January 7, 2016

Introduction

In this report we

- Describe a tree simulation
- Show results of the MLE fitting to simulated data
- Attach the R code

Data Simulation

- To generate simulated data (phylogenetic tree) we use the model described in the report of 7/12/15.
- In the pseudo-code we denote a_j as the characteristic value of the specie j. x_i is the number of species on time t_i.
- We generate the initial a_1 from an exponential distribution. After a speciation event we generate the characteristic number of the new specie randomly from the characteristic number of their antecessor (using a gamma distribution).
- We set the parameters θ and φ arbitrarily
- The R implementation is in the appendix section.

Data: N, θ, ϕ
Result: Pseudo-code of Phylogenetic tree Simulation (i.e T, X, A)
Initialization;
$a_1 = rexp(4)$;
$x_0 = 1$;
$t_0 = 0$;
$i = 2$;
$\theta = (3, 4)$;
$\varphi = (1, 2)$;
while $i < N$ *and* $x > 0$ **do**
  $\beta = e^{\theta_1 + \theta_2 a(t_i)}$;
  $\delta = e^{\varphi_1 + \varphi_2 a(t_1)}$;
  $S = \sum |\beta| + |\delta|$;
  $t_i = rexp(S)$;
  $p = (\beta, \delta)/S$;
  Sample a Birth-Death event from probability p ; from the sample we get a birth or death value, and the corresponding specie x^*. ;
  if *Death* **then**
    $x_t = x_{t-1} - 1$;
    remove characteristic number of the extincted specie x^*;
  else
    $x_t = x_{t-1} + 1$;
    $r = rgamma(1, 100, 100)$;
    $a_{newx} = r a_{x^*}$;
  end
end

Algorithm 1: Pseudo-code of the Phylogenetic tree simulation

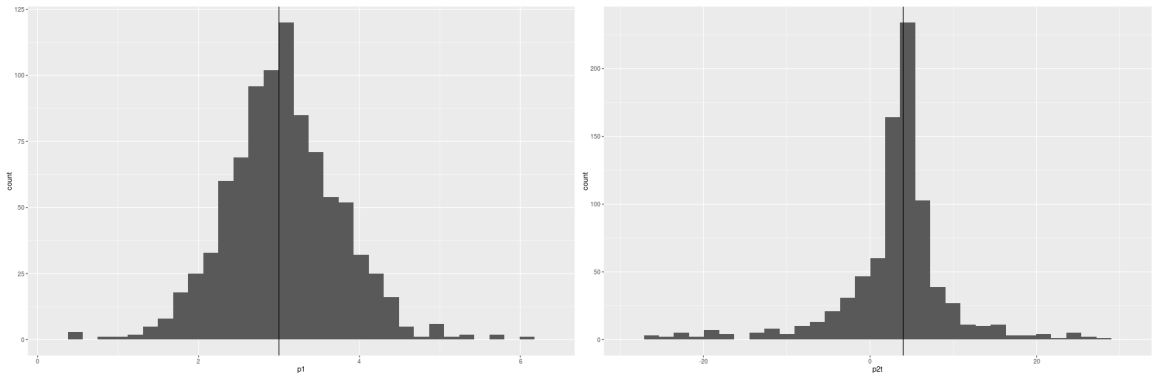
Some results

As a first experiment we simulated 896 phylogenetic trees from same parameters and we performed the MLE procedure described in the report of 7/12/15.

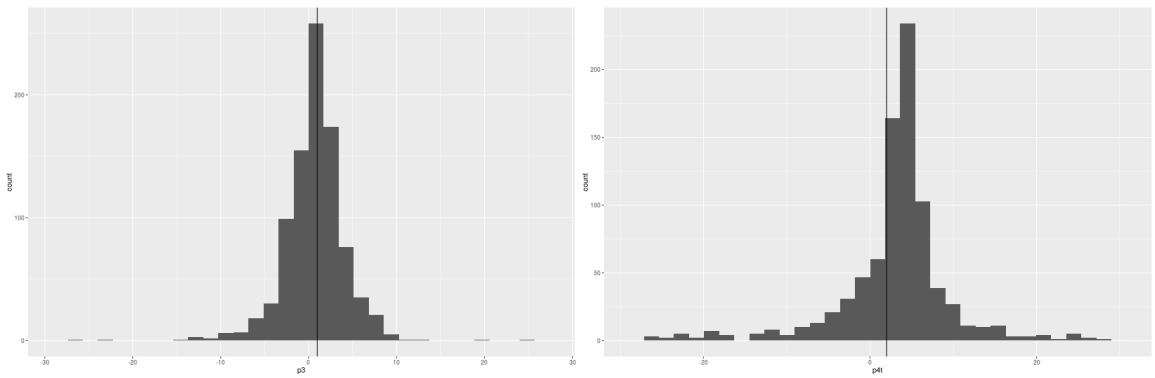
The table bellow shows the mean, median, max and min of the estimated values. Note that the mean of the estimations does not gives a good estimation of the real values, specially for parameters θ_2 and φ_2, which have high variance (we can have an idea from the min/max) and extreme values. The median, however, gives a much better estimation of the real parameters.

n = 896	real value	mean	median	min	max
θ_1	3.00	3.05	3.03	0.39	6.00
θ_2	4.00	0.61	3.85	-4136.88	949.77
ϕ_1	1.00	0.68	0.81	-26.92	24.46
ϕ_2	2.00	5.00	1.80	-2712.26	2164.94

Moreover, plotting the histograms of the estimations we can see bellow some symetry in the estimations; the black vertical line corresponds to the real value. This histrograms shows us that the estimations are actually around the real values, the lack of precision on the mean is due to the outliers.



(a) θ_1 (b) θ_2



(c) φ_1 (d) φ_2

Figure 1: Histograms of the estimated values. The black vertical line corresponds to the real values.

Apendix

code

```
nP = 1000000
P <- matrix(nrow=nP,ncol=4)

for(k in 1:nP){
  nT = 100
  theta = c(3,4)
  phi = c(1,2)
  #set.seed(123)
  dat<-NULL
  A=rexp(1,4)
  x=1
  dat[[1]]<-list(tm=0,x=x,A=A)
  X=x
  Tm=0
  i<-2
  a=A
  stm <- 0
  E=1
  while (i<=nT){
    empty = FALSE
    beta<-exp(theta[1]+theta[2]*dat[[i-1]]$A)
    delta<-exp(phi[1]+phi[2]*dat[[i-1]]$A)
    prev.x=dat[[i-1]]$x
    tm<-rexp(1,(sum(beta)+sum(delta)))
    stm <- stm+tm
    prob<-c(delta,beta)/(sum(beta)+sum(delta))
    BD<-sample(2*prev.x,1,prob=prob)
    if(BD<=prev.x){
      x = prev.x-1
      if (x==0){
        empty = TRUE
      break
      }
      a[i] = dat[[i-1]]$A[BD]
      A = dat[[i-1]]$A[-BD]
      E[i] = 0
    }
    else {
      x = prev.x + 1
      a[i] = dat[[i-1]]$A[BD-prev.x]
      A = c(dat[[i-1]]$A,dat[[i-1]]$A[BD-prev.x]*rgamma(1,100,100))
      E[i] = 1
    }
    dat[[i]]<-list(tm=stm,x=x,A=A)
    X[i]=x
    Tm[i]=tm
    i<-i+1
    }
  if (!empty){
    fn <- function(theta) {
    beta = 0
    delta = 0
    parm = (theta[1] + theta[2]*a)*E+(theta[3] + theta[4]*a)*(1-E)
    sig = 0
    for (i in 1:nT){
      sig[i] = sum(exp(theta[1] + theta[2]*dat[[i]]$A) + exp(theta[3] + theta[4]*dat[[i]]$A))
    }
    fn=sum(-sig*Tm+parm)
    -fn
    }
    op <- nlm(fn, theta <- c(1,1,1,1), hessian=TRUE)
    P[k,1]=op$estimate[1]
```

```
    P[k,2]=op$estimate[2]
    P[k,3]=op$estimate[3]
    P[k,4]=op$estimate[4]
    B[k]=length(E[E==1])
    D[k]=length(E[E==0])

    }
}
```