dmeav02 Usage

Francisco Richter M.

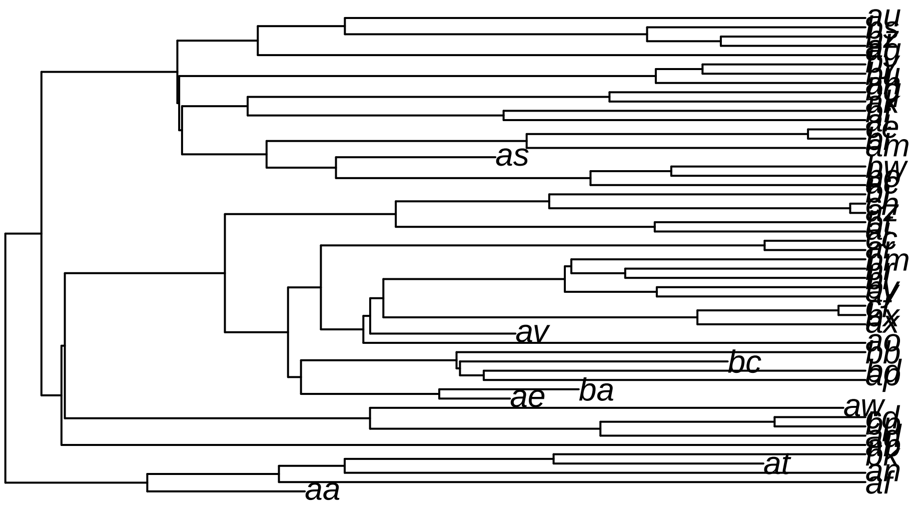
June 1, 2017

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] 0.1383254801 0.2459384824 0.1365898580 0.0228753368 0.5619208863
##  [6] 0.2048925967 0.0124894873 0.0195494426 0.2920043929 0.1550712881
## [11] 0.0692253155 0.0598664962 0.0841746497 0.0618916445 0.0885017936
## [16] 0.0261366915 0.1095168402 0.1026140228 0.0595337754 0.0013141998
## [21] 0.1257419791 0.0455745166 0.0009834259 0.0898544025 0.0848448115
## [26] 0.2961029214 0.1177084265 0.0236650533 0.1617383800 0.0795842388
## [31] 0.0557120165 0.0435096940 0.0376273734 0.0760805363 0.1537792038
## [36] 0.0299617416 0.0763848749 0.0445332509 0.0502264875 0.0805323362
## [41] 0.0674012442 0.0617125992 0.1072223898 0.1491431440 0.0521204394
## [46] 0.0072177146 0.0067955430 0.0986376771 0.1780730714 0.0350412684
## [51] 0.1247127825 0.0467368519 0.2435510580 0.0081451041 0.0680144717
## [56] 0.2275525540 0.2079541004 0.0332926129 0.0462735826 0.1038234400
##
## $E
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1
## [36] 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1
##
## $n
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 15 16 17 18 19 20 21
## [24] 22 23 24 25 26 27 28 27 28 27 26 27 28 29 30 31 30 31 32 33 34 35 36
## [47] 37 38 39 40 41 42 41 40 41 42 43 44 43 44
##
## $S
##  [1] "aa" "ab" "ab" "ad" "aa" "ac" "ac" "ac" "ae" "ai" "ag" "ac" "af" "ae"
## [15] "ae" "aa" "ao" "ac" "an" "al" "ao" "ad" "av" "ax" "aj" "ae" "ap" "ap"
## [29] "ap" "as" "ai" "ae" "av" "am" "az" "at" "ay" "bl" "ba" "ac" "ad" "ak"
## [43] "bl" "al" "aj" "ah" "ay" "bo" "ax" "bu" "al" "bc" "at" "ar" "bp" "bi"
## [57] "bx" "aw" "az"
##
## $br
##  [1] 0.1383255 0.3842640 0.5208538 0.5437292 1.1056500 1.3105426 1.3230321
##  [8] 1.3425816 1.6345860 1.7896573 1.8588826 1.9187491 2.0029237 2.0648154
## [15] 2.1533172 2.1794538 2.2889707 2.3915847 2.4511185 2.4524327 2.5781747
## [22] 2.6237492 2.6247326 2.7145870 2.7994318 3.0955347 3.2132432 3.2369082
## [29] 3.3986466 3.4782308 3.5339429 3.5774525 3.6150799 3.6911605 3.8449397
## [36] 3.8749014 3.9512863 3.9958195 4.0460460 4.1265784 4.1939796 4.2556922
## [43] 4.3629146 4.5120577 4.5641782 4.5713959 4.5781914 4.6768291 4.8549022
## [50] 4.8899434 5.0146562 5.0613931 5.3049441 5.3130892 5.3811037 5.6086563
## [57] 5.8166104 5.8499030 5.8961766 6.0000000
```

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

```
## $wt
##  [1] 0.1383254801 0.2459384824 0.1365898580 0.0228753368 0.5619208863
##  [6] 0.2048925967 0.0124894873 0.0195494426 0.2920043929 0.1550712881
## [11] 0.0692253155 0.0598664962 0.0841746497 0.0618916445 0.0885017936
## [16] 0.0261366915 0.1095168402 0.1026140228 0.0595337754 0.0013141998
## [21] 0.1257419791 0.0455745166 0.0009834259 0.0898544025 0.0848448115
## [26] 0.2961029214 0.1177084265 0.0236650533 0.1617383800 0.0795842388
## [31] 0.0557120165 0.0435096940 0.0376273734 0.0760805363 0.1537792038
## [36] 0.0299617416 0.0763848749 0.0445332509 0.0502264875 0.0805323362
## [41] 0.0674012442 0.0617125992 0.1072223898 0.1491431440 0.0521204394
## [46] 0.0072177146 0.0067955430 0.0986376771 0.1780730714 0.0350412684
## [51] 0.1247127825 0.0467368519 0.2435510580 0.0081451041 0.0680144717
## [56] 0.2275525540 0.2079541004 0.0332926129 0.0462735826 0.1038234400
##
## $E
##  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 0 0 1 1
## [36] 1 1 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 1 1 1 0 1
##
## $n
##  [1]  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 15 16 17 18 19 20 21
## [24] 22 23 24 25 26 27 28 27 28 27 26 27 28 29 30 31 30 31 32 33 34 35 36
## [47] 37 38 39 40 41 42 41 40 41 42 43 44 43 44
```

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

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

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

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

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

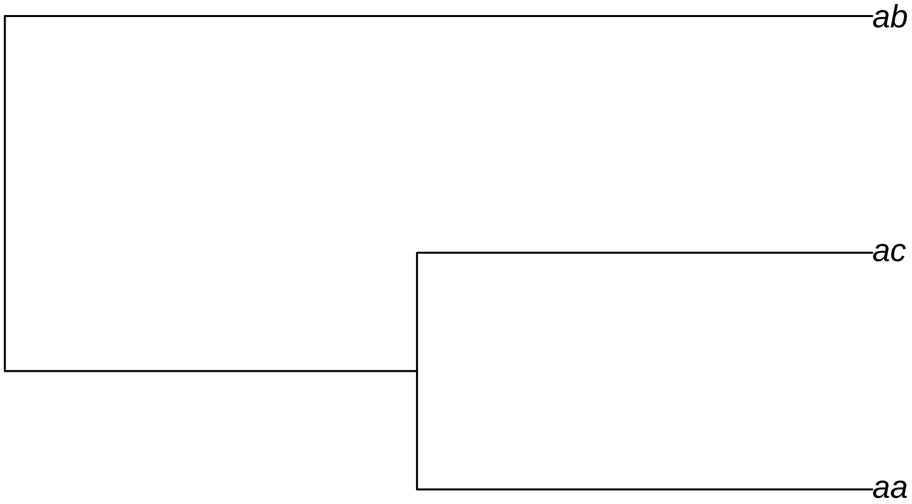
```
## [1] TRUE
```

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
```

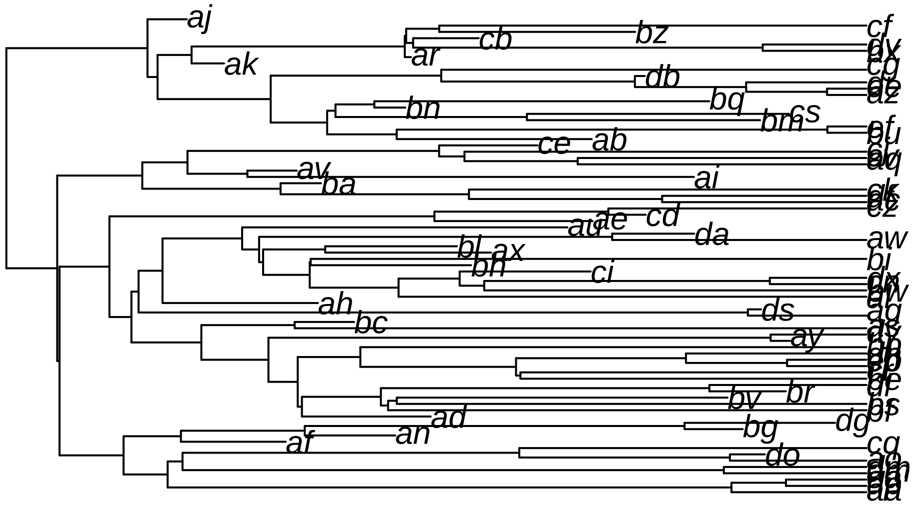
```
up = update.tree(s2,0.5,1.5)
up
```

```
## $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
```

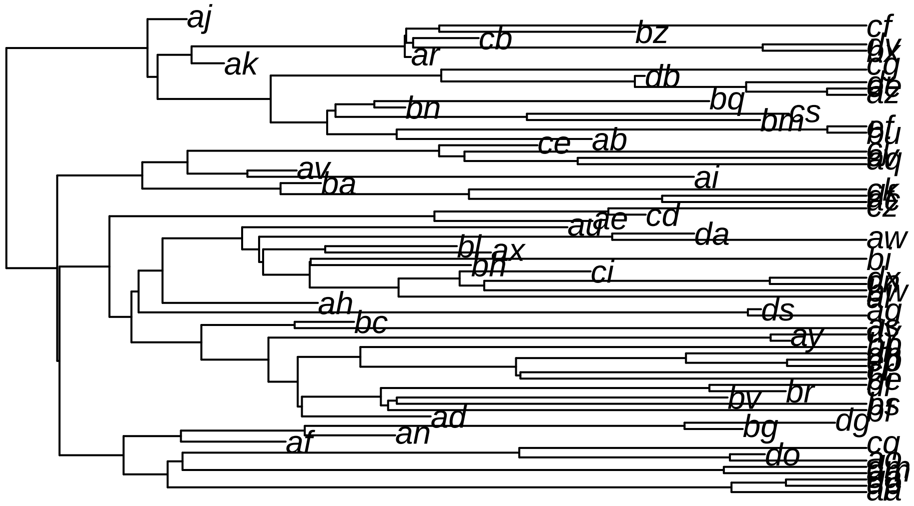
1. Doing update tree I realized that I should be more carefully in analysis when we lost crown time.
2. phylo2vectors really needs to take into account topology

more about consistency

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



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

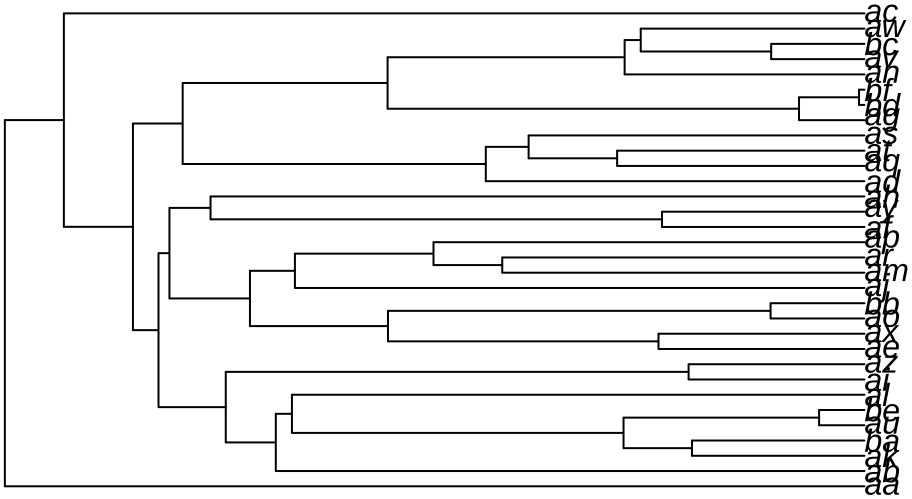


```
all.equal(s$phylo,vectors2phylo(s$tree))
```

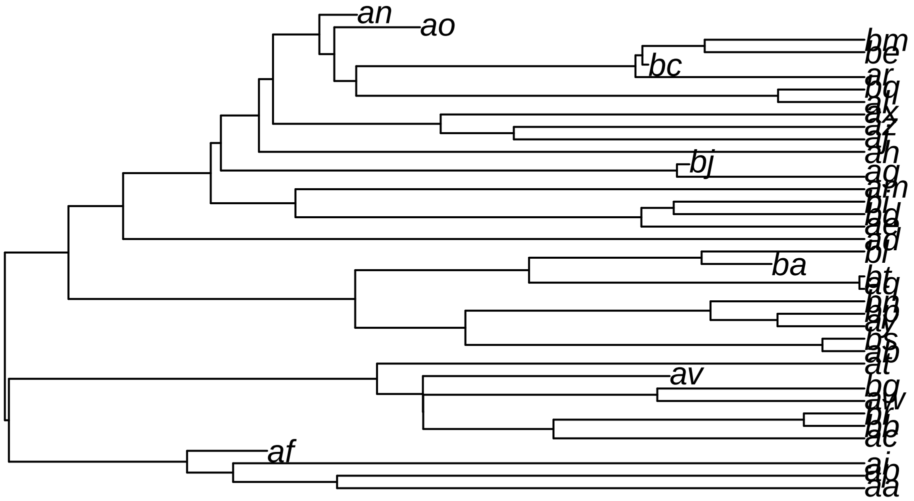
```
## [1] TRUE
```

The reconstruction algorithm

```
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
```

```
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
##
## $rprob
##  [1] 0.733842012 0.730828149 0.705777248 0.408065035 0.930232930
##  [6] 0.490659244 0.708220941 0.629890449 0.345575163 0.069316967
## [11] 0.789385326 0.375509867 0.327230250 0.477880465 0.865833863
## [16] 0.484890388 0.507389505 0.095139108 0.271290493 0.007786103
## [21] 0.952373724 0.978683634 0.175927445 0.215296907 0.046829629
## [26] 0.377046911 0.815311206 0.343783620 0.680361248 0.936534881
## [31] 0.095699728 0.563937302 0.045373159 0.773303541 0.815644699
## [36] 0.046174552 0.464394168 0.826324066 0.230087210 0.011032399
## [41] 0.697338138 0.968281646 0.221839803 0.347918886 0.564778395
## [46] 0.918708052
##
## $sprob
##  [1] 0.3395872 0.6639348 0.6851095 0.5362820 0.9006472 0.6737675 0.8398017
##  [8] 0.7560476 0.6263046 0.2543059 0.9024988 0.6822504 0.6260797 0.7456663
## [15] 0.9014340 2.1960416 2.3300786 0.3324403 1.7906276 0.1395670 0.6879176
## [22] 0.8576885 1.5121047 0.3807689 0.2906664 0.6576033 0.7147040 0.4611523
## [29] 0.6893029 0.7343755 0.7119265 0.8133144 0.6319128 0.9460766 0.9553725
## [36] 0.7486046 0.9192250 0.9801530 0.8575421 0.9753269 0.9786036 0.9981353
## [43] 0.9284271 0.9640358 0.9960116 0.9999532
##
## $et
##  [1] "speciation" "speciation" "nothing"    "nothing"    "nothing"
##  [6] "nothing"    "nothing"    "nothing"    "nothing"    "extinction"
## [11] "nothing"    "nothing"    "nothing"    "nothing"    "nothing"   
## [16] "speciation" "speciation" "extinction" "speciation" "extinction"
## [21] "nothing"    "nothing"    "speciation" "nothing"    "nothing"   
## [26] "nothing"    "nothing"    "nothing"    "nothing"    "nothing"   
## [31] "extinction" "nothing"    "extinction" "nothing"    "nothing"   
## [36] "extinction" "nothing"    "nothing"    "speciation" "extinction"
## [41] "nothing"    "nothing"    "nothing"    "nothing"    "nothing"   
## [46] "nothing"   
## 
## $logweight
## [1] -35.68686
```

let´s observe one estimation

```r
s = sim.tree(seed = 1)
mle.tree(s$tree)
```

```
## [1]  1.07613068  0.09936119 39.02277473
```

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

```
## [1]  0.85218549  0.09604088 39.09871958
```

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)
```