Methodology on the EMPHASIS framework

1. Introduction

In this document we describe the methodology behind the EMPHASIS (Expectation-Maximizatioin on PHylogenetic Analysis with Simulations and Importance Sampling) framework. Additionally, we include the code to compute every step for the diversity dependance case, contained in the emphasis R package.

The main goal of EMPHASIS is the integration of phylogenetic and ecological information in a comprehensive framework capable to include a broad type of models in a flexible way. This is a challenging goal for several reasons including incompleteness of data sources (especialy on phylogenetic trees) and complexity of ecological interactions.

We approach this problem by applying an EM-type algorithm, which defines the EM iteration $\phi^* \to \phi$ as

E-step Compute $Q(\phi|\phi^*) = E_{\phi^*}(\log f(x|\phi)|y)$,

M-step Choose ϕ to be the value of $\phi \in \Omega$ which maximizes $Q(\phi|\phi^*)$

Starting from an initial value $\phi \in \Omega \subset \mathbb{R}^m$ we perform these two steps iteratively until reaching convergence on the parameter space Ω. In the following sections we explain how we implement it in each step of the algorithm.

Notation

T is the total time from crown time to present. We call a *missing species* (or an extinct species), those branches on the phylogenetic tree which their tip is not at time T.

We define $\mu_{t,j}$ and $\lambda_{t,j}$ as the extinction and speciation rates of species j at time $t \in (0,T)$ and $\lambda_t = \sum_{j=1}^{n_t} \lambda_{t,j}$. n_t is the number of extant species at time t, m_t the number of missing species at time t, and we will be especially interested in the branching points of the phylogenetic tree $\{t_1, t_2, ..., t_d\}$ and their corresponding waiting times $\{\Delta t_1, ..., \Delta t_d\}$. We also define the sum of rates at a specific moment t_i as

$$\sigma_i = \sum_{j=1}^{n_{t_i}} \lambda_{t_i,j} + \mu_{t_i,j}$$

and the the sum of speciation rates at time t_i as [1]

$$s_i = \sum_{j=1}^{n_{t_i}} \lambda_{t_i,j}$$

Moreover, we define the topology component of a tree at time t_i as the vector $\tau_i \in \{0,1\}^{2*n_{t_i}}$. So, this vector contains a 1 in the position j if species j has speciated at time t_i, or a 1 in position $(n_{t_i}+j)$ if species j becames extinct at time t_i, and 0 elsewhere. We finally define

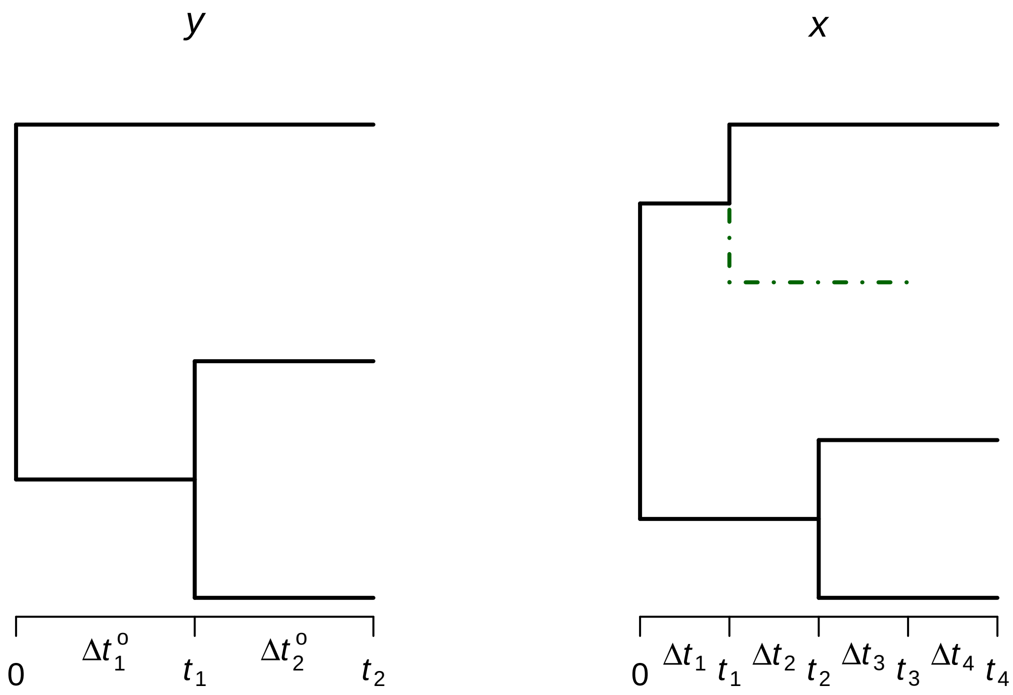
$$\rho_i = \prod_{j=1}^{n_{t_i}} (\lambda_{(t_i,j)})^{\tau_i^{(j)}} \prod_{j=1}^{n_{t_i}} (\mu_{t_i,j})^{\tau_i^{(n_{t_i}+j)}}$$

where $\tau_i^{(j)}$ is the j-component of the vector τ_i. Note that ρ_i is just the speciation or extinction rate of species diversifying at branching time t_i. As we assume that only one speciation or extinction take place at a

[1] Note that $s_i = \lambda_{t_i}$, they are just continuos/discrete caracterizations of the same process.

moment t [2], the vector τ_i in principle contains only one 1 and 0??s elsewhere and we use the notation $\mathbb{1}_j$ for a trans-dimensional vector with a 1 at position j and zeros elsewhere.

Hereafter we refer to $x \in \mathcal{X}$ as a variable representing a complete tree (extinct species included) and $y \in \mathcal{Y}$ as variable representing the observed (ultrametric) tree [3], moreover, $\mathcal{X}(y)$ is the subset of complete trees that, pruning extinctions, is equivalent to the ultrametric tree y. For a detailed explanation of this time-tree spaces we refer to [?]. On the chart bellow we see an example of $y \in \mathcal{Y}$ and $x \in \mathcal{X}(y)$.



2. E-step

On the E-step we want to calculate

$$Q(\phi|\phi^*) = E_{\phi^*}(\log f_X(x;\phi)|y) = \int_{\mathcal{X}(y)} \log f_X(x;\phi) f_{X|Y}(x|y,\phi^*)dx$$

where $f_X(x;\phi)|y)$ is the probability density of a complete tree defined as

$$f_X(x;\phi) = \left(\prod_{i=1}^{d} e^{-\sigma_i \Delta t_i}\right)\left(\prod_{i=1}^{d-1} \rho_i\right) \tag{1}$$

and $f_{X|Y}(x|y,\phi)$ is the probability density of the complete tree x given that $x \in \mathcal{X}(y)$. Given the complexity of the space $\mathcal{X}(y)$, $Q(\phi|\phi^*)$ does not holds a close form, then a numerical approximation is needed.

[2]this can be generalized to multiple speciations/extinctions. For potential further projects/extenctions/generalizations please see the appendix.

[3]Mathematically, a phylogenetic tree is a combination of a vector containing the branching times on the continous space, and a vector with the topology values on a discrete space.

2.1 Monte-Carlo approximation and data augmentation algorithm

One way to perform this task is considering a Monte-Carlo sampling method, where, given a set of sampled trees $x_1, ..., x_m$ from $f_{X|Y}(x|y, \phi^*)$, we approximate $Q(\phi|\phi^*)$ by

$$E_{\phi^*}(\log f(x;\phi)|y) \approx \frac{1}{m}\sum_{i=1}^{m} \log f_X(x_i;\phi) \tag{2}$$

We can, theoretically, make this approximation as accurate as we want by increasing the sample size p if we know the exact sampling probability $f_{X|Y}(x|y, \phi^*)$, however this distribution does not hold a close form on the general case. Thus, an approximated data augmentation algorithm, and general approach incorporating importance sampling method is needed.

2.2 Probability of a missing species to occur

In order to simulate the extinct part of a tree when the extant species and their diversification events are observed, we consider the random variable:

"$\Delta t_{M_j}^{t_0}$: *Waiting time, starting at* t_0*, for branch* j *to speciate into a new species that is going to get extinct before present.*"

Thus, the r.v $\Delta t_{M_j}^{t_i}$ is exponentially distributed with a rate of $\lambda_{t_i+t,j}\left(1-e^{-\int_{t_i+t}^{T}\mu_{s,j}ds}\right)$, i.e. the probability distribution of this process would be

$$f_{M_j}^{t_i}(\Delta t_{M_j} = t) = \lambda_{t_i+t,j}\left(1-e^{-\int_{t_i+t}^{T}\mu_{s,j}ds}\right) e^{-\int_{t_i}^{t}\lambda_{z,j}\left(1-e^{-\int_{z}^{T}\mu_{s,j}ds}\right)dz} \tag{3}$$

Moreover we define $\Delta t_M^{t_i} = \min\{\Delta t_{M_1}^{t_i}, ..., \Delta t_{M_n}^{t_i}\}$, then

$$f_M^{t_i}(\Delta t_M^{t_i} = t) = \left(\sum_{j=1}^{n_t}\lambda_{t_i+t,j}\left(1-e^{-\int_{t_i+t}^{T}\mu_{s,j}ds}\right)\right) e^{-\int_{t_i}^{t_i+t}\sum_{j=1}^{n_t}\lambda_{z,j}\left(1-e^{-\int_{z}^{T}\mu_{s,j}ds}\right)dz} \tag{4}$$

This equation is simple in many particular scenarios. One case is when all species have same rates at any time t:

$$\lambda_{t,j} = \lambda_{t,0}; \quad \mu_{t,j} = \mu_{t,0}, \quad \forall j = 1, ..., n_t \tag{5}$$

in that case equation 4 would be equivalent to

$$f_M^{t_i}(\Delta t_M^{t_i} = t) = n_{t_i+t}\lambda_{t_i+t,0}\left(1-e^{-\int_{t_i+t}^{T}\mu_{s,0}ds}\right) e^{-\int_{t_i}^{t}n_t\lambda_{z,0}\left(1-e^{-\int_{z}^{T}\mu_{s,0}ds}\right)dz}. \tag{6}$$

We will hereafter focus on this case. We calculate equation 4 in three other particular cases in appendix A.1.

On the other hand, note that the probability density for a species j to be the next (missing) speciation after t_i is

$$f(\Delta t_M^{t_i} = t, \tau_i = \mathbb{1}_j) = f_M^{t_i}(\Delta t_M^{t_i} = t) f_\tau(\tau_i = \mathbb{1}_j | \Delta t_M^{t_i} = t) \tag{7}$$

where

$$f_\tau(\tau_i = \mathbb{1}_j | \Delta t_M^{t_i} = t) = \frac{\lambda_{t_{i+1},j}}{\sum_{k=1}^{n_t} \lambda_{t_{i+1},k}} \tag{8}$$

Finally, in order to find the probability distribution for the missing species to occurs, we also need the probability of the extinction time of each of those missing species considering the random variable

"t_{E_i}: *Time for missing species born at time t_i to become extinct.*"

By definition $t_{E_i} < T$, then the probablity distribution would be

$$f_{E_i}(t_{E_i} = t) = f(t_{E_i} = t | t < T, \Delta t_M^{t_i} = t_s, \tau_i = \mathbb{1}_j) = \frac{\mu_{t,j} e^{-\int_{t_i+t_s}^{t} \mu_{z,j} dz}}{(1 - e^{-\int_{t_i+t_s}^{T} \mu_{t,j} dt})}. \tag{9}$$

With those results we are ready to calculate the probability distribution of a missing species. Starting at time t_i, we consider the next missing species mathematicaly defined as the triad $M^{t_i} = (\Delta t_M^{t_i}, \tau_i, t_{E_i})$, and we can calculate its probability distribution using equations 4, 9 and 7

$$\begin{aligned} f_D^{t_i}(M^{t_i} = (t_s, \mathbb{1}_k, t_e)) =& f_M^{t_i}(\Delta t_M^{t_i} = t_s) f_\tau(\tau_i = \mathbb{1}_k | \Delta t_M^{t_i} = t_s) f(t_{E_i} = t_e | t < T, \Delta t_M^{t_0} = t_s, \tau_i = \mathbb{1}_j) \\ =& \lambda_{t_s,k} \mu_{t_e,k} e^{-\int_{t_i}^{t_i+t_s} n_t \lambda_{z,k} \left(1 - e^{-\int_z^T \mu_{s,k} ds}\right) dz - \int_{t_i+t_s}^{t_e} \mu_{z,k} dz} \end{aligned} \tag{10}$$

2.3 Data augmentation algorithm

The data augmentation algorithm will be:

Input: Set of waiting times $\Delta t_1^o, ..., \Delta t_d^o$ of the observed tree.

1. set $i = 1, t = 0$ and the set of extinctions $E = \emptyset$
2. $\Delta t = \Delta t^o$. If $i > d$ go to step 8
3. $t \leftarrow t + \Delta t$
4. Draw the triad $M^t = (\Delta t_M^t, \tau, t_E)$ from f_D^t (See A.2 for details)
5. Draw time $\Delta t_e = \min\{E\} - t$ of next extinction
6. Check what is the next event
   - If $\Delta t < \Delta t_e$ and $\Delta t < \Delta t_M^t$: set $i \leftarrow i + 1$ and go to step 2
   - If $\Delta t_M^t < \Delta t_e$ and $\Delta t_M^t < \Delta t$: add new speciation $t + \Delta t_M^t$ to the tree, add new extinction $t_E + t + \Delta t_M^t$ to E, set $\Delta t \leftarrow \Delta t - \Delta t_M^t$, and go to step 3
   - If $\Delta t_M^t > \Delta t_e$ and $\Delta t_e < \Delta t$: add new extinction $t_E + t$ to the tree, remove extinction from E, set $\Delta t \leftarrow \Delta t - t_e$, and go to step 3
7. Return the complete tree.

Once we get the missing part of the tree, we neet to calculate the importance weights by using the sampling probability and the real probability of the whole tree. We do that on next section.

2.4 Importance Sampling

Due to the complexity of the problem we cannot sample from $f_{X|Y}(x|y, \phi^*)$, but we can still sample from an approximated distribution $g_{X|Y}(x|y, \phi^*)$ (using the data argumentation algorithm described on previous section), and then correct via importance sampling re-writting equation (2) in the general version

$$E_{\phi^*}(\log f(x;\phi)|y) \approx \sum_{i=1}^{p} \log f_{X;\phi^*}(x_i;\phi) \frac{f_{X|Y}(x_i|y,\phi^*)}{g_{X|Y}(x_i|y,\phi^*)}$$

Then we must find an expression for $f_{X|Y}(x_i|y, \phi^*)$. We can write this as

$$f_{X|Y}(x_i|y,\phi^*) = \frac{f_{X,Y}(x_i, y|\phi^*)}{f_Y(y|\phi^*)}$$

using the law of conditional probabilities. Because the denominator is the same for all x_i and does not depend on ϕ it will not affect our maximization step, and we can simply write

$$Q(\phi|\phi^*) = E_{\phi^*}(\log f(x;\phi)|y) \approx \frac{1}{f_Y(y|\phi^*)} \sum_{i=1}^{m} \log f_{X;\phi^*}(x_i;\phi) \frac{f_{X,Y}(x_i, y|\phi^*)}{g_{X|Y}(x_i|y,\phi^*)}$$

where $\frac{1}{f_Y(y|\phi^*)}$ is just a (unknown) constant value. . Note that the dependence on ϕ (which is important for the maximization step) only occurs in the term $\log f_{X;\phi^*}(x_i;\phi)$.

To compute $g_{X|Y}(x_i|y, \phi^*)$ we use the nonhomogeneous Poisson process characterized by equation (10).

Then

$$g_{X|Y}(x_i|y,\phi^*) = \prod_{i=1}^{d} e^{-\int_{t_i}^{t_i+t_s} n_t \lambda_{z,k} \left(1-e^{-\int_z^T \mu_{s,k} ds}\right) dz} \prod_{j=1}^{m} \lambda_{t_j+t_s,j} \mu_{t_j+t_s,j} e^{-\int_{t_j+t_s}^{t_e} \mu_{t,j} dt} \tag{11}$$

We call

$$w_i = \frac{f_{X,Y}(x_i, y|\phi^*)}{g_{X|Y}(x_i|y,\phi^*)}$$

the *importance weights.*

So,

$$Q(\phi|\phi^*) \propto \sum_{i=1}^{m} w_i \log f_{X;\phi^*}(x_i;\phi) \tag{12}$$

defines completely the E-step.

3. M-step

The M-step consists on the optimization procedure

$$\phi^* = \underset{\phi\in\mathbb{R}^n}{\text{argmax}} Q(\phi|\phi^*)$$

4. Standard errors and confidence intervals

The standard errors for the MCEM algorithm is defined as

$$SD(\hat{\beta}_i) = \sqrt{\frac{-H_{i,i}^{-1}}{N} + VMCE}$$

Where $-H_{i,i}^{-1}$ corresponds to the diagonal components of the information matrix, N is the MC sample size, and $VMCE$ is the variance of the MC error.

Thus, the confidence intervals for the MCEM estimations would be defined as

$$[\hat{\beta} - 1.96 * SD(\hat{\beta}_i), \hat{\beta} + 1.96 * SD(\hat{\beta}_i)]$$

5. Stopping criteria and sample size

For the stopping criteria and the sample size needed to ensure convergence we use the procedure described on Chan et. al. The idea behind this is, considering the differences in likelihoods from one MCEM iteration to its previous one

$$\Delta l_Y(\phi_j, \phi_{j+1}) = log(f_Y(y|\phi_{j+1})) - log(f_Y(y|\phi_j))$$

because of Chebychev??s inequality, $\Delta l_Y(\phi_j, \phi_{j+1})$ will lie on the interval $(\mu - L\sigma, \mu + L\sigma)$ with probability no less than $100(1-1/L^2)\%$, where L is an integer and μ, σ are the mean and standard deviation of $\Delta l_Y(\phi_j, \phi_{j+1})$. As noticed before, that quantity is in most case not possible to calculate, but can be estimated by

$$\tilde{\Delta} l_Y(\phi_j, \phi_{j+1}) = -log\left(\left(\sum_{k=1}^{m} \frac{f_{X|Y}(x_i|y, \phi_k)}{f_{X|Y}(x_i|y, \phi)}\right)/m\right) \tag{13}$$

with ϕ being the true parameter. We use that expression for the stopping criteria.

To determine the appropriate MC sample size the following pilot study is suggested:

Imput: Set initial parameter θ_0 and natural number l corresponding to number of preliminary iterations of the EM routine. Set m_1 corresponding to the preliminary sample size on each MC routine.

S1: Do l MCEM iterations starting from θ_{init}, using Monte Carlo sample size m_1, and save the last 10 parameters $\theta_{l-10}, ..., \theta_l$ obtained by the last 10 MCEM iterations.

S2: Use each $\theta_{l-10}, ..., \theta_l$ as an initial value of an 1-iteration MCEM algorithm and save the new $\theta_{l-10}^*, ..., \theta_l^*$ and for each of those calculate the relative likelihood $R_i = \tilde{\Delta} l_Y(\theta_i^*, \theta_i)$ given by equation 13.

S3: Let $\tilde{\Delta} l_{ij}$ be the jth replicate of $\tilde{\Delta} l_y(\phi_j, \phi_{j+1})$ and compute the pooled variance

$$s^2 = \frac{1}{(l-1)(K_2 - K_1 + 1)} \sum_{i=K_1}^{K_2} \sum_{j}^{l} (\tilde{\Delta} l_{ij} - \tilde{\Delta} l_i)^2,$$

where $\tilde{\Delta} l_i$ is the average of the replicates of $\tilde{\Delta} l_Y(\phi.\phi)$.

S4: Take the suggested sample size m as any value satisfying

$$m \geq \frac{m_1 s}{\epsilon}$$

Moreover, on the MCEM routine, the iterations are stopped at the Kth iterate when $\Delta l_Y(\phi_K, \phi_{K+1})$ differs from zero by less than $2L\sigma$, where

$$\sigma \approx \frac{m_1 s}{m}$$

6. The EMPHASIS framework

The whole framework follows the following steps:

Input: Observed phylogeny y, a tolerance value $\epsilon > 0$ and a preliminary initial value ϕ_{init}

S1: Run pilot study suggested of section 5 and get sampling size m, initial values ϕ, and a tolerance value γ

S2: Sample $x_1, ..., x_m$ using algorithm 2.1.2 and calculate

$$Q(\phi|\phi^*) \propto \sum_{i=1}^{m} w_i \log f_{X;\phi^*}(x_i;\phi)$$

S3: Calculate

$$\phi^* = \underset{\phi \in \mathbb{R}^n}{\text{argmax}} Q(\phi|\phi^*)$$

and $\tilde{\Delta} l_Y(\phi, \phi^*)$ from equation 13

S4: If $|\tilde{\Delta} l_Y(\phi, \phi^*)| \geq \gamma$ set $\phi \leftarrow \phi^*$ and go to step S.2, else go to step S.5

S.5 Return ϕ^*

7. The EMPHASIS Package

The first example we explore is the diversity dependance case. That process define diversification rates as

$$\begin{aligned} \lambda_{t,j} &= \lambda_0 - (\lambda_0 - \mu_0)\frac{n_t}{K}, \\ \mu_{t,j} &= \mu_0 \end{aligned}$$

Then $\sigma_i = (\lambda_0 - (\lambda_0 - \mu_0)\frac{n_t}{K} + \mu_0)n_t$ and $\rho_i = (\lambda_0 - (\lambda_0 - \mu_0)\frac{n_t}{K})\tau_i + \mu_0(1 - \tau_i)$, where in this case the topology value is simplified as a scalar equal to 0 if there was an extinction at time t_i and 1 if there was an speciation at time t_i. That is because all species have same rates at time t.

Below is the code for the negative loglikelihood function

```
#negative logLikelihood of a tree
nllik.tree <- function(pars,tree){
  wt = tree$wt
  to = tree$E
  n = c(2,2+cumsum(to)+cumsum(to-1))
  lambda = (pars[1]-(pars[1]-pars[2])*(n/pars[3]))
  mu = pars[2]
  sigma = (lambda + mu)*n
  rho = pmax(lambda[-length(lambda)]*to+mu*(1-to),0)
  nl = -(sum(-sigma*wt)+sum(log(rho)))
  if(min(pars)<0){nl = Inf}
  return(nl)
}
```

with that, we can define inmediatelly the Q function of equation (12)

```
Q.approx = function(pars, st){
  m = length(st$rec)
  l = vector(mode = 'numeric',length = m)
  w = vector(mode = 'numeric',length = m)
  for(i in 1:m){
    s = st$rec[[i]] # complete tree
    w[i] = st$w[i] # corresponding weight
    if(w[i]!=0){ # if weight is non-zero, calculate likelihood
      l[i] = nllik.tree(pars,tree=s)
    }else{
      l[i] = 0
    }
  }
  w = w/sum(w) #normalization of weights
  L = sum(l*w)
  return(L)
}
```

The next step is to write down the data augmentation algorithm, so, given an observed (ultrametric) tree, simulate the extincted part:

```
###  simulation of extincted new version
sim.extinct <- function(brts,pars,model='dd',seed=0){
  if(seed>0) set.seed(seed)
  wt = -diff(c(brts,0))
  ct = sum(wt)
  lambda0 = pars[1]
  mu0 = pars[2]
  K = pars[3]
  dim = length(wt)
  ms = NULL # missing speciations, for now we just add time. When we consider topology we do it with sp
  me = NULL # missing extinctions (in the uniform plane)
  bt = NULL
  bte = NULL
  to = NULL
  cbt = 0
  N = 2
  h = 1 # index to fill probabilities
  for(i in 1:dim){
    cwt = wt[i]
    cbt = sum(wt[0:(i-1)])
    key = 0
    while(key == 0){
      if(model == "dd"){  # diversity-dependence model
        lambda = max(1e-99, lambda0 - (lambda0-mu0)*N/K)
        mu = mu0
        s = N*lambda
      }else{print('Model not implemented yet, try dd')}
      t.spe = rexp(1,s)
      t.ext = extinction.processes(u=me,inits=ms,mu0=mu0)
      t_ext = ifelse(length(t.ext)>0,min(t.ext),Inf)-cbt  # if is not empty gives the waiting time for 
      mint = min(t.spe,t_ext)
      if(mint < cwt){
        if(mint == t.spe){#speciation
          u = runif(1)
          if(u < pexp(ct-(cbt+t.spe),mu)){
            ms = c(ms,cbt+t.spe)
            me = c(me,u)
            bt = c(bt,cbt+t.spe) #SHOULD INCLUDE GOSTTIME NO?
            text = extinction.processes(u=u,inits=cbt+t.spe,mu0=mu0)
            bte = c(bte,text)
            to = c(to,1)
            tspe = cbt+t.spe
            N = N + 1
          }
          cwt = cwt - t.spe
          cbt = cbt + t.spe
        }
        else{#extinction
          extinctone = which(t.ext == min(t.ext))
          tspe = ms[extinctone]
          text = t.ext[extinctone]
          bt = c(bt,text)
          bte = c(bte,Inf)
          to = c(to,0)
          ms = ms[-extinctone]
          me = me[-extinctone]
          cwt = cwt - mint
          cbt = cbt + mint
          N = N-1
        }
      }
      else{
        key = 1
      }
    }
    N = N+1
  }
  df = data.frame(bt = c(bt,ct-brts),bte = c(bte, rep(Inf,length(wt))),to = c(to,rep(2,length(wt))))
  df = df[order(df$bt),]
  df$t.ext = df$bte-df$bt
  df = df[-1,]
  df = rbind(df,data.frame(bt=ct,bte=Inf,to=0,t.ext=Inf))
  return(df)
}

extinction.processes <- function(u,inits,mu0){
  nm = length(u)
  t.ext = vector(mode='numeric',length=nm)
  if(nm > 0){
    for(i in 1:nm){
      t.ext[i] = inits[i] - log(1-u[i])/mu0  #Inverse of the intensity function for constant extinction
    }
  }
  return(t.ext)
}
```

given that in this case speciation rates are piece-wise contant, in every waiting time, we can write equation (4) as

$$P_M(t) = s_\lambda(1-e^{-\mu(r-t.)})e^{-\int_0^t s_\lambda(1-e^{-\mu(r-z)})dz} = s_\lambda(1-e^{-\mu(r-t)})e^{-s_\lambda\left[t+\frac{1}{\mu}(e^{-\mu r}-e^{-\mu(r-t)})\right]}, t<r \tag{14}$$

then, equation (11) would be equivalent to

$$g_{X|Y}(x|y,\phi) = \prod_{i=1}^{d}\left(\lambda_i\mu_0 e^{-s_i\left[\Delta t_i+\frac{1}{\mu_0}(e^{-\mu_0 r_i}-e^{-\mu_0(r_i-\Delta t_i)})\right]-\mu_0 t_{ext}}\right)^{\xi_i}\left(e^{-s_i\left[\Delta t_i+\frac{1}{\mu_0}(e^{-\mu_0 r_i}-e^{-\mu_0(r_i-\Delta t_i)})\right]}\right)^{1-\xi_i}$$

and

$$\begin{aligned} log(g_{X|Y}(x|y,\phi)) = &\sum_{i=1}^{d}\xi_i\left(log(s_i)+log(\mu_0)-s_i\left[\Delta t+\frac{1}{\mu_0}(e^{-\mu_0 r}-e^{-\mu_0(r-t)})\right]-\mu_0 t_{ext}\right) \\ &+(1-\xi_i)\left(-s_i\left[\Delta t+\frac{1}{\mu}(e^{-\mu r}-e^{-\mu(r-t)})\right]\right) \end{aligned} \tag{15}$$

```
logweight <- function(pars,df,ct){
  xi = rep(0,times=dim(df)[1])
  xi[df$bte<ct] = 1
  wtT = c(diff(c(0,df$bt)),ct-df$bt[length(df$bt)])
  n.tree = list(wt=wtT,E=df$to)
  n.tree$E[n.tree$E==2] = 1
  E = n.tree$E
  n = c(2,2+cumsum(E)+cumsum(E-1))
  lambda = (pars[1]-(pars[1]-pars[2])*(n/pars[3]))
  mu = pars[2]
  s = lambda*n
  sprob = da.prob(xi=c(xi,0),wt=n.tree$wt,t_ext=c(df$t.ext,Inf),s=s,mu=pars[2],r=c(ct,ct-df$bt),n=n)
  lsprob = sum(log(sprob))
  lrprob = -nllik.tree(pars,n.tree)
  logweight = lrprob-lsprob
  if(logweight==Inf) logweight = -Inf
  return(logweight)
}

prob.ms <- function(wt,t_ext,s,mu,r,n){
  return((s/n)*mu*exp(-s*(wt+(exp(-mu*r)-exp(-mu*(r-wt)))/mu)-mu*t_ext))
}
prob.nospecies <- function(wt,s,mu,r){
  return(exp(-s*(wt+(exp(-mu*r)-exp(-mu*(r-wt)))/mu)))
}
da.prob <- function(xi,wt,t_ext,s,mu,r,n){
  g =ifelse(xi,prob.ms(wt,t_ext,s,mu,r,n),prob.nospecies(wt,s,mu,r))
  return(g)
}
```

Moreover, the montecarlo sampling method

```
sim.sct <- function(brts,pars,m=10,oc=0){
    no_cores <- detectCores() - oc
    cl <- makeCluster(no_cores)
    registerDoParallel(cl)
    trees <- foreach(i = 1:m, combine = list) %dopar% {
      ct =  emphasis::sim.extinct(brts = brts,pars = pars) #complete tree
      lw = ct$logweight
      return(list(wt=ct$wt,E=ct$E,logweight=ct$logweight,lw=lw))
    }
    stopCluster(cl)
    lw = sapply(trees,function(list) list$lw)
    dim = sapply(trees,function(list) length(list$wt))
    lw = lw - max(lw)
    w = exp(lw)
  return(list(rec = trees, w=w,dim=dim))
}
```

The M-step consists on the optimization procedure

$$\phi^* = \underset{\phi\in\mathbb{R}^n}{\operatorname{argmax}} Q(\phi|\phi^*)$$

```
mle.st <- function(S,init_par = c(0.5,0.5,100)){
  po = subplex(par = init_par, fn = Q.approx, st=S,hessian = TRUE)
  return(po)
}
```

The pilot study applied to the DD case is writen on the following code

```
# relative likelihood
rel.llik <- function(S1,p0,p1){
  m = length(S1)
  f1 = vector(mode='numeric',length = m)
  f2 = vector(mode='numeric',length = m)
  d = vector(mode='numeric',length = m)
  S1 = S1$rec[S1$w>0]
  for(i in 1:m){
    s = S1[[i]]
    f1[i] = nllik.tree(pars=p1,tree=s)
    f2[i] = nllik.tree(pars=p0,tree=s)
    d[i] = length(s$tree$wt)
    if(is.na(f1[i])) print(s)
  }
  Delta = -log(sum(f1/f2)/m)
  return(Delta)
}

# Pilot study
pilot.study <- function(brts,epsilon,m1=10,printprocess=FALSE,init_par=c(1.2,0.3,60),l1=20){
  # pilot study suggested by Chan et. al
  pars = init_par
  M = matrix(ncol = 3,nrow = l1)
  H = matrix(ncol = 3,nrow = l1)
  for(i in 1:l1){
    S = sim.sct(brts,pars,m=m1)
    mle =  mle.st(S = S)
    pars = mle$par
    H[i,] = try(diag(solve(mle$hessian))/m1)
    M[i,] = pars
    print(paste('Q:',mle$value,'pars:',pars[1],pars[2],pars[3]))
  }
  l = 10
  PM = M[1:(l1-10),]
  PH = H[1:(l1-10),]
  M = M[(l1-9):l1,]
  H = H[(l1-9):l1,]
  Q = vector(mode="numeric",length = (l1-10))
  MLE = list()
  for(i in 1:10){
    Delta = vector(mode="numeric",length = l)
    Me = matrix(ncol = 3,nrow = l)
    if(printprocess) print(paste('iteration',i))
    for(j in 1:l){
      S = sim.sct(brts,pars=M[i,],m=m1)
      mle = mle.st(S = S)
      pars = mle$par
      Me[j,] = pars
      Delta[j] = rel.llik(S1 = S,p0 = M[i,], p1 = pars)
    }
    MLE[[i]] = Me
    mD = mean(Delta)
    Q[i] = sum((Delta-mD)^2)
  }
  s2 = sum(Q)/((l-1)*(l1-10+1))
  s1 = sqrt(s2)
  m = m1*s1/epsilon
  m = floor(m) + 1
  return(list(m=m,p=M[10,],s1=s1,M=M,H=H,MLE=MLE,PM=PM,PH=PH))
}
```

the code for the whole routine would be

```
mcem.tree <- function(brts,m,init,s1,m1){
  sig = m1*s1/m
  init_m = m
  tol = 2*sig*sqrt(1/5)  # 95% confidence interval
  D = Inf
  k = 1
  print("initializing mcem")
  pars = init
  PARS = pars
  H = c(NULL,NULL,NULL)
  tE = NULL # computing times for E-step
  tM = NULL # computing times for M-step
  efficiency = NULL
  prop = 1
  while(abs(D)>tol){
    if(m*prop<init_m) m <- m/prop
    time = proc.time()
```

```r
    S = sim.sct(brts = brts,pars=pars,m = m)
    prop = sum(S$w>0)/length(S$w)
    efficiency = c(efficiency,prop)
    tE = c(tE,get.time(time))
    time = proc.time()
    M = mle.st(S = S)
    tM = c(tM,get.time(time))
    npars = M$par # next parameters estimation
    h1 = try(diag(solve(M$hessian))/m)
    if(is.numeric(h1)) H =  rbind(H,h1)  # esto deberia crear NaN cuando no es numerico no?
    D = rel.llik(S1 = S,p0 = pars,p1 = npars)
    PARS = rbind(PARS,npars)
    pars = npars
    print(paste("iteration",k,"Q: ",M$value,'proportion of useful trees',prop,'sampling size',m*prop, "
    k = k+1
  }
  return(list(pars=pars,PARS=PARS,H=H,tE=tE,tM=tM,efficiency=efficiency))
}

get.time <- function(time,mode='sec'){
  dif = proc.time()-time
  ti = as.numeric(dif[3])
  if(mode == 'min')  ti = ti/60
  if(mode == 'hou') ti = ti/3600
  return(ti)
}
```

Appendix

A.1

In order to calculate the probability distribution of a missing species we can get close forms for three important cases:

Case 1: $\mu_{t,j} = \mu_0, \quad \forall t > t_0.$

In this case equation 4 would be equivalent to

$$f_M^{t_0}(\Delta t_M^{t_0} = t) = \lambda_{t_0+t}(1 - e^{-\mu_0(T-(t_0+t))})e^{-\int_{t_0}^{t} \lambda_z(1-e^{-\mu_0(T-z)})dz} \tag{16}$$

Case 2: $\lambda_{t,j} = \lambda_0, \quad \forall t > t_0$ and $\mu_{t,j} = \mu_{t,0}, \quad \forall j \in \{1, ..., n_t\}, \forall t > t_0$

In this case equation 4 would be equivalent to

$$f_M^{t_0}(\Delta t_M^{t_0} = t) = \lambda_0 \left(1 - e^{-\int_{t_0+t}^{T} \mu_{s,0} ds}\right) e^{-\lambda_0 \int_{t_0}^{t} \left(1-e^{-\int_z^T \mu_{s,0} ds}\right) dz} \tag{17}$$

Case 3: $\lambda_{t,j} = \lambda_0, \quad \forall t > t_0$ and $\mu_{t,j} = \mu_0, \quad \forall j \in \{1, ..., n_t\}, \forall t > t_0$

In this case equation 4 would be equivalent to

$$f_M^{t_0}(\Delta t_M^{t_0} = t) = \lambda_0(1 - e^{-\mu_0(T-(t_0+t))})e^{-\lambda_0 \int_{t_0}^{t} (1-e^{-\mu_0(T-z)})dz} \tag{18}$$

or

$$f_M^{t_0}(\Delta t_M^{t_0} = t) = \lambda_0(1 - e^{-\mu_0(T-(t_0+t))})e^{-\frac{\lambda_0}{\mu_0}\left(t\mu_0 - e^{-\mu_0(T-t_0)}(e^{\mu_0 t}-1)\right)} \tag{19}$$

We will use cases 1 and 3 in the application to the Diversity-Dependance model.

A.2 Drawing the triad $M^t = (\Delta t_M^t, \tau, t_E)$

For simulation of the nonhomogenous Poisson process described on section 2.2 we use the following results due to Cinlar, 1975:

> The random variables $T_1, T_2, ...$ are event times corresponding to a nonhomogeous Poisson process with expectation function $\Lambda(t)$ if and only if $\Lambda(T_1), \Lambda(T_2)...$ are the event times corresponding to a homogenous Poisson process with rate 1.

The expectation function of the speciation process is

$$\Lambda_m(t) = \int_0^t \lambda_{z,j}\left(1 - e^{-\int_z^T \mu_{s,j}ds}\right)dz$$

whereas the expectation function of the extinction process is

$$\Lambda_e(t) = \int_0^t \mu_{z,j}dz.$$

Note that in both cases we need to know $\mu_{s,j}$ for $s > t$, that is in most cases not possible [4].

For now we foccus on the case $\mu_{t,j} = \mu_0$, so both $\Lambda_m(t)$ and $\Lambda_e(t)$ are well defined at any $t \in (0, T)$. On that case the simulation of the speciation process would follow the steps

0. $t_m = 0$
1. Simulate $\Delta t \sim exp(\lambda_{t,j})$
2. Simulate $u \sim Unif(0, 1)$. (Save u for the extinction process)
3. If $u > 1 - e^{-\mu_0 T}$: Set $T \to T - \Delta t$, $t_m \to t_m + \Delta t$ and go to step 1.
4. Return $\Delta t_M^{t_0} = t_m$

For the extinction process we use the u obtained on step 2. and calculate

$$t_E = \frac{-log(1-u)}{\mu_0}$$

Finally, we simulate

$$\tau \sim \text{Multinom}\,(1, \boldsymbol{p})$$

where $\boldsymbol{p} = (p_1, ..., p_n)$ and

$$p_j = \frac{\lambda_{t_{i+1},j}}{\sum_{k=1}^{n_t}\lambda_{t_{i+1},k}}$$

[4]See project 2 on A.4

A.3 Sampling from non-homogenuos exponential

The waiting time for and speciation of a missing species is distributed exponential with rate

$$r_t = \lambda_t(1 - e^{-\mu_t(T-t)}), \quad t < T$$

where λ_t is the speciation rate, μ_t is the extinction rate and T is the time to present. To simulate random numbers from that distribution we can use two ways. The first assumes that both λ_t and μ_t are constant in a interval $(0, T)$:

0. $t = 0$
1. Draw $t_{spe} \sim Exp(\lambda)$. set $t = t + t_{spe}$
2. Draw $u \sim Unif(1)$
3. If $u > (1 - e^{(T-t)})$ go to step 1
4. Return t

That would be computed on the following R function

```r
rnonh1 <- function(lambda,mu,Ti){
  Total = Ti
  rv = 0
  key = 0
  while(key == 0 & (rv < Total)){
    ex = rexp(1,lambda)
    rv = rv + ex
    u = runif(1)
    if(u < pexp(Total-rv,mu)){
      key = 1
    }
  }
  return(rv)
}
```

The second method of simulating non-homogenous poisson processes would be based on the following theorem (Cinlar, 1975)

> Let $\Lambda(t), t \geq 0$ be a positive-valued, continuous, nondecreasing function. Then the random variables $T_1, T_2, ...$ are event times corresponding to a nonhomogeneous Poisson process with expectation function $\Lambda(t)$ if and only if $\Lambda(T_1), \Lambda(T_2), ...$ are the event times corresponding to a homogeneous Poisson process with rate one.

Then the method follows the two steps

1. Draw $t_e \sim Exp(1)$
2. Return $t = \Lambda^{-1}(t_e)$

This method is more general in the sense that does not need piece-wise constant rates, so it can be applied to time-dependent rates for instance.

To compare with the first method we need to calculate $\Lambda^{-1}(t)$ for constant rates. On this case

$$\Lambda(t) = y = \lambda\left(t - \frac{e^{-\mu T}}{\mu}(e^{\mu t} - 1)\right)$$

to calculate $t = \Lambda^{-1}(y)$ we need to find a solution for t in the equation

$$\frac{e^{-\mu T}}{\mu}e^{\mu t} = t + \frac{e^{-\mu T}}{\mu} - \frac{y}{\lambda}$$

if we consider $x = t + \frac{e^{-\mu T}}{\mu} - \frac{y}{\lambda}$ then

$$\frac{e^{-\mu T}}{\mu} e^{\mu(x - \frac{e^{-\mu T}}{\mu} + \frac{y}{\lambda})} = x$$

or

$$-e^{-\mu T} e^{\mu(-\frac{e^{-\mu T}}{\mu} + \frac{y}{\lambda})} = -\mu x e^{-\mu x}$$

the solution for x on this equation is

$$x = -\frac{W(-e^{-\mu T} e^{\mu(-\frac{e^{-\mu T}}{\mu} + \frac{y}{\lambda})})}{\mu}$$

where $W(\cdot)$ is the Lambert-W function.

So

$$t = \frac{y}{\lambda} - \frac{W(-e^{-\mu T - e^{-\mu T} + \mu\frac{y}{\lambda}})}{\mu} - \frac{e^{-\mu T}}{\mu}$$

and the R implementation would be

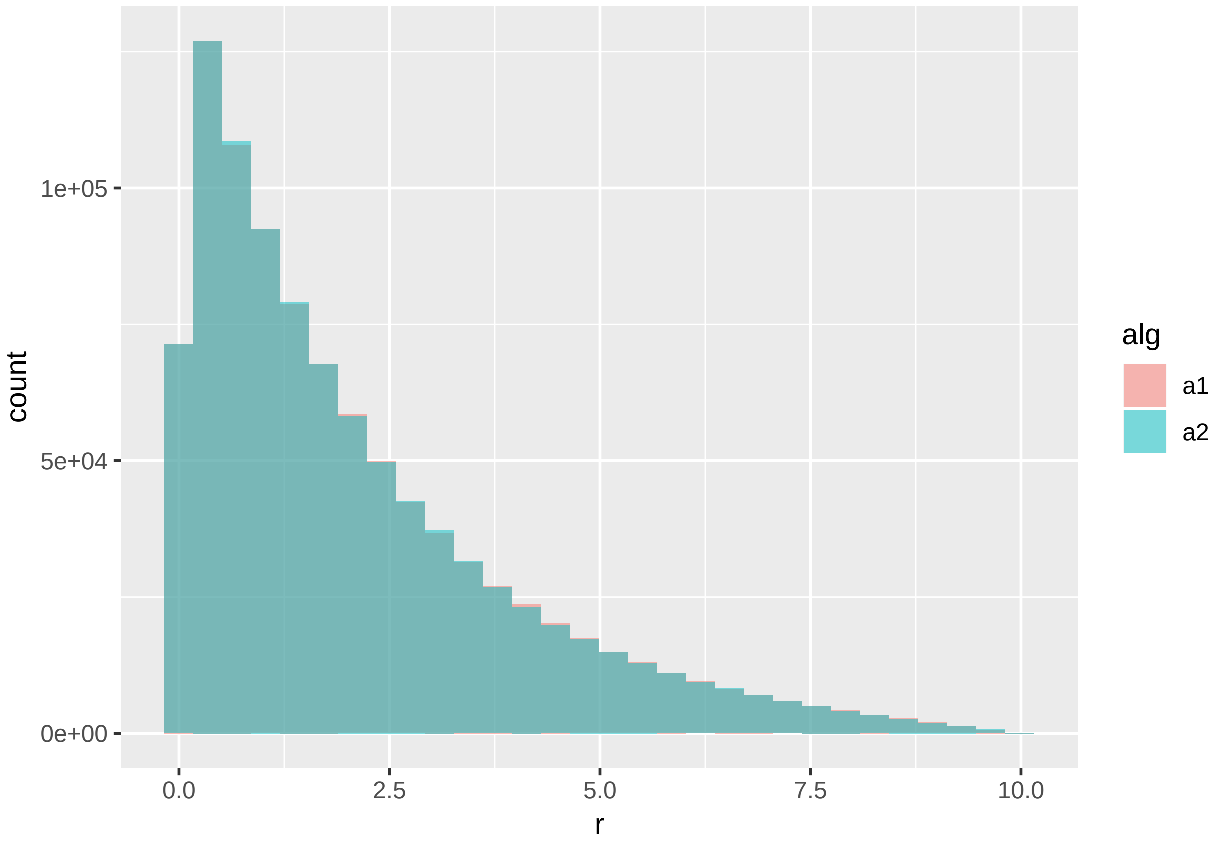
```
IntInv <- function(r,mu,s,u){
  t = -W(-exp(-r*mu+mu*u/s-exp(-r*mu)))/mu+u/s-exp(-r*mu)/mu
  return(t)
}

rnonh2 <- function(lambda,mu,Ti){
  ex = rexp(1)
  top = IntInv(Ti,mu,lambda,ex)
  if(is.na(top)){
    rv = 99
  }else{
    rv = IntInv(Ti,mu,lambda,ex)
  }
  return(rv)
}
```

Thus, we can test both methods to see if they generate the same process

```
lambda = 0.5
mu = 0.2
Ti = r = 10

# exp1

m=1000000
r1 = vector(mode = "numeric",length = m)
r2 = vector(mode = "numeric",length = m)
time = proc.time()
for(i in 1:m){
  r1[i] = rnonh1(lambda,mu,Ti)
}
t1 = get.time(time)
```

```
time = proc.time()
for(i in 1:m){
  r2[i] = rnonh2(lambda,mu,Ti)
}
t2 = get.time(time)

r1 = r1[r1<Ti]
r2 = r2[r2<Ti]

dfh = data.frame(r=c(r1,r2),alg = c(rep("a1",length(r1)),rep("a2",length(r2))))
ggplot(dfh,aes(x=r,fill=alg)) + geom_histogram(alpha=0.5,position='identity')
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



which seems to be the case. We can also check the computational times

```
t1
```

```
## [1] 7.294
```

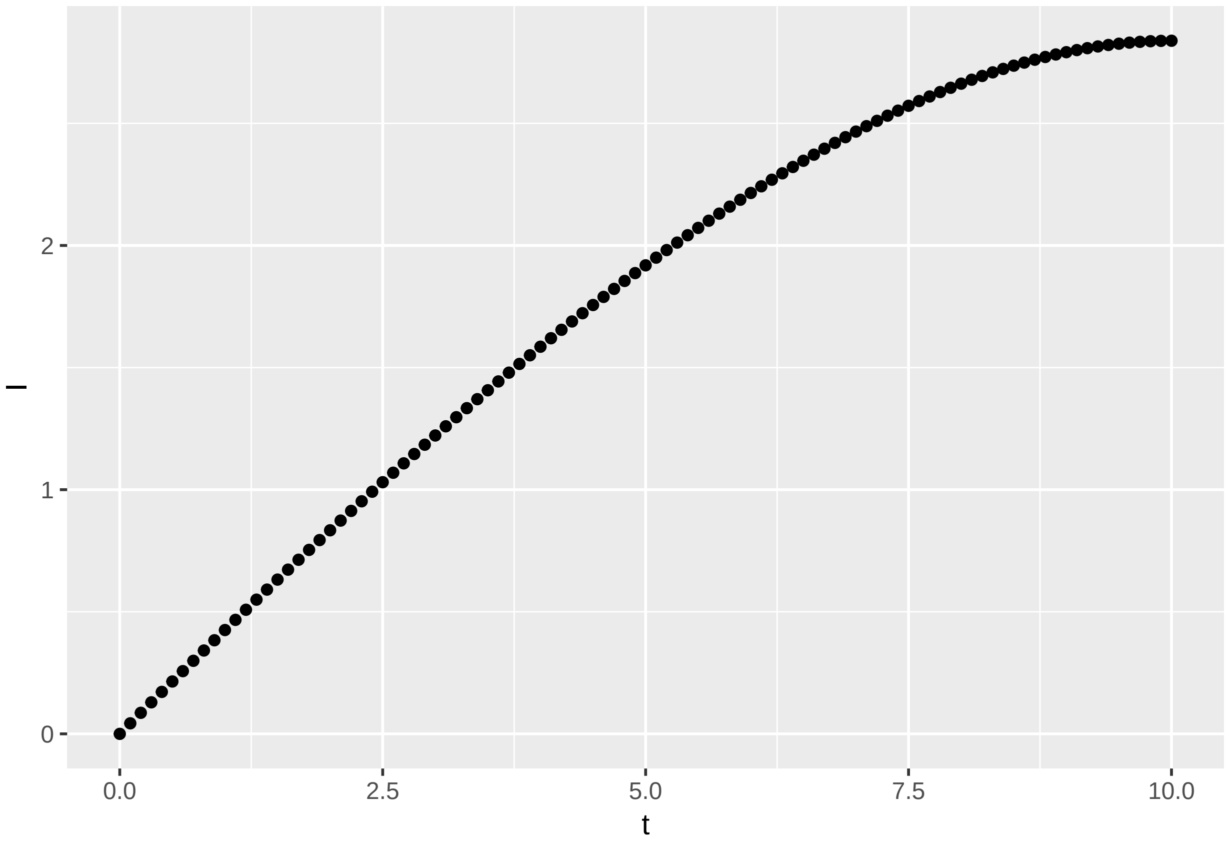
```
t2
```

```
## [1] 64.789
```

So we can conclude that while both methods are equivalent method one is faster, but method two is general.

On the case when a close for for the inverse of the intensity function is not available, we can still do the method on a numerical way. For instance we can vizualize $\Lambda(t)$ integrating and then calculate the inverse.

```
rate <- function(t,mu,lambda,Ti){
  lambda*(1-exp(-mu*(Ti-t)))
}

intensity <- function(t,mu,lambda,Ti){
  int  = integrate(rate,lower=0,upper = t,mu=mu,lambda=lambda,Ti=Ti)
  return(int)
}

t = seq(0,Ti,by=0.1)
I = NULL
for(s in t){
  Int = intensity(s,mu,lambda,Ti)
  I = c(I,Int$value)
}
qplot(t,I)
```



A.4 Further projects / extentions

1. Multiple speciation
2. Diversity-dependance on extinction
3. Time-dependance (continuous (non step-wise) case)
   - check the complete continous version, It might be computationally intractable (?)
   - if that does not work, check what is the discrete aproximation of the continous process.
4. Trait-dependance