Tree Simulation assignment

Gaurav Kandlikar

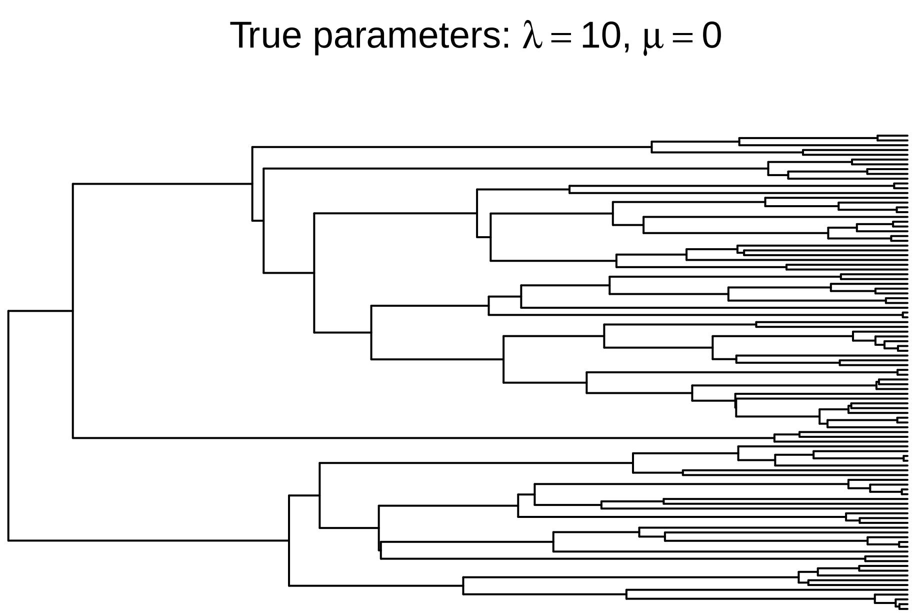
November 5, 2015

getting set up

```
library(geiger); library(phytools); library(diversitree)

source("~/grad/Dropbox/200a 2015 (1)/alfaro-lectures/lab1-trees/rabosky_functions.R")
```

Using `simulateTree` in `phytools` to simulate tree under birth-death model

```
pars <- c(10, 0) # First lambda, then mu- these are the true parameters underlying the tree

tt <- simulateTree(pars = pars, max.taxa = 100)

plot(tt, show.tip.label = F, main = bquote(paste("True parameters: ",
                                                 lambda == .(pars[1]),
                                                 ", ", mu == .(pars[2]))))
```



We know the paramters underlying the birth-death process here, so we can check how well we can estimate them froom the tree. For this we use the `made.bd()` function of `diversitree`. This is the description:

Prepare to run a constant rate birth-death model on a phylogenetic tree. This fits the Nee et al. 1994 equation, duplicating the birthdeath function in ape. Differences with that function include (1) the function is not constrained to positive diversification rates (mu can exceed lambda), (2) [eventual] support for both random taxon sampling and unresolved terminal clades (but see bd.ext), and (3) run both MCMC and MLE fits to birth death trees.

We then use `fitDiversitree()` from `rabosky_functions.r`

```r
make.bd(tt)
```

```
## Constant rate birth-death likelihood function:
##   * Parameter vector takes 2 elements:
##      - lambda, mu
##   * Function takes arguments (with defaults)
##      - pars: Parameter vector
##      - condition.surv [TRUE]: Condition likelihood on survial?
##      - intermediates [FALSE]: Also return intermediate values?
##   * Phylogeny with 100 tips and 99 nodes
##      - Taxa: sp1, sp2, sp3, sp4, sp5, sp6, sp7, sp8, sp9, ...
##   * Reference:
##      - Nee et al. (1994) doi:10.1098/rstb.1994.0068
## R definition:
## function (pars, condition.surv = TRUE, intermediates = FALSE)
```

```r
fit <- fitDiversitree(make.bd(tt))

# Extract parameter estimates
fit$pars
```

```
##    lambda        mu
## 12.570808  5.078002
```

The `lambda` and `mu` estimates above seem significantly different from the true values of 10 and 0.

We can do a null model to get a 95% confidence interval on these estimates:

```r
reps <- 1000
pars <- c(10, 0) # First lambda, then mu- these are the true parameters underlying the tree

lambdas <- numeric(reps)
mus <- numeric(reps)

for (i in 1:reps) {
  fit <- fitDiversitree(make.bd(simulateTree(pars = pars, max.taxa = 100)))
  estimates <- fit$pars
  lambdas[i] <- estimates["lambda"]
  mus[i] <- estimates["mu"]
}

mean(lambdas); mean(mus)
```

```
## [1] 10.83928
```

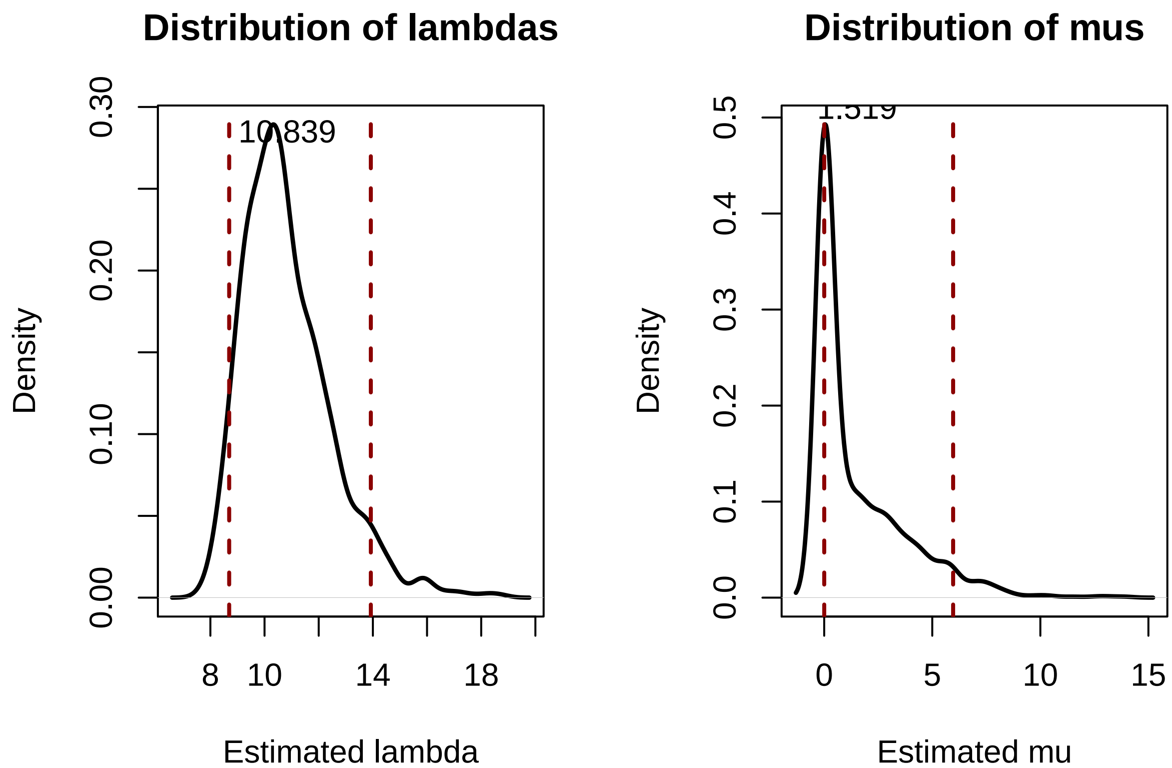
```
## [1] 1.518696
```

```
# Quantiles
mu_lines <- quantile(mus, probs = c(0.05, 0.95))
lambda_lines <- quantile(lambdas, probs = c(0.05, 0.95))

mu_lines; lambda_lines
```

```
##       5%      95%
## 0.001000 5.965295
```

```
##        5%       95%
##  8.693827 13.924250
```

```
# Plot
par(mfrow = c(1,2))

# Lambda plot
plot(density(lambdas), main = "Distribution of lambdas", xlab = "Estimated lambda", lwd = 2.2)
abline(v = lambda_lines, lwd = 2, col = "darkred", lty = 2)
# text(x = c(lambda_lines[1]-1, lambda_lines[2]+1), y = 0.285, labels = round(lambda_lines, 3))
text(x = mean(lambdas), y = 0.285, labels = round(mean(lambdas), 3))

# Mu plot
plot(density(mus), main = "Distribution of mus", xlab = "Estimated mu", lwd = 2.2)
abline(v = mu_lines, lwd = 2, col = "darkred", lty = 2)
# text(x = c(mu_lines[1]-1, mu_lines[2]+1), y = 0.51, labels = round(mu_lines, 3))
text(x = mean(mus), y = 0.51, labels = round(mean(mus), 3))
```



TO DO:

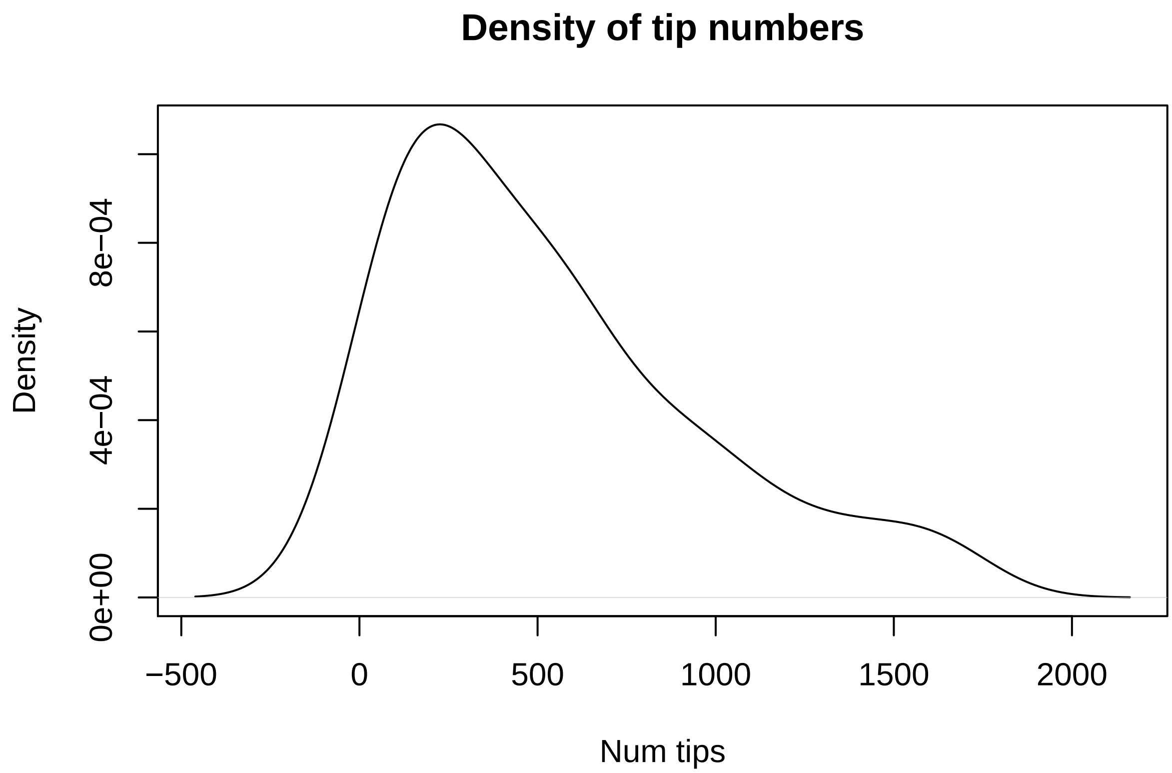
What does this imply about what we can learn from empirical studies of molecular phylogenies about diversification?

Exercise 2

Simulate 100 trees under a constant rate of birth and death. Extract the number of species from each tree. Create a histogram of the resulting distribution. How much stochasticity is associated with the outcome of a birth-death process? What does this suggest about our ability to intuitively identify clades that have undergone exceptional speciation?

I will use `pbtree()` in `phytools` to make trees with constant birth death rates:

```
reps <- 100
num_tips <- numeric(reps)
for (i in 1:reps) {
  tt <- pbtree(b = .5, d = .05, t = 12)
  num_tips[i] <- length(tt$tip.label)
}
par(mfrow = c(1,1))
plot(density(num_tips), main = "Density of tip numbers", xlab = "Num tips")
```




```r
mean(num_tips); quantile(num_tips, c(0.05, 0.95))
```

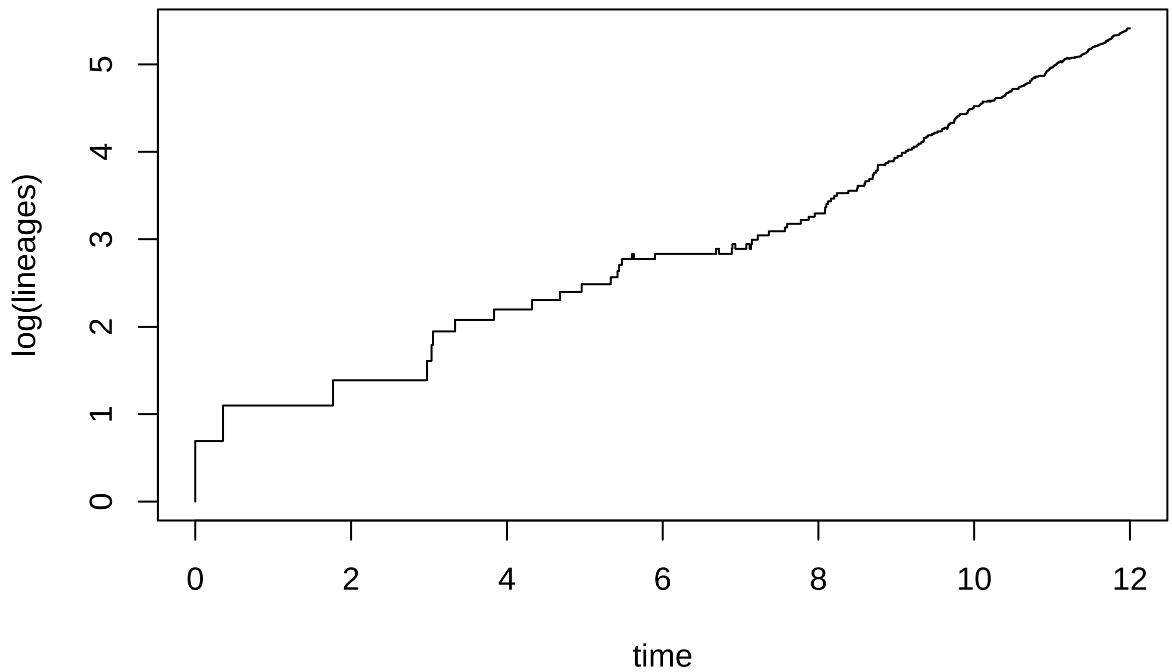
```
## [1] 531.95
```

```
##      5%     95%
##   40.70 1458.25
```

There is clearly a wide range of final number of species derived from the same underlying birth/death rates- this means that the estimates we derive from a phylogeny can in turn lead to a wide range of topologies.

We can use `ltt()` to make a lineage-through-time plot

```r
ltt(tt) # This will plot the last tree made in the loop above
```




```
## $ltt
##       242 243 432 244 245 246 274 291 247 292 251 461 390 293 433 434 211
##    1    2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  16
## 281 467 147 248 435   1 294 126 314 462 252 372 295 381 463 391 256 297
##  17  18  17  18  19  18  19  18  19  20  21  22  23  24  25  26  27  28
## 468 275 347 282 315 478 392 454 286 408 317 299 479 278 283 305 427 353
##  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46
## 373 306 357 318 464 307 393 308 279 374 441 266 436 409 267 276 359 410
##  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63  64
## 444 394 309 469 413 300 396 280 241 397 383 455 358 342 472 367 480 277
##  65  66  67  68  69  70  71  72  71  72  73  74  75  76  77  78  79  80
## 428 257 258 398 445 473 310 377 354 298 460 474 348 319 259 439 470 151
##  81  82  83  84  85  86  87  88  89  90  91  92  93  94  95  96  97  96
## 411 336 148 447 429 320 456 437 360 384 287 457 355 477 249 312 414 415
##  97  98  97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112
## 440 385 268 253 321 378 382 343 332 301 443 350 400 446 379 417 152 337
## 113 114 115 116 117 118 119 120 121 122 123 124 125 126 127 128 127 128
## 218 405 401 322 418 368 361 448 475 362 449 419 349 323 386 262 303 106
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 142
## 450 416 324 371 438 476 363 402 334 458 451 263 136 110 338 264 481 313
## 143 144 145 146 147 148 149 150 151 152 153 154 153 152 153 154 155 156
## 254 270 465 265 193  64 311 260 380 134 325 423 395 399 271 442 255 331
## 157 158 159 160 159 158 159 160 161 160 161 162 163 164 165 166 167 168
## 375 420 250 352 369 328 366 452 326 284 302 406 412 388 422 141 333 351
## 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 182 183 184
## 426 387 273 356 466 424 346 430 316 329 431  83 459 269 403 421 370 330
```

```
## 185 186 187 188 189 190 191 192 193 194 195 194 195 196 197 198 199 200
## 364 453 344 327 340 288 376 425 404 365 341 289 290 345 296 335 261 407
## 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216 217 218
## 272 389 339 304 471 285 131
## 219 220 221 222 223 224 224
##
## $times
##                    242        243        432        244        245
##  0.0000000  0.0000000  0.3552876  1.7669864  2.9736889  3.0339120
##        246        274        291        247        292        251
##  3.0507227  3.3366925  3.8356173  4.3224678  4.6810529  4.9602006
##        461        390        293        433        434        211
##  5.3328693  5.4209698  5.4434936  5.4789667  5.6089006  5.6293696
##        281        467        147        248        435          1
##  5.9028241  6.6856558  6.7258151  6.8870805  6.8957407  6.9340468
##        294        126        314        462        252        372
##  7.0748718  7.1161094  7.1367954  7.1448932  7.2204295  7.3643409
##        295        381        463        391        256        297
##  7.5711560  7.6000019  7.7736396  7.8742608  7.9522886  8.0851573
##        468        275        347        282        315        478
##  8.0882592  8.1020961  8.1235387  8.1617651  8.2039502  8.2365231
##        392        454        286        408        317        299
##  8.3845542  8.4962080  8.5039527  8.5911894  8.6020689  8.6529174
##        479        278        283        305        427        353
##  8.6966484  8.6989320  8.7115292  8.7414610  8.7601019  8.7618872
##        373        306        357        318        464        307
##  8.7658811  8.8608471  8.8980884  8.9666226  8.9768157  9.0166838
##        393        308        279        374        441        266
##  9.0693164  9.0720076  9.1194890  9.1517348  9.2019803  9.2244504
##        436        409        267        276        359        410
##  9.2683043  9.2861704  9.3203387  9.3459374  9.3545391  9.3547458
##        444        394        309        469        413        300
##  9.3863494  9.4063426  9.4591815  9.4889051  9.5280341  9.5823441
##        396        280        241        397        383        455
##  9.5887456  9.6178297  9.6513986  9.6570554  9.6625332  9.6642803
##        358        342        472        367        480        277
##  9.6770497  9.6945485  9.7425263  9.7450987  9.7454302  9.7560317
##        428        257        258        398        445        473
##  9.7694596  9.7835865  9.8083282  9.8156320  9.9018508  9.9155799
##        310        377        354        298        460        474
##  9.9158778  9.9251945  9.9388478  9.9771269  9.9917047  9.9958562
##        348        319        259        439        470        151
## 10.0650768 10.0768747 10.0877788 10.1066427 10.1078390 10.1143348
##        411        336        148        447        429        320
## 10.1144907 10.1770853 10.1983265 10.2147557 10.2562597 10.2655996
##        456        437        360        384        287        457
## 10.2741599 10.3534339 10.3661320 10.3923693 10.3988368 10.4072559
##        355        477        249        312        414        415
## 10.4137517 10.4401655 10.4548563 10.4800438 10.4818964 10.4897691
##        440        385        268        253        321        378
## 10.5685932 10.5704209 10.5799737 10.6077836 10.6403023 10.6453696
##        382        343        332        301        443        350
## 10.6683764 10.6821709 10.7143872 10.7148633 10.7168502 10.7358029
##        400        446        379        417        152        337
```

```
## 10.7371572 10.7426962 10.7572612 10.7670671 10.7703325 10.7826477
##        218        405        401        322        418        368
## 10.7848793 10.7870079 10.7884408 10.8240535 10.9005233 10.9070778
##        361        448        475        362        449        419
## 10.9096527 10.9123680 10.9240046 10.9240304 10.9288513 10.9335080
##        349        323        386        262        303        106
## 10.9496389 10.9619726 10.9661044 10.9702045 10.9844815 10.9856071
##        450        416        324        371        438        476
## 10.9857027 11.0085939 11.0106732 11.0228342 11.0285114 11.0480002
##        363        402        334        458        451        263
## 11.0562574 11.0621812 11.0678906 11.0795335 11.0950317 11.1089266
##        136        110        338        264        481        313
## 11.1133881 11.1165174 11.1313390 11.1385875 11.1387596 11.1519129
##        254        270        465        265        193         64
## 11.1607320 11.1619880 11.1843411 11.1954593 11.1992226 11.2160828
##        311        260        380        134        325        423
## 11.2195974 11.2446141 11.2882413 11.2927784 11.2963003 11.3228101
##        395        399        271        442        255        331
## 11.3579519 11.3728080 11.3754435 11.3821923 11.3942806 11.4123384
##        375        420        250        352        369        328
## 11.4311608 11.4375697 11.4472122 11.4570703 11.4592633 11.4603846
##        366        452        326        284        302        406
## 11.4651590 11.4727135 11.4786642 11.5005176 11.5059397 11.5153280
##        412        388        422        141        333        351
## 11.5236820 11.5392718 11.5467957 11.5566775 11.5571394 11.5794188
##        426        387        273        356        466        424
## 11.5913603 11.5999593 11.6150891 11.6403654 11.6564485 11.6657767
##        346        430        316        329        431         83
## 11.6827733 11.6905470 11.6920650 11.6926046 11.7039841 11.7162841
##        459        269        403        421        370        330
## 11.7181726 11.7194741 11.7217126 11.7379651 11.7529213 11.7643745
##        364        453        344        327        340        288
## 11.7660391 11.7671473 11.7714284 11.7792056 11.7825568 11.7879196
##        376        425        404        365        341        289
## 11.7936777 11.8408378 11.8605330 11.8659610 11.8689699 11.8713268
##        290        345        296        335        261        407
## 11.8874724 11.8967423 11.9027802 11.9151691 11.9290873 11.9437227
##        272        389        339        304        471        285
## 11.9530488 11.9565431 11.9571694 11.9634954 11.9641838 11.9694177
##        131
## 12.0000000
##
## $gamma
## [1] 0.6070936
##
## $p
## [1] 0.5437888
```

```r
# Should the slope of this line be equal to (b-d)?
# abline(a = c(0, 0.45))
```

Next

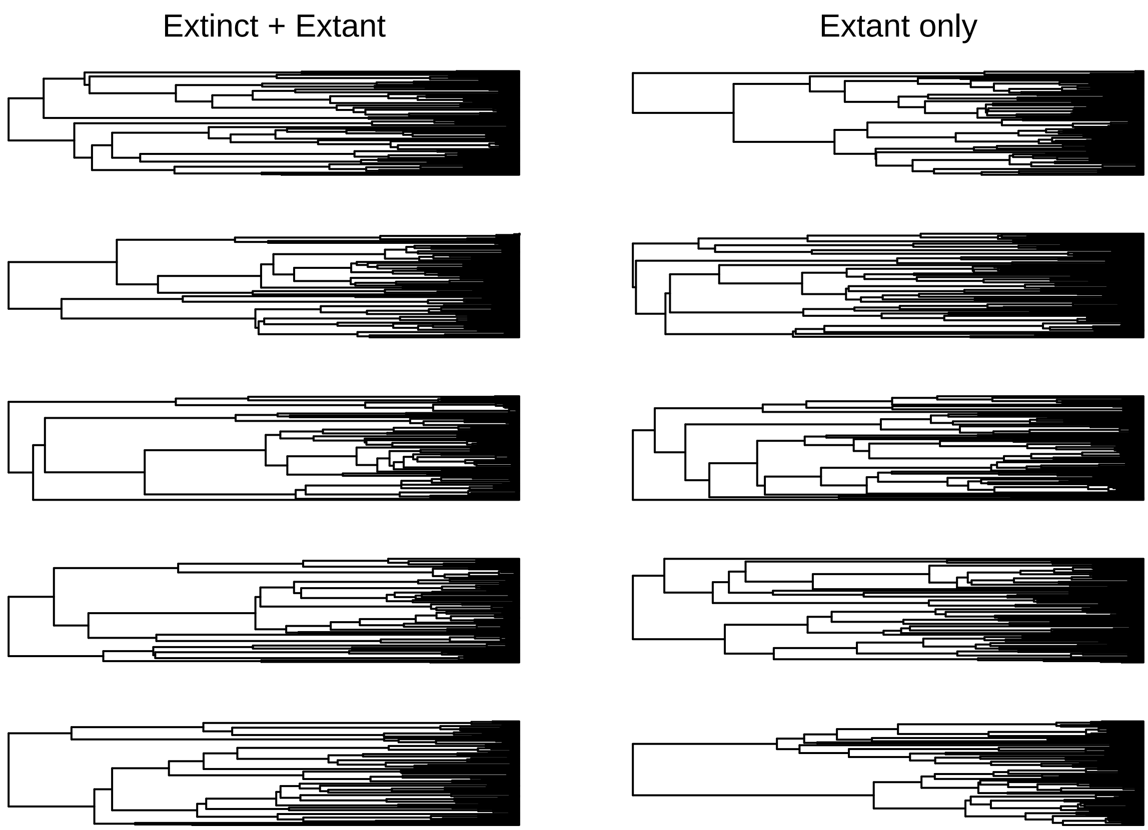
Simulation where extinct taxa are analysed. We use te function `sim.bdtree` from `geiger`.

```
pars <- c(10, 5)
tt <- simulateTree(pars, max.taxa=100)
ttEx <- sim.bdtree(b = 10, d = 1, stop = "taxa", t = 100, n = 100 )

livingOnly <- drop.fossil(ttEx) # Subset to extant taxa only
```

Exercise 3. What does extinction do to the shape of the tree?

Simulate 5 trees with and without extinction that have similar net diversification rates. Can you say anything about the general shape of the trees that have been simulated with extinction?

```
reps <- 5

par(mfrow = c(reps ,2), mar = c(1,1,1,1), oma = c(0,0,2,0))

for(i in 1:reps) {

  pars <- c(10, 5)
  tt <- simulateTree(pars, max.taxa=100)

  ttEx <- sim.bdtree(b = 10, d = 1, stop = "taxa", t = 100, n = 100)
  livingOnly <- drop.fossil(ttEx) # Subset to extant taxa only

  plot(tt, show.tip.label = FALSE)
  if(i == 1) (mtext(side = 3, line = 1, text = "Extinct + Extant"))
  plot(livingOnly, show.tip.label = FALSE)
  if(i == 1) (mtext(side = 3, line = 1, text = "Extant only"))
}
```


Extinct + Extant
Extant only