##>> URL to the used state of the repository:
https://github.com/djhwueng/kappapcm/tree/4ada256dc4f467af2b4ca412664fae0ae520ef5e
##>> All comments with are made with the tag '##>>'


#######################
##>> REPRODUCING FIG 3
#######################


##>> Here, we first try to reproduce the results presented in Fig. 3.
##>> The code to produce this figure seem to be in `email_code.R`, from line 500
##>> The result of this analysis should then be stored in object `kappa_taxa_sim.RData`


##>> We start by loading the file
kappa_taxa_sim_con <-
url("https://github.com/djhwueng/kappapcm/raw/4ada256dc4f467af2b4ca412664fae0ae520ef5e/k
appa_taxa_sim.RData")
load(kappa_taxa_sim_con)
close(kappa_taxa_sim_con)


##>> We then plot the data using the r script
##>> l.578 - 583 from  `email_code.R`
##>> URL:
https://github.com/djhwueng/kappapcm/blob/4ada256dc4f467af2b4ca412664fae0ae520ef5e/email
_code.R#L578
plot(apply(log(kappa.array),2,median),type="b",
ylim=c(min(mean.kappa,mean.kappa2,mean.kappa1),
max(mean.kappa,mean.kappa2,mean.kappa1)), xlab="number of taxa", ylab= "log kappa",
xaxt='n', main="kappa vs. taxa (Simulated Data)" )
axis(1,at=1:length(treesize.array),treesize.array)
lines(1:length(treesize.array),mean.kappa1,type="b",pch=2, col="blue")
lines(1:length(treesize.array),mean.kappa2,type="b",pch=2, col="red")


##>> The resulting figure is indeed the one presented in Fig.3
##>> Black line is in the middle, and smoother
##>> Red line is above and blue is below, both "unstable"
##>> Below I explain some of the issues with this figure


###########################
##>> ISSUE 1: MIS-LABELLING
```

```
##########################

##>> In BLACK is plotted `apply(log(kappa.array),2,median)`
##>> It is said to be the **`rtree`** method (l.62-64 of the manuscript):
##>> "The black line was obtained from random trees using random splits trees method"
##>> "(implemented using R package: rtree (Paradis et al., 2004) along with computed BM"
##>> "(Felsenstein, 1985) to make a chronogram)"

##>> However, examining code l.517 - 533 from `email_code.R`
##>> URL:
https://github.com/djhwueng/kappapcm/blob/4ada256dc4f467af2b4ca412664fae0ae520ef5e/email
_code.R#L517
##>> we find that this simulations actually match to a **`sim.bd.taxa.age`** method.
##>> that is suposed to be the BLUE curve in Fig. 3

##>> So the conclusions of the paragraph lines 68-77 are reversed compared
##>> to what is actually simulated ans studied.

##>> The actual labelling from the script is:
##>> BLACK: `sim.bd.taxa.age`
##>> BLUE: `rtree`
##>> RED: `pbtree`

#############################################
##>> ISSUE 2: SIMULATION REPLICATION ERROR
#############################################

##>> But the RED and BLUE curve are actually misleading, even with their correct labels.

##>> Indeed, they are based on data that is not variable at all:
summary(kappa.array1)
summary(kappa.array2)

all(log(kappa.array1[1, ]) == mean.kappa1)
all(log(kappa.array2[1, ]) == mean.kappa2)

##>> So something must have gone wrong in the simulations, and the
##>> red and blue curves are actually based on just **ONE** replicate,
##>> instead of 100 for the black curve.

##>> This in itself explains the "unstability" of the red and blue curves.
```

##>> Note that this error is probably due to a mis-use of function `rep` in the simulations.

#############################
##>> ISSUE 3: MEDIAN vs MEAN
#############################

##>> The manuscript states that (l. 61)
##>> "each line represents the **median** of 100 runs".

##>> This is true for the black line, but false for the red and blue lines, where the **mean** is
plotted.
##>> Even if the simulations were correct, the scores plotted here are not the same for all the
curves.

####################################
##>> ISSUE 4: NON-ULTRAMETRIC TREES
####################################

##>> All the paper assumes that the tree considered are ultrametric.
##>> However, even if the simulations were correct, the `rtree` method **does not** produce a
ultrametric tree.
##>> So the comparison between birth-death models and split-tree method is biaised.

##>> From l.524 of `email_code.R`
##>> URL:
https://github.com/djhwueng/kappapcm/blob/4ada256dc4f467af2b4ca412664fae0ae520ef5e/email
_code.R#L524

```
tree <- ape::rtree(treesize.array[treesizeIndex])
ape::is.ultrametric(tree)
```