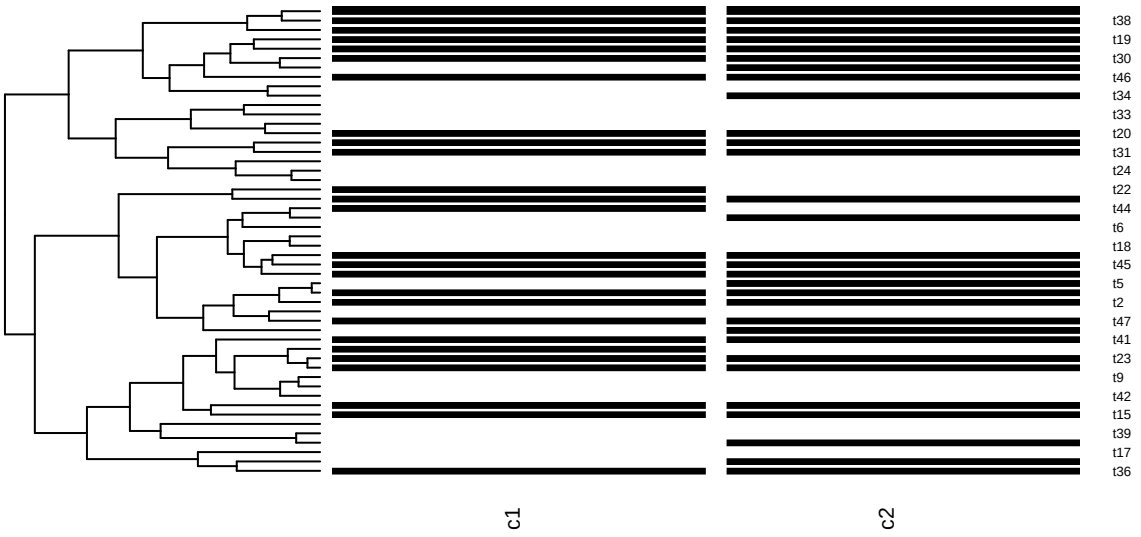
A Brief Tutorial for CCM

Simulate phylogenetic profiles

```
library(evolCCM)
library(ape)
# Generate a random tree
set.seed(123)
t <- rtree(50)
# convert the tree to dendrogram for visualization purpose
d <- TreeToDend(t)
```

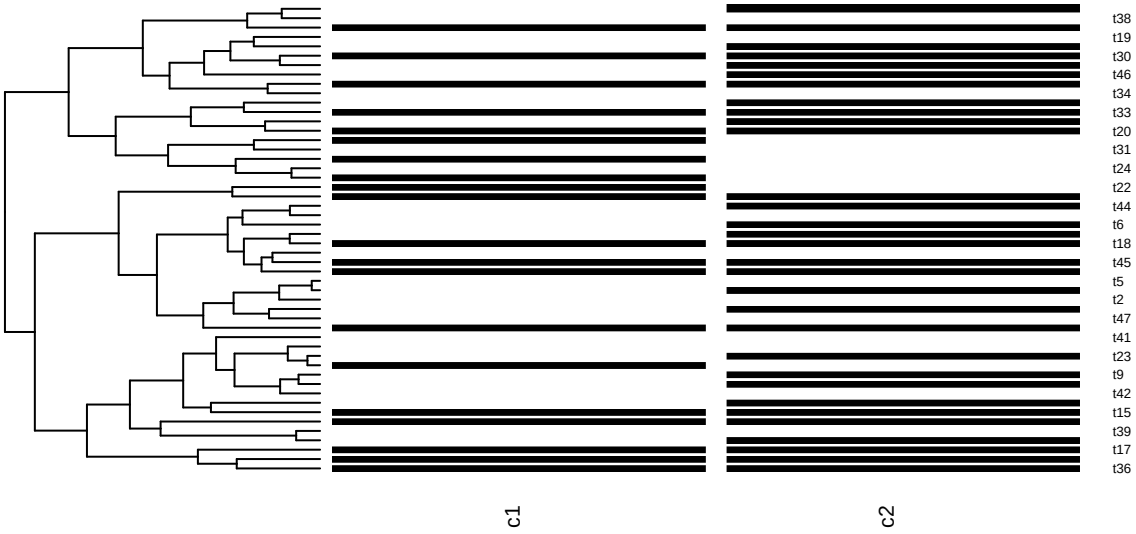
Simulate a pair of genes with interaction

```
n <- 2 # a pair of genes
alpha <- c(0.1, 0.1) # intrinsic rates
B<-matrix(0,n,n)
diag(B) <- c(-0.3,0.3) # gain / loss difference
# a pair of genes with interaction
B[1,2] <- B[2,1] <- 0.8

# simulate the profile
simDF <- SimulateProfiles(t, alpha, B)
# plot the profiles
ProfilePlot(simDF, d)
```



Simulate a pair of genes with no interaction

```
# set interaction to 0
B[1,2] <- B[2,1] <- 0
# simulate the profile
simDF <- SimulateProfiles(t, alpha, B)
# plot the profiles
ProfilePlot(simDF, d)
```



Estimate the parameters

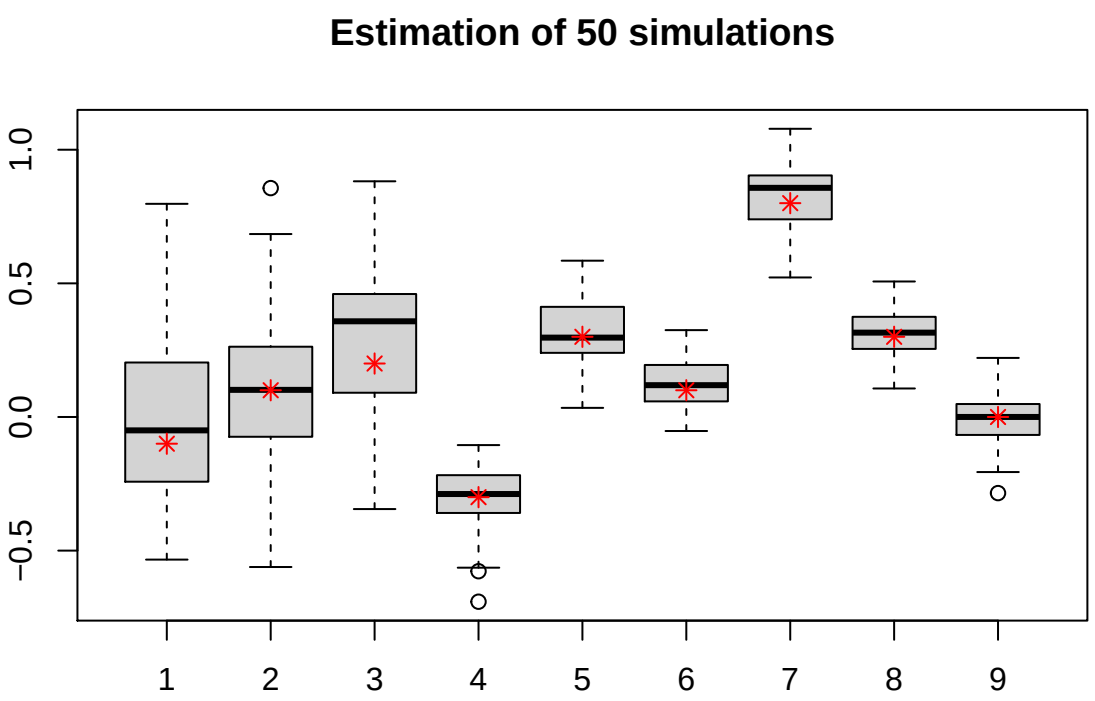
Set up parameters for a triplet with one conditionally independent link

```
# generate a random 200-tip tree
t <- rtree(200)
n <- 3 # a triplet of genes
alpha <- c(-0.1, 0.1, 0.2) # intrinsic rates
B<-matrix(0,n,n)
diag(B) <- c(-0.3,0.3,0.1) # gain / loss difference
# (1,2) and (1,3) have interactions, but (2,3) is conditionally independent
B[1,2] <- B[2,1] <- 0.8
B[1,3] <- B[3,1] <- 0.3
```

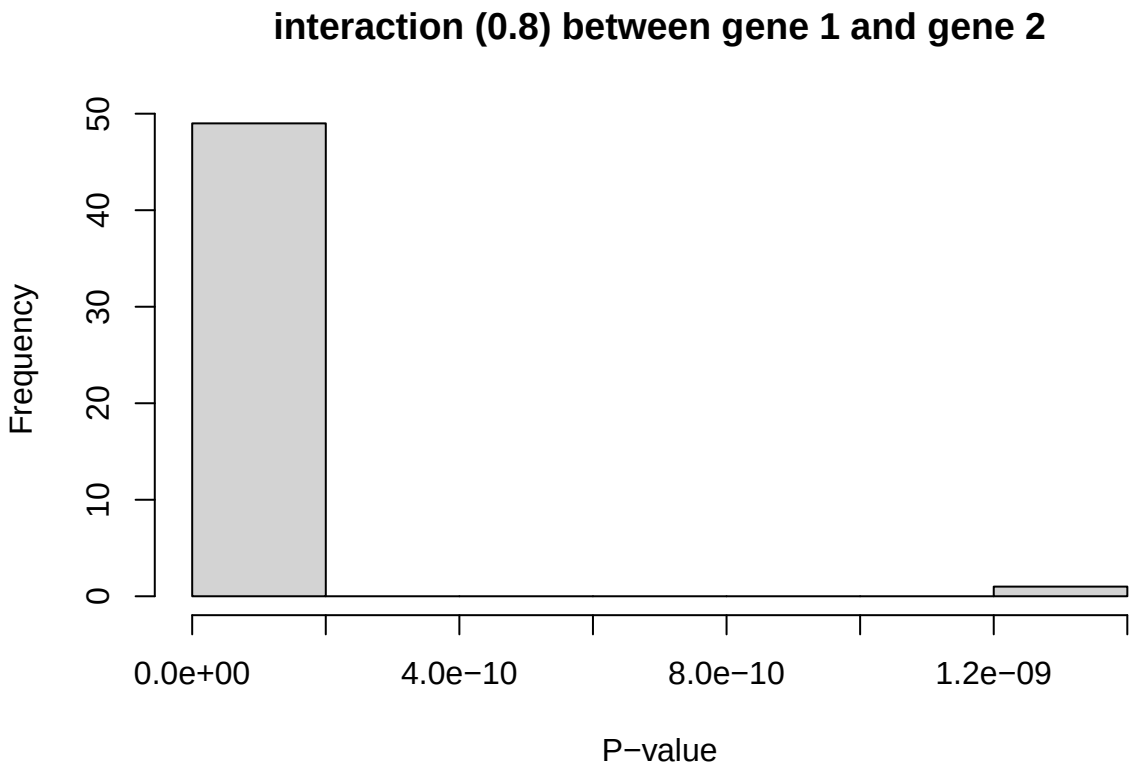
Evaluate the estimation

```
nrun <- 50 # number of simulations
trueP <- c(alpha, diag(B), B[upper.tri(B)])
estP <- matrix(NA, nrow=nrun, ncol=length(trueP))
estSE <- matrix(NA, nrow=nrun, ncol=length(trueP))
covRates <- c()
for (i in 1:nrun){
    simDF <- SimulateProfiles(t, alpha, B)
    aE <- EstimateCCM(profiles=simDF, phytree=t)
```

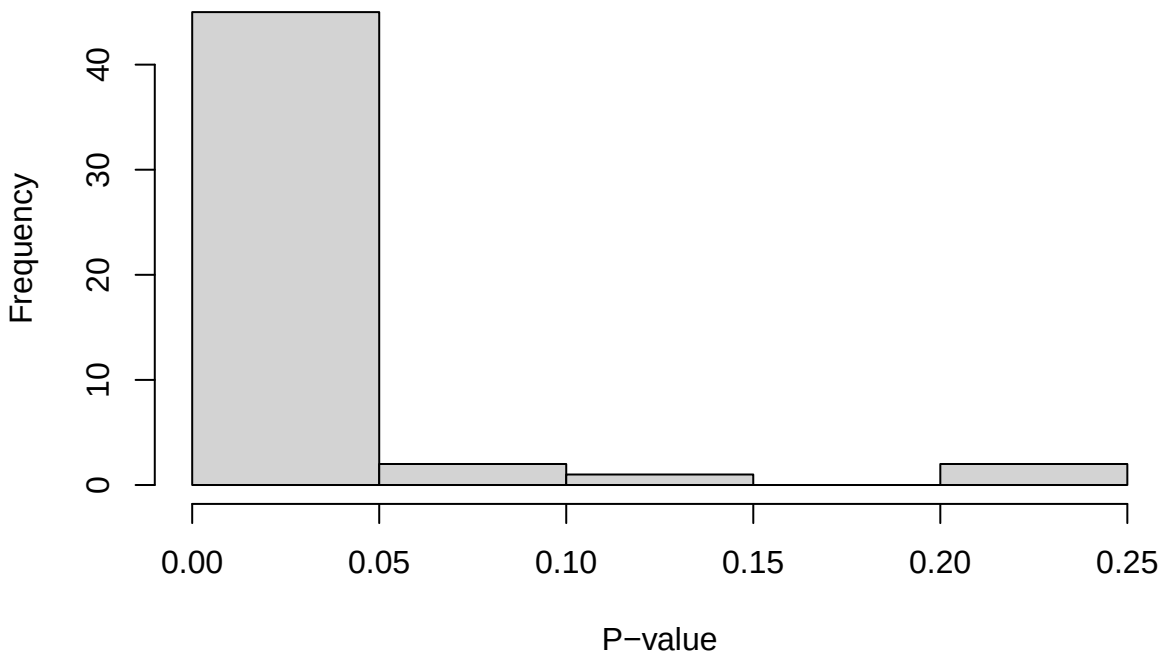
```
    estP[i,] <- c(aE$alpha, diag(aE$B), aE$B[upper.tri(aE$B)])
    paE <- ProcessAE(aE)
    estSE[i,] <- paE$hessianSE
    # negative or very large convergence rates mean not good convergence
    covRates <- c(covRates, paE$rate)
}

# plot the distribution of estimation
boxplot(estP, main=paste0("Estimation of ",nrun," simulations"))
points(1:length(trueP), trueP, pch=8, col="red")
```



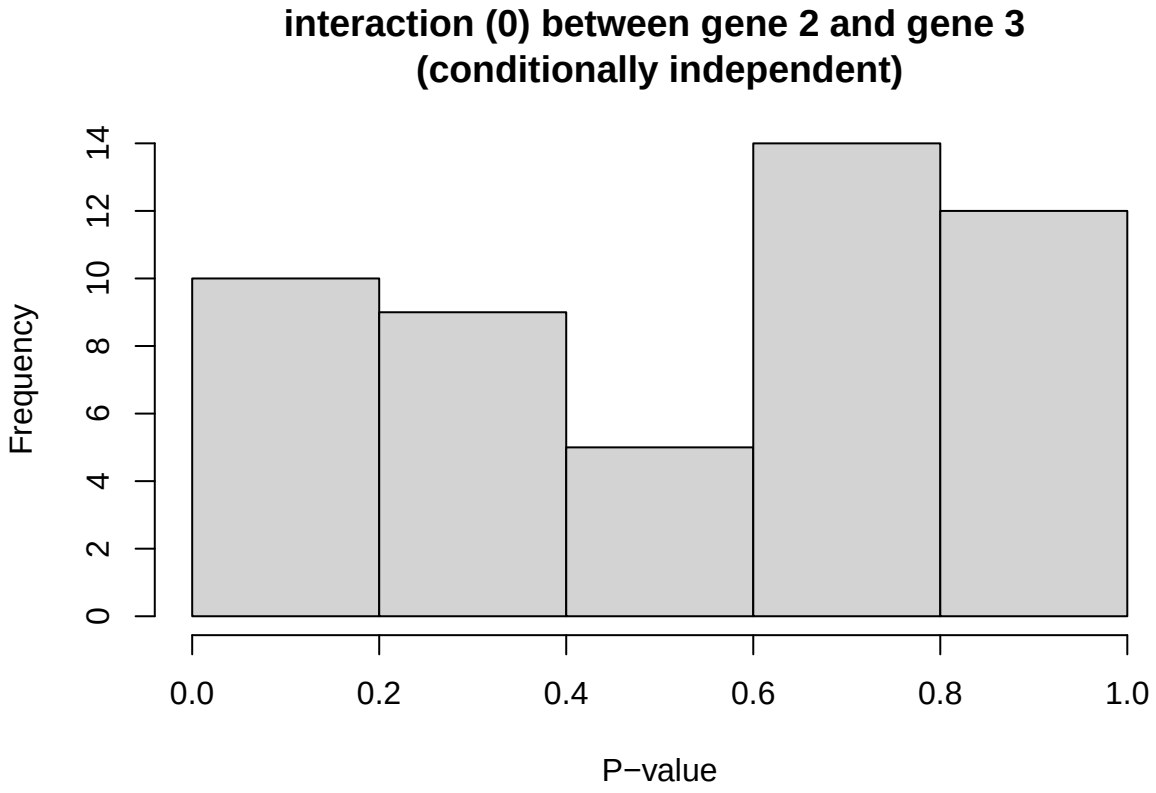

```
# distribution of p-values
hist(2*(1-pnorm(abs(estP[,7]/estSE[,7]))),xlab = "P-value",
     main="interaction (0.8) between gene 1 and gene 2")
```




```
hist(2*(1-pnorm(abs(estP[,8]/estSE[,8]))),xlab= "P-value",
     main="interaction (0.3) between gene 1 and gene 3")
```




```
hist(2*(1-pnorm(abs(estP[,9]/estSE[,9]))),xlab="P-value",
     main="interaction (0) between gene 2 and gene 3 \n(conditionally independent)")
```



Other notes

- Larger tree contains more information and tends to give better MLEs.
- Rates should be set in a reasonable scale according to the tree to avoid the simulated profiles being all 0s or 1s.
- Convergence rate in `ProcessAE()` can be used to decide whether the fittings successfully converge or not.
- Adding penalty could improve the convergence of MLE but may introduce bias into the estimations.
- To estimate a large gene community, we can first use the random initials to obtain the roughly estimated rates, which then can be used as a good set of initial values for next fitting.