

Lab 3 R Source Code

Code:

```
1 #
2 # Shawn Schwartz , 2019
3 # EEB C174 UCLA Spring 2019
4 # Lab 3 HW – Simulating Brownian Motion
5 #
6
7 #clean up workspace
8 rm(list=ls())
9
10 #includes
11 library(phytools)
12 library(dplyr)
13 library(car)
14 library(geiger)
15
16 #directories
17 wd_path <- "~/Developer/EEB-C174-Labs/Lab3"
18 setwd(wd_path)
19 output_path <- "output/"
20 resources_path <- "resources/"
21
22 ##### 1. Read in Time-Calibrated Tree #####
23 tree <- read.tree(paste0(resources_path,"Labridae.tre"))
24 tree <- ladderize(tree)
25 pdf(paste0(output_path,"1_tc_tree_labridae.pdf"))
26   plot(tree, cex = 0.25)
27   axisPhylo()
28   add.scale.bar()
29 dev.off()
30
31 ##### 2. Simulate Brownian Motion #####
32 sig2 <- 0.01 # sigma^2
33 n_steps <- 1:100 # number of steps (time)
34
35 ## simulate Brownian evolution on a tree with fastBM
36 #x <- fastBM(tree, sig2 = sig2, internal = TRUE)
37 x <- fastBM(tree, sig2 = sig2, a = 0)
38 ## visualize Brownian evolution on a tree
```

```
39 pdf(paste0(output_path, "2_tc_fast_bm_a0.pdf"))
40   phenogram(tree, x, spread.labels = TRUE, spread.cost = c(1, 0),
41     fsiz = 0.3)
42
43 x <- fastBM(tree, sig2 = sig2, internal = TRUE)
44 pdf(paste0(output_path, "2_tc_fast_bm_internal.pdf"))
45   phenogram(tree, x, spread.labels = TRUE, spread.cost = c(1, 0),
46     fsiz = 0.3)
47
48 ##### 3. Visualize Tree with Simulated Trait Values #####
49 pdf(paste0(output_path, "3_fast_bm_viz.pdf"))
50   plotTree.wBars(tree, x, tip.labels = TRUE, fsiz = 0.3)
51 dev.off()
52
53 ##### 4. BMLk() #####
54 ##### Fitting BM Models #####
55 BMLk <- function(C, inv.C, sigmasq, root.state, data) {
56   N <- length(data); # the number of tips
57   EX <- rep(root.state, N) # creates a vector of the expected
58     trait value - which under  $V <- C * \text{sigmasq}$ ; # multiply the
59     entries in C by the BM rate
60   V <- C * sigmasq;
61   inv.V <- inv.C * sigmasq ^ -1; # do the same for the inverted
62     matrix using the inverse of the rate
63   lnNum <- -0.5 * (data - EX) %*% inv.V %*% (data - EX)
64   lnDen <- log(sqrt((2*pi)^N * det(V)))
65   L <- lnNum - lnDen
66   return(L);
67 }
68
69 c <- vcvPhylo(tree, anc.nodes = FALSE)
70 inv.c <- solve(c) #inverse of phylogenetic variance-covariance
71   matrix
72
73 data <- fastBM(tree, sig2 = sig2, a = 0)
74
75 estimate_1 <- BMLk(c, inv.c, sigmasq = sig2, root.state = 0, data
76   = data) #true rate
77 estimate_2 <- BMLk(c, inv.c, sigmasq = 0.02, root.state = 0, data
78   = data)
```

```
73 estimate_3 <- BMLk(c, inv.c, sigmasq = 0.03, root.state = 0, data
  = data)
74 estimate_4 <- BMLk(c, inv.c, sigmasq = 0.03, root.state = 0, data
  = data)
75 estimate_5 <- BMLk(c, inv.c, sigmasq = 0.05, root.state = 0, data
  = data)
76
77 estimates <- c(estimate_1, estimate_2, estimate_3, estimate_4,
  estimate_5)
78 estimates
79
80 ##### 5. Visualize Likelihood Surface #####
81 vals <- numeric(100)
82 sigmas <- (1:100)/1000
83
84 for(i in 1:100)
85 {
86   vals[i] <- BMLk(c, inv.c, sigmasq = sigmas[i], root.state = 0,
     data = data)
87 }
88
89 vals[!is.finite(vals)] <- 0
90
91 pdf(paste0(output_path, "5_likelihood_surface.pdf"))
92 plot(sigmas, vals, xlim = c(0.011, 0.079), type = "l", ylab = "
  ln(L)")
93 dev.off()
94
95 mod_estimates <- vals[vals < max(vals)]
96 best_estimate <- which.max(mod_estimates)
97 best_estimate
98 vals_estimate <- mod_estimates[best_estimate]
99 vals_estimate
100 best_estimate_sigma <- sigmas[best_estimate]
101 best_estimate_sigma
102
103 # check to see what the best estimate is (actual value compared to
  surface plot)
104 bm_fit <- fitContinuous(tree, data, model = "BM")
105 bm_fit
106 ML_z0 <- bm_fit$opt$z0
107 ML_sig2 <- bm_fit$opt$sigsq
```

```
108 BMlk(c, inv.c, sigmasq = ML_sig2, root.state = ML_z0, data = data)
```