A Tutorial for the poumm package

Venelin Mitov

Introduction

The **poumm** package provides an implementation of the Pylogenetic Mixed Model assuming an Ornstein-Uhlenbeck (OU) or a Brownian Motion (BM) process of trait evolution along the branches. The package comes with an efficient likelihood calculation that scales to large phylogenies (currently tested with up to 10000 tips) and, if needed, can perform the likelihood calculation using the package **Rmpfr** for high precision floating point operations. Currently, the package depends on the package **ape** and the package **data.table**. The most important functions of the package are "likVTReeOU" (see ?likVTreeOU), which performs likelihood calculation, "mlOUTree" (type ?mlOUTree), which fits POUMM to a tree and data at the tips using maximum likelihood and mcmcOUTree, which runs an MCMC fit to a tree and data. Although the algorithm for likelihood calculation has been extensively tested and applied on various real as well as synthetic data, the package is still in a beta-version for the following reasons:

- The signature of the functions in the package may change;
- Some of the functions are not yet documented.

Note: The high-precision likelihood calculation is usually needed in case of accumulating numerical errors due to extreme values of some of the parameters alpha, theta, sigma or sigmae or extreme (i.e. very long or very short) branch lengths in the phylogeny. As the Rmpfr package is currently not supported on all linux platforms, the poumm package doesn't explicitly depend on it, but you will need to install it manually and import it in case you specify usempfr=2 in the call to likVTreeOU.

Installing the package

```
install.packages('poumm_0.1.tgz', type='binary', repos=NULL)
```

A Hello World example

```
library(poumm)

set.seed(1)
# Number of tips
N <- 1000
# true parameters
alpha <- 1
theta <- 4.5
sigma <- 2
sigmae <- 2

tree <- rtree(N)
tree$edge.length <- tree$edge.length/10
```

```
g <- generateTraitOU(tree, g0=6, alpha=alpha, theta=theta, sigma=sigma)
v <- rnorm(N, mean=0, sd=sigmae)+g[1:N]

#likelihood at the true value
lik <- likVTreeOU(v, tree, alpha, theta, sigma, sigmae, log=T, distgr='maxlik')

# setting the maxit parameter to a very small value (i.e. 2) while the
# option 'hybrid' is on, would transform the pso algorithm into a gradient
# descent with a number of different original locations.
fit <- mlOUTree(v=v, tree=tree, distgr='maxlik')

# maximum likelihood fit:
print(fit$par)
```

```
##    alpha    theta    sigma   sigmae
## 0.723194 6.900631 1.624662 1.993503
```

```
# ml likelihood value versus value at the original true params
print(c(-fit$value, lik))
```

```
## [1] -2181.628 -2187.276
```

```
# estimating broad-sense heritability
print(c(calcH2AtTime(fit$par[1], fit$par[3], fit$par[4], t=Inf, tm=0),
        calcH2AtTime(alpha, sigma, sigmae, t=Inf, tm=0)))
```

```
## [1] 0.3146958 0.3333333
```

Further information and help

For further information, look in the package help-pages. If you cannot find the answer, please write me to venelin.mitov@bsse.ethz.ch.