YUIMA for simulating traits and phylogenetics in the pcmabc R package

Krzysztof Bartoszek
joint work with Pietro Liò

Third Yuima Workshop
Brixen–Bressanone

LINKÖPING UNIVERSITY

Evolutionary models

Infinitesimal model: offspring follow a Gaussian distribution around mean of parents, with trait independent variance

Neutral evolution, **Brownian motion** (Felsenstein 1985)

- $\mathrm{d}X_t = \sigma_a \mathrm{d}B_t,\ X_0 = X_0$
- $\mathrm{E}\left[X\right](t) = X_0, \quad \mathrm{Var}\left[X\right](t) = \sigma_a^2 t \to \infty$
- $\mathrm{Cov}\left[X_i, X_j\right](t) = \sigma_a^2 t_{ij}$

Adaptation, **Ornstein–Uhlenbeck** (Hansen 1997)

- $\mathrm{d}X_t = -\alpha(X_t - \theta)\mathrm{d}t + \sigma_a \mathrm{d}B_t,\ X_0 = X_0$
- $\mathrm{E}\left[X\right](t) = e^{-\alpha t} X_0 + (1 - e^{-\alpha t})\theta \to \theta$
- $\mathrm{Var}\left[X\right](t) = \frac{\sigma_a^2}{2\alpha}(1 - e^{-2\alpha t}) \to \frac{\sigma_a^2}{2\alpha}$
- $\mathrm{Cov}\left[X_i, X_j\right] = \frac{\sigma_a^2}{2\alpha}(e^{-2\alpha t_{ij}} - e^{-2\alpha t})$

Branching process model

$$\text{Birth and death rates} \quad \lambda(t),\ \mu(t)$$

$$\text{Yule (pure birth)} \quad \lambda(t) = \lambda,\ \mu(t) = 0, \quad T_i \sim exp(\lambda i)$$

pcmabc: trait dependent speciation

Stochastic model for trait X_t
Birth and death rates functions of X_t

`pcmabc`: trait dependent speciation

Stochastic model for trait X_t
Birth and death rates functions of X_t

Birth and death rates $\lambda(t, X_t),\ \mu(t, X_t)$

Estimation: ABC
Distance between phylogenies:
TV distance between exps
`treedist::wRF.dist()`
Distance between trait samples:

$$\sqrt{((\mathrm{Var}_1 - \mathrm{Var}_2)/(\mathrm{Var}_1 + \mathrm{Var}_2))^2 + ((\mathrm{E}_1 - \mathrm{E}_2)/(\mathrm{E}_1 + \mathrm{E}_2))^2}$$

Interface with YUIMA

```
simulate_OU_sde<-function(t,params,X0,step){
    A <- c(paste("(-",params$a11,")
␣␣␣␣*(x1-(",params$psi1,"))",sep=""))

    S <- matrix( params$s11, 1,1)
    yuima.1d <- yuima::setModel(drift = A,
    diffusion = S, state.variable=c("x1"),
    solve.variable=c("x1") )
    simulate_sde_on_branch(t,yuima.1d,X0,step)
}

fbirth_rate_constrained<-function(x,params,...){
    x<-x[2]; params$b/(1+exp(-x))}
```

Simulating *simulate_phylproc()*

1. $X(t)$:=simulate trait trajectory on lineage with length height.

2. Mark birth and death events.

3. End lineage at first death event.

4. If some stopping condition is met end simulation.

5. From each birth event repeat steps 1–5.

Rejection sampling algorithm

$X(t)$:=simulate trait on lineage with length T

Calculate the birth rate $\lambda(t)$ as a function of $X(t)$
Calculate the death rate $\mu(t)$ as a function of $X(t)$

$\Lambda = \max \lambda(t), \quad$ decompose $\lambda(t) = \Lambda p_\lambda(t)$
Simulate a Poisson process for time height and rate Λ
Accept events from the Poisson process with prob. $p_\lambda(t)$

$\mathcal{M} = \max \mu(t), \quad$ decompose $\mu(t) = \mathcal{M} p_\mu(t)$
Simulate a Poisson process for time height and rate $\mathcal{M}$
Accept events from the Poisson process with prob. $p_\mu(t)$

Interface with YUIMA

$$\mathrm{d}x_1(t) = -a_{11}(x_1(t) - \psi_1)\mathrm{d}t + s_{11}\mathrm{d}B(t)$$

```
    A <- c(paste("(-",params$a11,")
␣␣␣␣*(x1-(",params$psi1,"))",sep=""))

    S <- matrix( params$s11, 1,1)

    model.yuima <- yuima::setModel(drift = A,
    diffusion = S, state.variable=c("x1"),
    solve.variable=c("x1") )
```

```
ns<-T/step;    sdedim<-length(X0)
options(warn= -1) ## for warning output of yuima
samp<-yuima::setSampling(Terminal=T, n=ns)

simulobj.yuima <-
yuima::setYuima(model=model.yuima, sampling=samp)

simulobj.yuima <- yuima::simulate(simulobj.yuima,
xinit=X0,space.discretized=TRUE)

time.grid.length<-
length(simulobj.yuima@sampling@grid[[1]])
options(warn= 0)
```

```
trait_data<-NA
if (is.element(".Data",
methods::slotNames(simulobj.yuima@data@original.data))){

    trait_data<-
    simulobj.yuima@data@original.data@.Data

    time_data<-simulobj.yuima@sampling@grid[[1]]
}
```

```
else{ if (dimsde==1){
  time_data<-
  attributes(simulobj.yuima@data@original.data)$index

  trait_data<-c(simulobj.yuima@data@original.data)

  class(trait_data)<-"numeric"
  names(trait_data)<-NULL

  vtorem<-which(time_data>T)
  if (length(vtorem)>0){
    trait_data<-trait_data[-vtorem]
    time_data<-time_data[-vtorem]     }
  time.grid.length<-length(time_data)
  }else{ stop("Error") } }
```

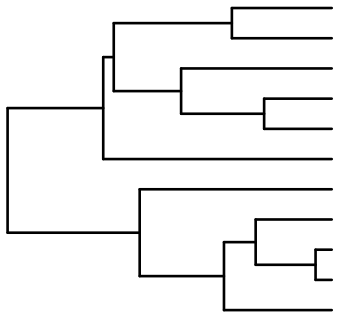
Interface with YUIMA

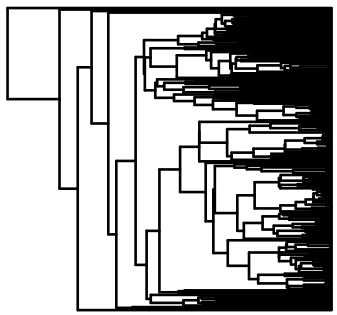
```
timepoints<-length(trait_data)/sdedim

if (timepoints!=time.grid.length)
{warning("Error in Yuima, wrong lengths of grid")}

##simulated data:
rbind(time_data[1:timepoints],
matrix(trait_data,ncol=timepoints,byrow=TRUE))
```

Trait dependent speciation (tree height: 1)

$$\mathrm{d}X_t = -3(X_t - \theta)\mathrm{d}t + 0.25\mathrm{d}W_t, \text{ birth rate: } 10 \cdot |\sin(X_t)|$$

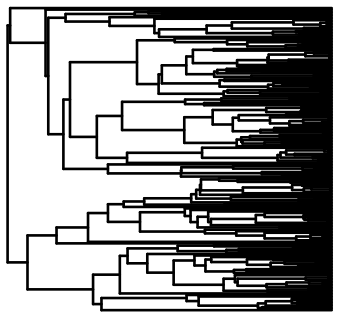
$X_0 = \frac{1}{4}\ \theta = \frac{1}{4}$

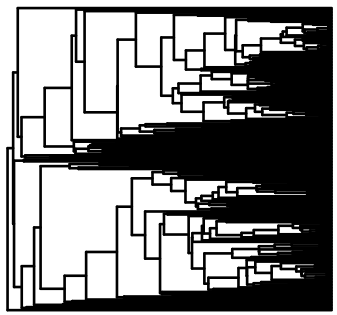




$X_0 = \frac{1}{4}\ \theta = \frac{5}{4}$

$X_0 = \frac{5}{4}\ \theta = \frac{1}{4}$





$X_0 = \frac{5}{4}\ \theta = \frac{5}{4}$

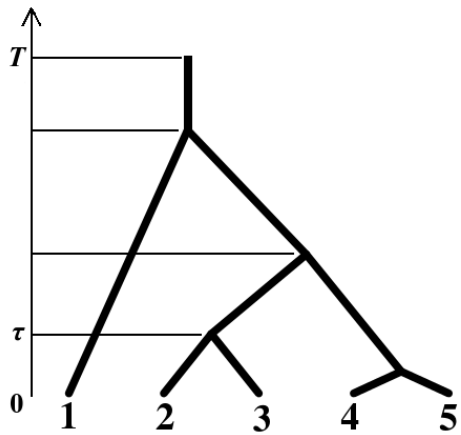
Trait dependent speciation (ABC)

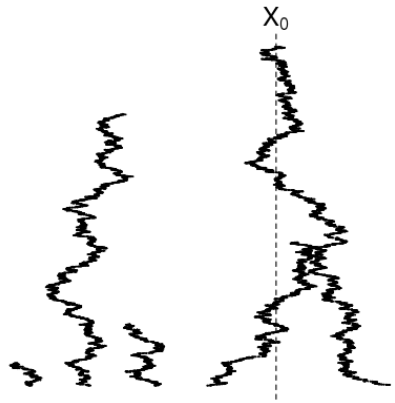
```
PCM_ABC(phyltree,data,par0,simulate_OU_sde,
fbirth,NULL,X0=0,step=0.001,abcsteps=1000,
eps=0.2,tree.fixed=FALSE,
dist_method=c("variancemean","wRF.dist"))
```

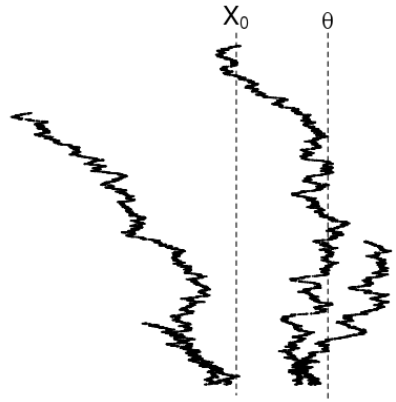
$$\mathrm{d}X_t = -\alpha(X_t - \theta)\mathrm{d}t + \sigma\mathrm{d}W_t = -X_t\mathrm{d}t + \mathrm{d}W_t$$
birth rate: $b/(1+\exp(-X_t)) = 5/(1+\exp(-X_t))$

$$\hat{\alpha} = 0.946, \quad \hat{\sigma} = 0.863, \quad \hat{\theta} = 0.184, \quad \hat{b} = 4.638$$

Including jumps







- $X_k(t_k^+) = X_k(t_k^-) + Z_k \cdot Y_k$
- $Y_k \sim \mathcal{N}(0, \sigma^2_{c,k})$
- $P(Z_k = 1) = p_k,\ P(Z_k = 0) = 1 - p_k$

Thank you!! Acknowledgements and references

P. Liò

Stefano Iacus, Yoshida Nakahiro

Vetenskapsrådet grant no. 2017–04951

K.B., P.L., **Modelling trait dependent speciation with Approximate Bayesian Computation.** *Acta Phys. Pol. B Proc. Suppl.* 12(1) 25–47, 2019

www.liu.se

LINKÖPING UNIVERSITY