dmeav02 Usage

Francisco Richter M.

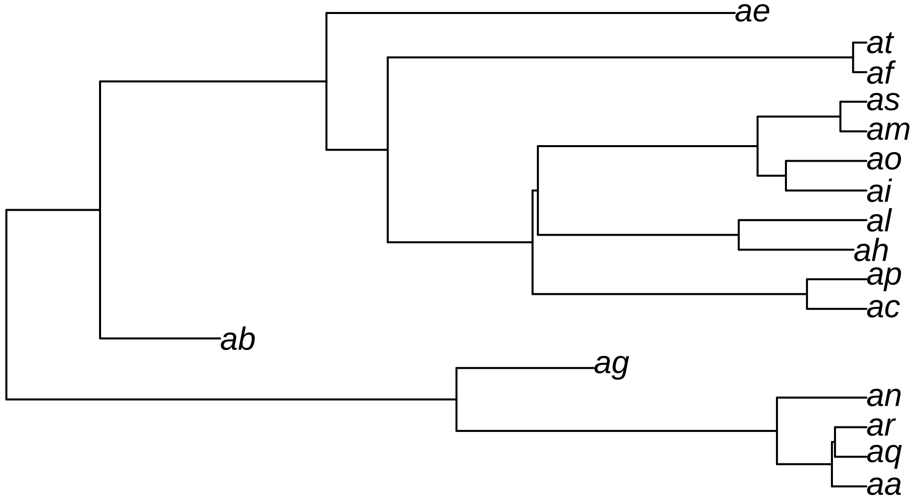
June 1, 2017

tree simulation and consistency

functions

1. `sim.tree` to simulate trees
2. `phylo2vectors` to convert a phylo into the set of vectors characterizing the tree

```
s = sim.tree(ct=6)
plot(s$phylo)
```



```
s$tree
```

```
## $wt
##  [1] 0.56409120 0.59227621 0.75881789 0.67193718 0.38730953 0.43466403
##  [7] 0.48070737 0.03372014 0.35309800 0.89151773 0.02503882 0.11899483
## [13] 0.12247843 0.05664045 0.13307675 0.15770937 0.01933013 0.03406518
## [19] 0.08045422 0.00293288 0.08113967
##
## $E
##  [1] 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 0
##
## $n
##  [1]  1  2  3  2  3  4  5  6  7  6  5  6  7  8  9 10 11 12 13 14 13
##
## $S
##  [1] "aa" "ab" "ab" "ac" "ac" "aa" "ac" "ah" "ag" "ae" "ah" "ai" "aa" "ai"
## [15] "ac" "aa" "aq" "am" "af" "ah"
##
## $br
##  [1] 0.5640912 1.1563674 1.9151853 2.5871225 2.9744320 3.4090960 3.8898034
##  [8] 3.9235235 4.2766215 5.1681393 5.1931781 5.3121729 5.4346513 5.4912918
## [15] 5.6243685 5.7820779 5.8014080 5.8354732 5.9159274 5.9188603 6.0000000
```

```
phylo2vectors(s$phylo)
```

```
## $wt
##  [1] 0.56409120 0.59227621 0.75881789 0.67193718 0.38730953 0.43466403
##  [7] 0.48070737 0.03372014 0.35309800 0.89151773 0.02503882 0.11899483
## [13] 0.12247843 0.05664045 0.13307675 0.15770937 0.01933013 0.03406518
## [19] 0.08045422 0.00293288 0.08113967
##
## $E
##  [1] 1 1 0 1 1 1 1 1 0 0 1 1 1 1 1 1 1 1 1 1 0
##
## $n
##  [1]  1  2  3  2  3  4  5  6  7  6  5  6  7  8  9 10 11 12 13 14 13
```

```
all.equal(s$tree$wt,phylo2vectors(s$phylo)$wt)
```

```
## [1] TRUE
```

```
all.equal(s$tree$E,phylo2vectors(s$phylo)$E)
```

```
## [1] TRUE
```

```
all.equal(s$tree$n,phylo2vectors(s$phylo)$n)
```

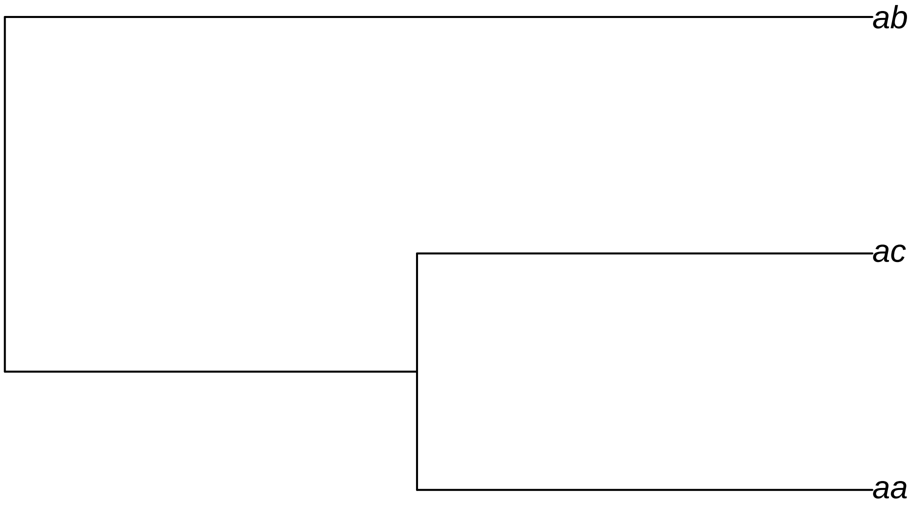
```
## [1] TRUE
```

to do:

1. phylo2vectorsdoes not give the vector S corresponding to topology. add it
2. remove the root on vector outputs, keep consistency
3. arenge the vectors2phylo function and make a lot simpler the sim.tree using it
4. add the log in an log output file

Tree manipulation

```
s = sim.tree(ct=2,seed=5)
plot(s$phylo)
```



```
s2=phylo2vectors(s$phylo)
s2
```

```
## $wt
## [1] 1.0954824 0.4297802 0.4747374
##
## $E
## [1] 1 1
##
## $n
## [1] 1 2 3
```

```
s$tree
```

```
## $wt
## [1] 1.0954824 0.4297802 0.4747374
##
## $E
## [1] 1 1
##
## $n
## [1] 1 2 3
##
## $S
## [1] "aa" "aa"
##
## $br
## [1] 1.095482 1.525263 2.000000
```

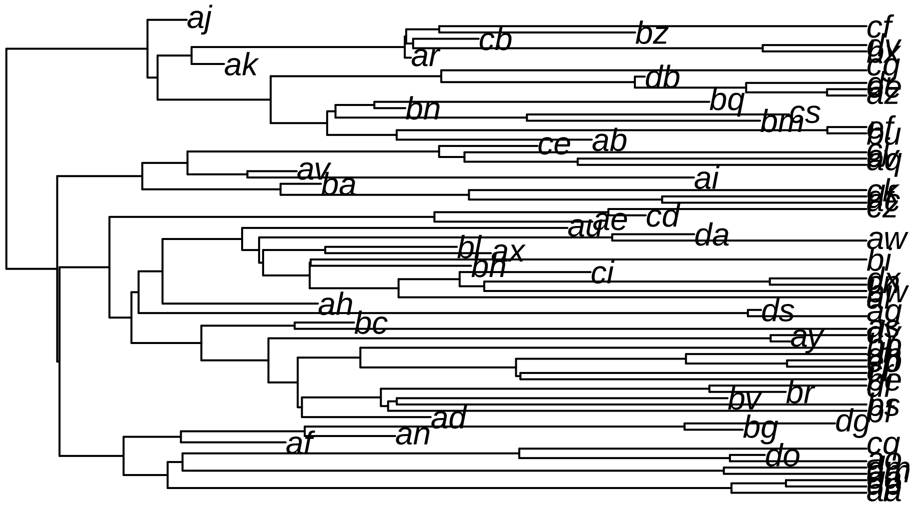
```
up = update.tree(s2,0.5,1.5)
up
```

```
## $wt
## [1] 0.50000000 0.59548244 0.40451756 0.02526264 0.47473736
##
## $E
## [1] 1 1 0 1
##
## $n
## [1] 1 2 3 2 3
```

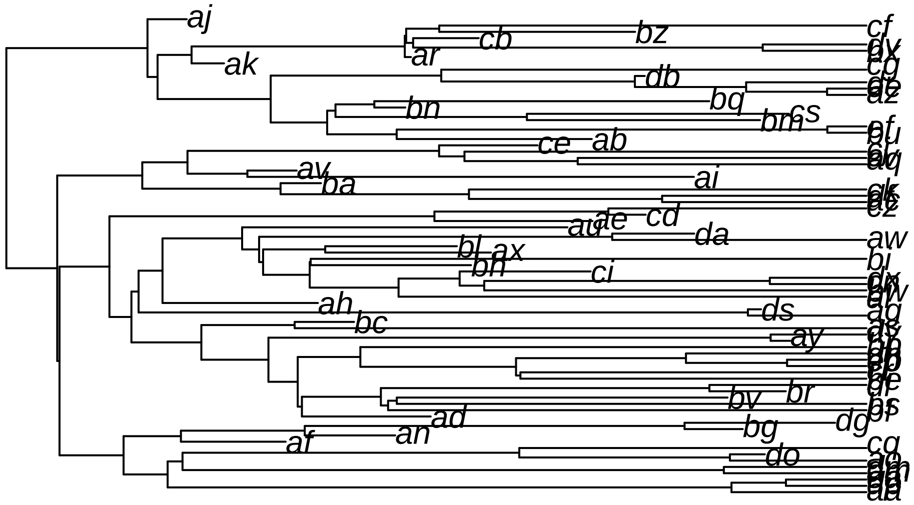
1. Doing update tree I realized that I should be more carefully in analysis when we lost crown time.
2. phylo2vectors really needs to take into account topology

more about consistency

```
s <- sim.tree()
plot(s$phylo)
```



```r
plot(vectors2phylo(s$tree))
```

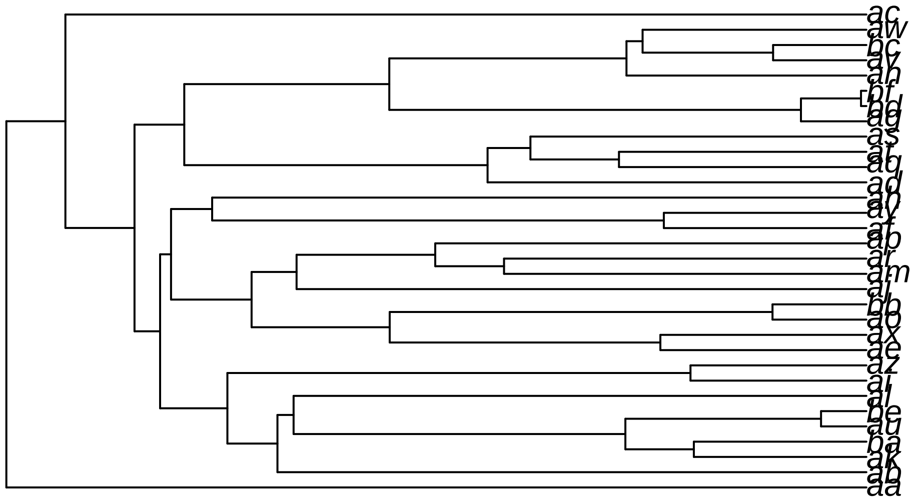


```r
all.equal(s$phylo,vectors2phylo(s$tree))
```

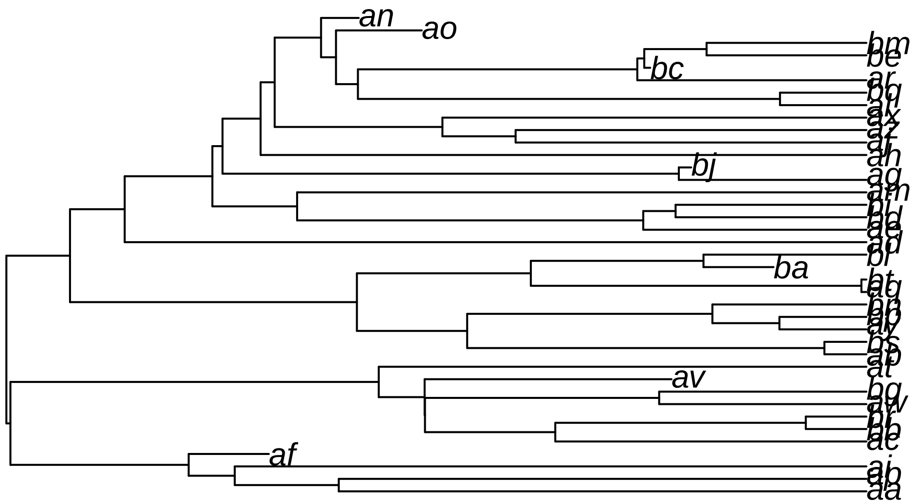
```
## [1] TRUE
```

The reconstruction algorithm

```r
s = sim.tree(ct=6)
plot(s$phylo.extant)
```



```
rec = rec.tree(tree=s$tree.extant,pars=c(0.8,0.1,40))
plot(vectors2phylo(rec))
```



```
s$tree.extant
```

```
## $wt
##  [1] 0.520024562 0.376312474 0.440344992 0.162930297 0.069802563
##  [6] 0.084344775 0.177608878 0.097022792 0.154247328 0.164701987
## [11] 0.102840481 0.019100064 0.590781785 0.002528972 0.290119065
## [16] 0.333377530 0.104880179 0.168239579 0.564265412 0.040787014
## [21] 0.006960564 0.102622237 0.113200338 0.022240266 0.169834369
## [26] 0.021267363 0.501251269 0.003713396 0.177518393 0.127974050
## [31] 0.254726892 0.034430135
##
## $E
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
##
## $n
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
## [24] 24 25 26 27 28 29 30 31 32
```

```r
rec
```

```
## $wt
##  [1] 0.082044932 0.028035271 0.409944359 0.376312474 0.440344992
##  [6] 0.162930297 0.069802563 0.084344775 0.177608878 0.056818678
## [11] 0.040204115 0.154247328 0.164701987 0.102840481 0.019100064
## [16] 0.124804516 0.006977283 0.005780751 0.137490296 0.296151939
## [21] 0.019577001 0.002528972 0.120048496 0.170070569 0.333377530
## [26] 0.104880179 0.168239579 0.564265412 0.040787014 0.006960564
## [31] 0.041161701 0.061460536 0.085317081 0.027883257 0.022240266
## [36] 0.085197504 0.084636865 0.021267363 0.039924189 0.420559779
## [41] 0.040767300 0.003713396 0.177518393 0.127974050 0.254726892
## [46] 0.034430135
##
## $E
##  [1] 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 0 1 0 1 1 1 1 1 1 1 1 1 1 0 1 0 1 1
## [36] 0 1 1 1 0 1 1 1 1 1
##
## $n
##  [1]  1  2  3  4  5  6  7  8  9 10  9 10 11 12 13 14 15 16 15 16 15 16 17
## [24] 18 19 20 21 22 23 24 25 24 25 24 25 26 25 26 27 28 27 28 29 30 31 32
```

let´s observe one estimation

```r
s = sim.tree(seed = 1)
tree = s$tree.extant
st = sim.srt(tree,pars=c(0.8,0.1,40),n_trees=100)
mle.st(st)
```

```
## [1]  0.85475693  0.09726184 39.07543002
```

How good is the last iteration of the MCEM algorithm?

```r
library(dmea)
n_sim = 537
n_trees = 10
MP = matrix(nrow=n_sim,ncol=3)
RP = matrix(nrow=n_sim,ncol=3)
p = proc.time()
for(i in 1:n_sim){
  est = sim.est(n_trees=n_trees,pars=c(0.8,0.1,40),seed=i)
  RP[i,] = est$real
  MP[i,] = est$est
}
print(proc.time()-p)
par_est_vis(P=MP,par=1,PR=RP)
par_est_vis(P=MP,par=2,PR=RP)
par_est_vis(P=MP,par=3,PR=RP)
```