MCEM simulations

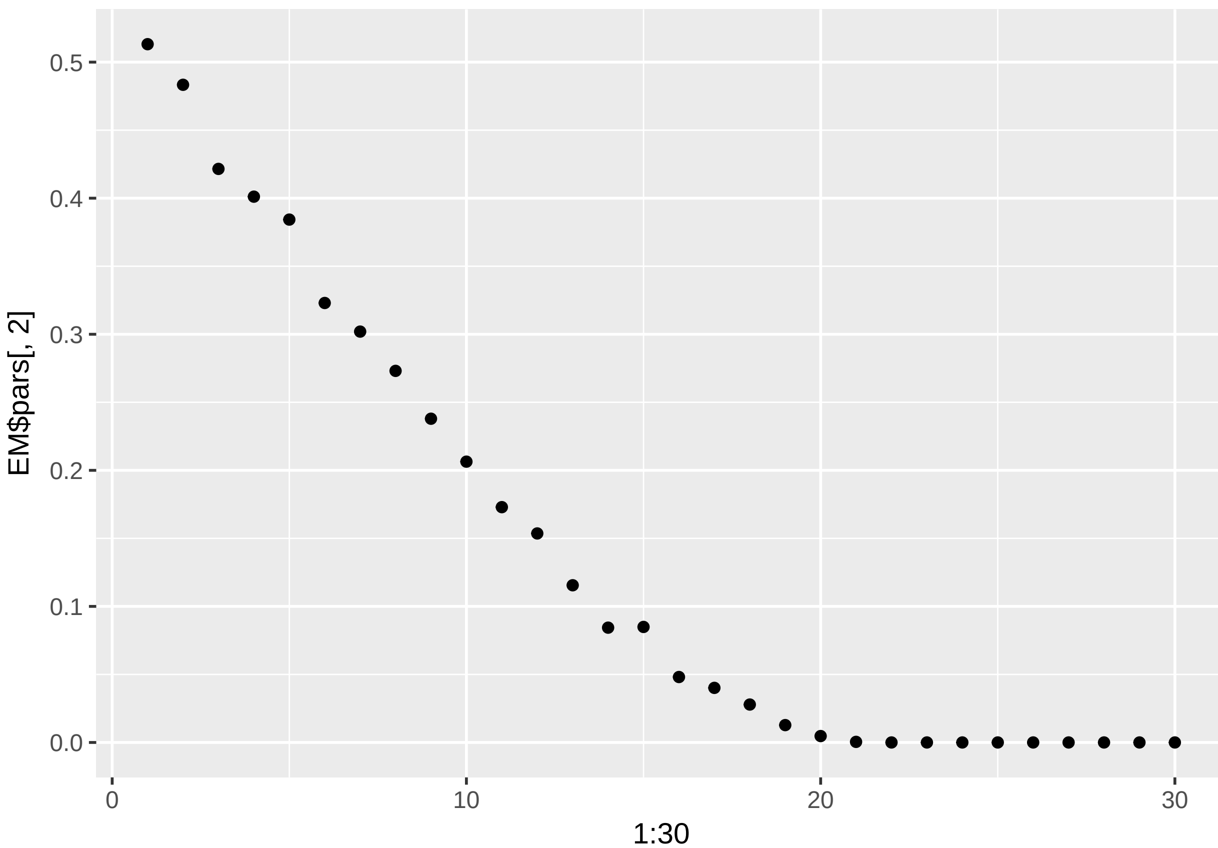
Simulations

```
seed = 3
s = sim_phyl(seed = seed)
s2 = phylo2p(drop.fossil(s$newick))
EM = EM_phylo(wt=s2$wt, init_par = c(1.8,0.6,80),n_trees = 10, parallel = T, impsam = T,tol=0.01)
```

```
## [1] "iteration # 1 :"
## [1]  1.8  0.6 80.0
## [1] "iteration # 2 :"
## [1]  1.6826445  0.5131967 84.4922020
## [1] "iteration # 3 :"
## [1]  1.2881443  0.4833367 83.7573343
## [1] "iteration # 4 :"
## [1]  1.1303295  0.4215241 84.2519790
## [1] "iteration # 5 :"
## [1]  1.0615753  0.4011336 79.5702025
## [1] "iteration # 6 :"
## [1]  1.0438218  0.3842778 75.8420451
## [1] "iteration # 7 :"
## [1]  0.8454106  0.3229822 81.2467222
## [1] "iteration # 8 :"
## [1]  0.7304166  0.3019546 72.9909105
## [1] "iteration # 9 :"
## [1]  0.9039366  0.2730956 66.8137113
## [1] "iteration # 10 :"
## [1]  0.8543810  0.2379271 64.5564159
## [1] "iteration # 11 :"
## [1]  0.7998869  0.2063764 55.5414296
## [1] "iteration # 12 :"
## [1]  0.7763548  0.1729484 52.0532026
## [1] "iteration # 13 :"
## [1]  0.8177746  0.1536267 46.2920135
## [1] "iteration # 14 :"
## [1]  0.8366668  0.1155314 44.8195607
## [1] "iteration # 15 :"
## [1]  0.66812378  0.08438512 44.55861710
## [1] "iteration # 16 :"
## [1]  0.70386661  0.08490554 38.69239864
## [1] "iteration # 17 :"
## [1]  0.58829123  0.04809249 38.56652279
## [1] "iteration # 18 :"
## [1]  0.51531251  0.04011941 37.32253484
## [1] "iteration # 19 :"
## [1]  0.48973490  0.02787306 37.44175420
## [1] "iteration # 20 :"
## [1]  0.40665261  0.01277044 39.18646867
## [1] "iteration # 21 :"
## [1]  0.374861377  0.004705835 40.491553076
## [1] "iteration # 22 :"
```

```
## [1] 3.502617e-01 4.478507e-04 4.174687e+01
## [1] "iteration # 23 :"
## [1] 3.485157e-01 2.887696e-06 4.184835e+01
## [1] "iteration # 24 :"
## [1] 3.485038e-01 4.594623e-17 4.184900e+01
## [1] "iteration # 25 :"
## [1] 3.485038e-01 7.845184e-18 4.184900e+01
## [1] "iteration # 26 :"
## [1] 3.485038e-01 7.845184e-18 4.184900e+01
## [1] "iteration # 27 :"
## [1] 3.485038e-01 7.845184e-18 4.184900e+01
## [1] "iteration # 28 :"
## [1] 3.485038e-01 7.845184e-18 4.184900e+01
## [1] "iteration # 29 :"
## [1] 3.485038e-01 7.845184e-18 4.184900e+01
## [1] "iteration # 30 :"
## [1] 3.485038e-01 7.845184e-18 4.184900e+01
```

```
qplot(1:30,EM$pars[,2])
```




```
EM$pars
```

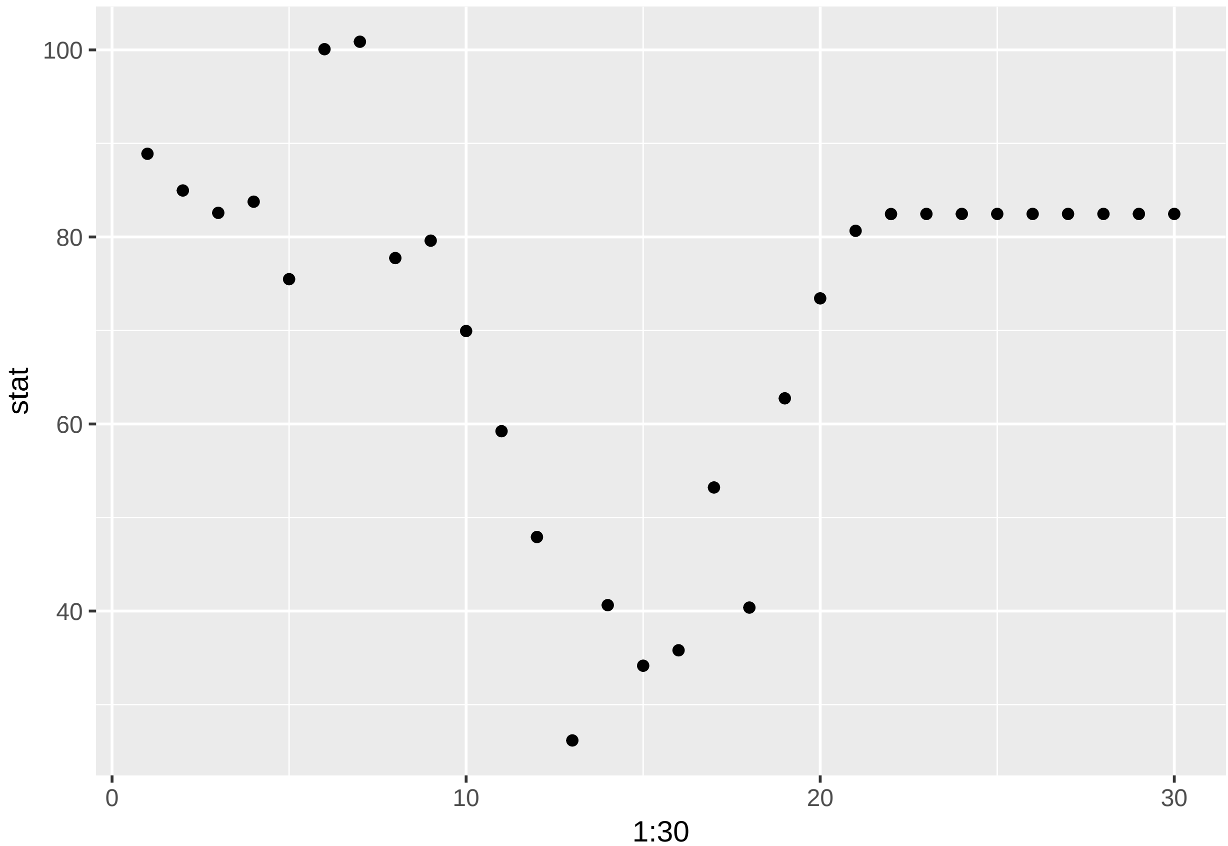
```
##            [,1]         [,2]     [,3]
##  [1,] 1.6826445 5.131967e-01 84.49220
##  [2,] 1.2881443 4.833367e-01 83.75733
##  [3,] 1.1303295 4.215241e-01 84.25198
##  [4,] 1.0615753 4.011336e-01 79.57020
##  [5,] 1.0438218 3.842778e-01 75.84205
```

```
##  [6,] 0.8454106 3.229822e-01 81.24672
##  [7,] 0.7304166 3.019546e-01 72.99091
##  [8,] 0.9039366 2.730956e-01 66.81371
##  [9,] 0.8543810 2.379271e-01 64.55642
## [10,] 0.7998869 2.063764e-01 55.54143
## [11,] 0.7763548 1.729484e-01 52.05320
## [12,] 0.8177746 1.536267e-01 46.29201
## [13,] 0.8366668 1.155314e-01 44.81956
## [14,] 0.6681238 8.438512e-02 44.55862
## [15,] 0.7038666 8.490554e-02 38.69240
## [16,] 0.5882912 4.809249e-02 38.56652
## [17,] 0.5153125 4.011941e-02 37.32253
## [18,] 0.4897349 2.787306e-02 37.44175
## [19,] 0.4066526 1.277044e-02 39.18647
## [20,] 0.3748614 4.705835e-03 40.49155
## [21,] 0.3502617 4.478507e-04 41.74687
## [22,] 0.3485157 2.887696e-06 41.84835
## [23,] 0.3485038 4.594623e-17 41.84900
## [24,] 0.3485038 7.845184e-18 41.84900
## [25,] 0.3485038 7.845184e-18 41.84900
## [26,] 0.3485038 7.845184e-18 41.84900
## [27,] 0.3485038 7.845184e-18 41.84900
## [28,] 0.3485038 7.845184e-18 41.84900
## [29,] 0.3485038 7.845184e-18 41.84900
## [30,] 0.3485038 7.845184e-18 41.84900
```

```r
time = proc.time()
stat=NULL
seed = 3
s = sim_phyl(seed = seed)
phylo1 = s$newick.extant
for(i in 1:30){
  expe = expectedLTT2(pars=EM$pars[i,])
  wt = c(expe$bt[1],diff(expe$bt))
  p = list(wt=wt,E=rep(1,length(expe$bt)),n=expe$Ex)
  phylo2 = p2phylo(p)
  ltt = ltt_stat(phylo1,phylo2)
  stat[i] = ltt
}
print(proc.time()-time)
```

```
##    user  system elapsed
## 128.528   0.028 128.633
```

```r
qplot(1:30,stat)
```

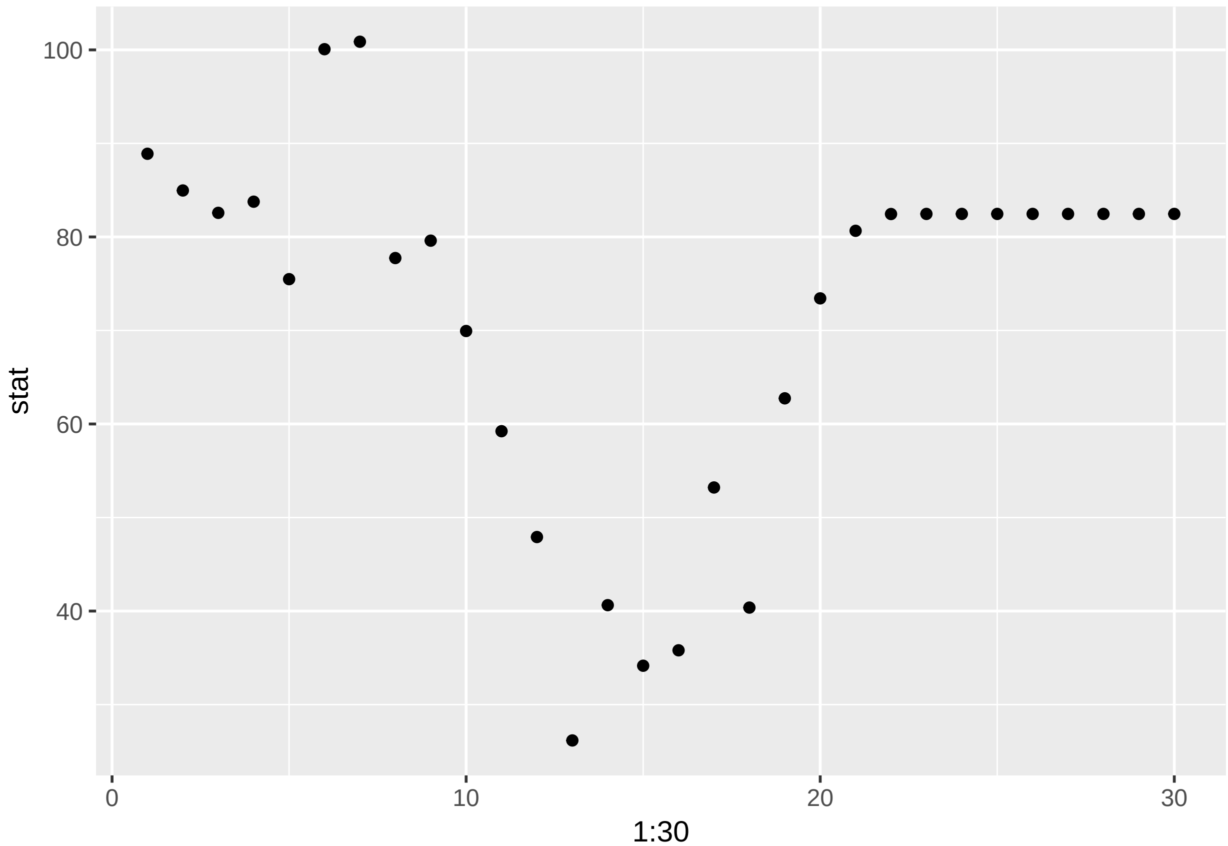


what if fo the expected thing with 100 trees?

```
time = proc.time()
stat=NULL
seed = 3
s = sim_phyl(seed = seed)
phylo1 = s$newick.extant
for(i in 1:30){
  expe = expectedLTT2(pars=EM$pars[i,],n_it=100)
  wt = c(expe$bt[1],diff(expe$bt))
  p = list(wt=wt,E=rep(1,(length(expe$bt)-1)),n=expe$Ex)
  phylo2 = p2phylo(p)
  ltt = ltt_stat(phylo1,phylo2)
  stat[i] = ltt
}
print(proc.time()-time)
```

```
##     user   system  elapsed
## 1166.820    0.128 1167.901
```

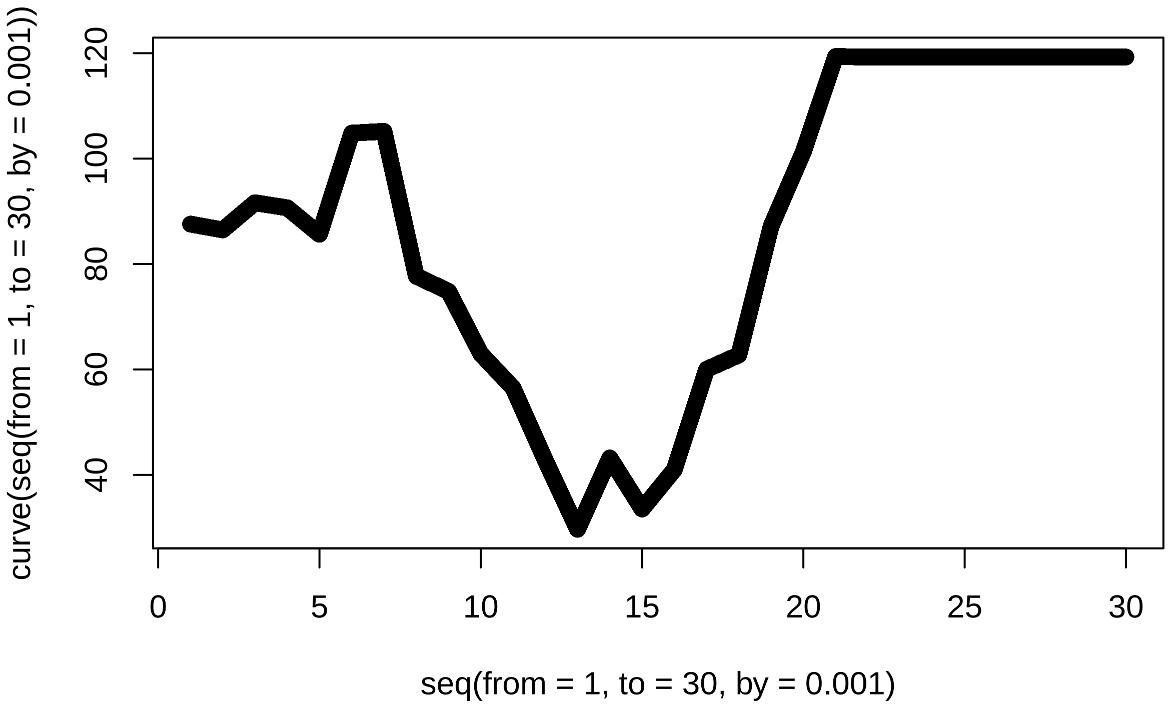
```
qplot(1:30,stat)
```




```
time = proc.time()
curve = approxfun(1:30,stat)
print(proc.time()-time)
```

```
##    user  system elapsed
##   0.000   0.000   0.003
```

```
plot(seq(from=1, to=30, by=0.001),curve(seq(from=1, to=30, by=0.001)))
```




```
Nmin = NNTbiomarker::argmin(curve(seq(from=1, to=30, by=0.001)))
```

wait, that was with importance sampling, now without that

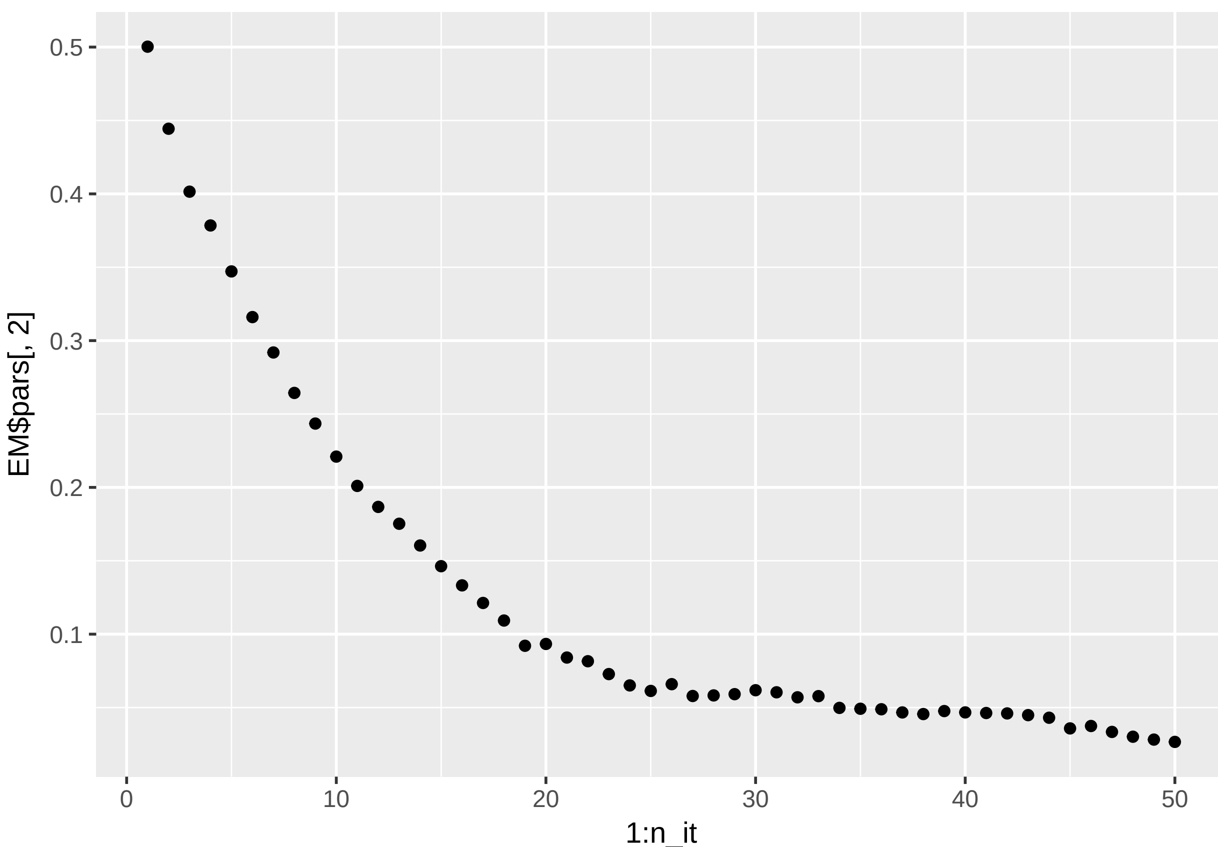
```
seed = 3
s = sim_phyl(seed = seed)
s2 = phylo2p(drop.fossil(s$newick))
n_it=50
EM = EM_phylo(wt=s2$wt, init_par = c(1.8,0.6,80),n_trees = 10, parallel = F, impsam = F,tol=0.01, n_it=
```

```
## [1] "iteration # 1 :"
## [1]  1.8  0.6 80.0
## [1] "iteration # 2 :"
## [1]  1.4800096  0.5002787 85.4243282
## [1] "iteration # 3 :"
## [1]  1.2579563  0.4443627 87.4208573
## [1] "iteration # 4 :"
## [1]  1.0996246  0.4014869 87.7871933
## [1] "iteration # 5 :"
## [1]  0.9785113  0.3784773 84.8077021
## [1] "iteration # 6 :"
## [1]  0.8628194  0.3471604 82.0986876
## [1] "iteration # 7 :"
## [1]  0.8105932  0.3160367 79.0540933
## [1] "iteration # 8 :"
## [1]  0.8144776  0.2919348 74.4551429
## [1] "iteration # 9 :"
## [1]  0.8233453  0.2643720 71.0613647
## [1] "iteration # 10 :"
## [1]  0.8230532  0.2434874 64.6091194
## [1] "iteration # 11 :"
## [1]  0.7719989  0.2209868 61.4260528
```

```
## [1] "iteration # 12 :"
## [1]  0.7750161  0.2010089 58.3438329
## [1] "iteration # 13 :"
## [1]  0.8166760  0.1867167 54.8565095
## [1] "iteration # 14 :"
## [1]  0.8474731  0.1752351 51.8729886
## [1] "iteration # 15 :"
## [1]  0.8523200  0.1604222 50.9711044
## [1] "iteration # 16 :"
## [1]  0.8376511  0.1463292 48.4262021
## [1] "iteration # 17 :"
## [1]  0.7494582  0.1332585 47.7805755
## [1] "iteration # 18 :"
## [1]  0.8183858  0.1212689 44.1677013
## [1] "iteration # 19 :"
## [1]  0.7576927  0.1092910 43.3418587
## [1] "iteration # 20 :"
## [1]  0.86880737  0.09208213 41.17101537
## [1] "iteration # 21 :"
## [1]  0.89782148  0.09333929 39.99732949
## [1] "iteration # 22 :"
## [1]  0.76019466  0.08407582 41.04701125
## [1] "iteration # 23 :"
## [1]  0.69651861  0.08157308 39.94524441
## [1] "iteration # 24 :"
## [1]  0.77862311  0.07281767 38.63477131
## [1] "iteration # 25 :"
## [1]  0.72226307  0.06510006 39.26715333
## [1] "iteration # 26 :"
## [1]  0.7999052  0.0613275 37.5236619
## [1] "iteration # 27 :"
## [1]  0.76207622  0.06593733 37.75888751
## [1] "iteration # 28 :"
## [1]  0.72799432  0.05787265 37.55283851
## [1] "iteration # 29 :"
## [1]  0.70949245  0.05827515 37.09537307
## [1] "iteration # 30 :"
## [1]  0.74499908  0.05913457 36.99650921
## [1] "iteration # 31 :"
## [1]  0.76418534  0.06181536 36.86620299
## [1] "iteration # 32 :"
## [1]  0.77313681  0.06039961 37.02573205
## [1] "iteration # 33 :"
## [1]  0.73639457  0.05698538 36.73684755
## [1] "iteration # 34 :"
## [1]  0.75679841  0.05778617 36.24479787
## [1] "iteration # 35 :"
## [1]  0.74312890  0.04977763 36.91567292
## [1] "iteration # 36 :"
## [1]  0.67200739  0.04918742 36.57351740
## [1] "iteration # 37 :"
## [1]  0.68956164  0.04884733 36.64304374
## [1] "iteration # 38 :"
## [1]  0.62557622  0.04668036 37.50158451
```

```
## [1] "iteration # 39 :"
## [1]  0.62341233  0.04558524 37.39430590
## [1] "iteration # 40 :"
## [1]  0.63447837  0.04761226 36.57212269
## [1] "iteration # 41 :"
## [1]  0.64432901  0.04670045 36.58572990
## [1] "iteration # 42 :"
## [1]  0.62454400  0.04628076 36.98183680
## [1] "iteration # 43 :"
## [1]  0.67079405  0.04601559 36.68866878
## [1] "iteration # 44 :"
## [1]  0.64004521  0.04484451 37.21283091
## [1] "iteration # 45 :"
## [1]  0.65502488  0.04307348 37.07714387
## [1] "iteration # 46 :"
## [1]  0.56370366  0.03579882 37.38282545
## [1] "iteration # 47 :"
## [1]  0.57251935  0.03743146 37.33666008
## [1] "iteration # 48 :"
## [1]  0.54565916  0.03340894 37.59725046
## [1] "iteration # 49 :"
## [1]  0.51310333  0.03014088 37.69768863
## [1] "iteration # 50 :"
## [1]  0.48820701  0.02817427 38.03782073
```

```r
qplot(1:n_it,EM$pars[,2])
```

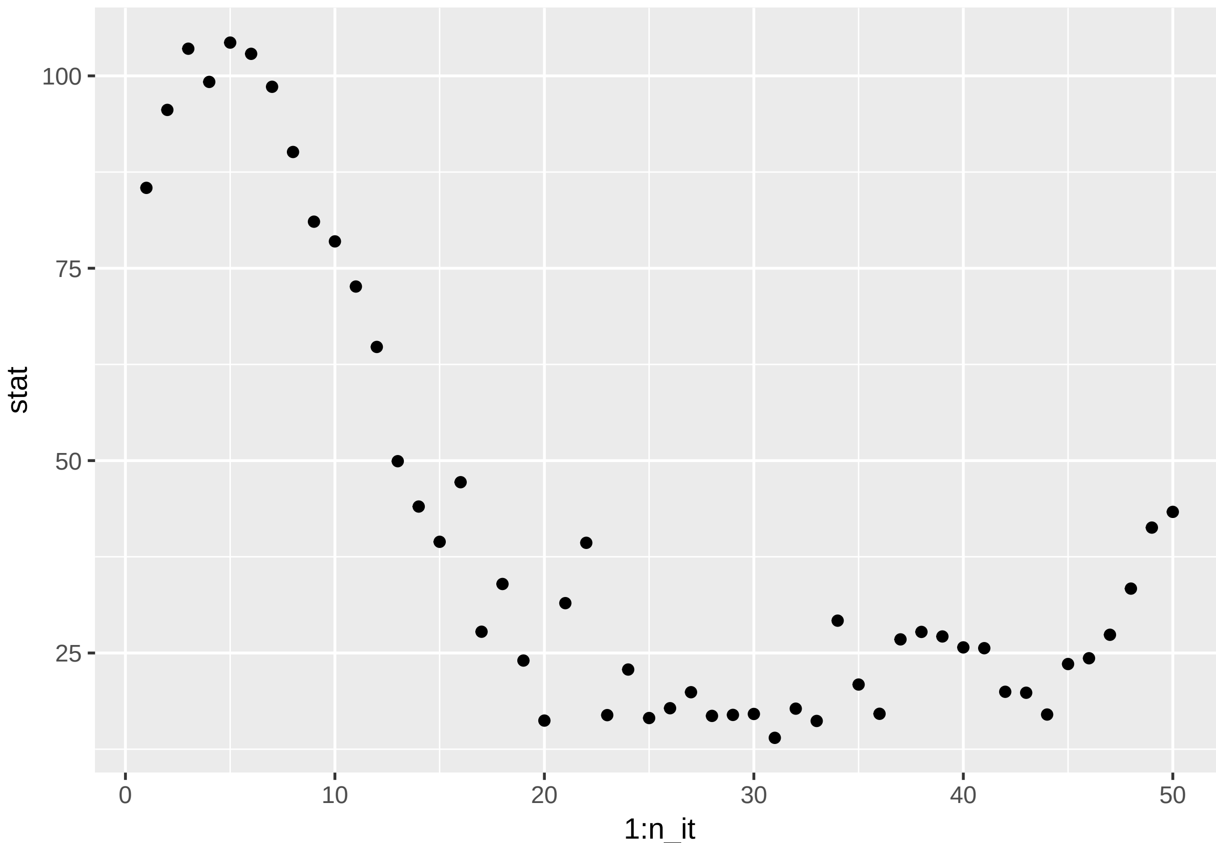


```
EM$pars
```

```
##             [,1]       [,2]     [,3]
##  [1,] 1.4800096 0.50027869 85.42433
##  [2,] 1.2579563 0.44436270 87.42086
##  [3,] 1.0996246 0.40148694 87.78719
##  [4,] 0.9785113 0.37847734 84.80770
##  [5,] 0.8628194 0.34716038 82.09869
##  [6,] 0.8105932 0.31603672 79.05409
##  [7,] 0.8144776 0.29193479 74.45514
##  [8,] 0.8233453 0.26437200 71.06136
##  [9,] 0.8230532 0.24348741 64.60912
## [10,] 0.7719989 0.22098683 61.42605
## [11,] 0.7750161 0.20100894 58.34383
## [12,] 0.8166760 0.18671668 54.85651
## [13,] 0.8474731 0.17523510 51.87299
## [14,] 0.8523200 0.16042222 50.97110
## [15,] 0.8376511 0.14632923 48.42620
## [16,] 0.7494582 0.13325847 47.78058
## [17,] 0.8183858 0.12126885 44.16770
## [18,] 0.7576927 0.10929098 43.34186
## [19,] 0.8688074 0.09208213 41.17102
## [20,] 0.8978215 0.09333929 39.99733
## [21,] 0.7601947 0.08407582 41.04701
## [22,] 0.6965186 0.08157308 39.94524
## [23,] 0.7786231 0.07281767 38.63477
## [24,] 0.7222631 0.06510006 39.26715
## [25,] 0.7999052 0.06132750 37.52366
## [26,] 0.7620762 0.06593733 37.75889
## [27,] 0.7279943 0.05787265 37.55284
## [28,] 0.7094925 0.05827515 37.09537
## [29,] 0.7449991 0.05913457 36.99651
## [30,] 0.7641853 0.06181536 36.86620
## [31,] 0.7731368 0.06039961 37.02573
## [32,] 0.7363946 0.05698538 36.73685
## [33,] 0.7567984 0.05778617 36.24480
## [34,] 0.7431289 0.04977763 36.91567
## [35,] 0.6720074 0.04918742 36.57352
## [36,] 0.6895616 0.04884733 36.64304
## [37,] 0.6255762 0.04668036 37.50158
## [38,] 0.6234123 0.04558524 37.39431
## [39,] 0.6344784 0.04761226 36.57212
## [40,] 0.6443290 0.04670045 36.58573
## [41,] 0.6245440 0.04628076 36.98184
## [42,] 0.6707940 0.04601559 36.68867
## [43,] 0.6400452 0.04484451 37.21283
## [44,] 0.6550249 0.04307348 37.07714
## [45,] 0.5637037 0.03579882 37.38283
## [46,] 0.5725193 0.03743146 37.33666
## [47,] 0.5456592 0.03340894 37.59725
## [48,] 0.5131033 0.03014088 37.69769
## [49,] 0.4882070 0.02817427 38.03782
## [50,] 0.4772370 0.02660999 38.03905
```

```
stat=NULL
seed = 3
s = sim_phyl(seed = seed)
phylo1 = s$newick.extant
for(i in 1:n_it){
  expe = expectedLTT2(pars=EM$pars[i,])
  wt = c(expe$bt[1],diff(expe$bt))
  p = list(wt=wt,E=rep(1,length(expe$bt)),n=expe$Ex)
  phylo2 = p2phylo(p)
  ltt = ltt_stat(phylo1,phylo2)
  stat[i] = ltt
}
```

```
qplot(1:n_it,stat)
```



by the way, what is the real mle?

```
s=sim_phyl(seed=3)
p <- subplex(par = c(2,0.2,60), fn = llik, n = s$n, E = s$E, t = s$wt)$par
p
```

```
## [1]  2.6666667  0.5333333 60.1666667
```

I am going to do the same but with 100 trees:

nice, I want to check for another tree

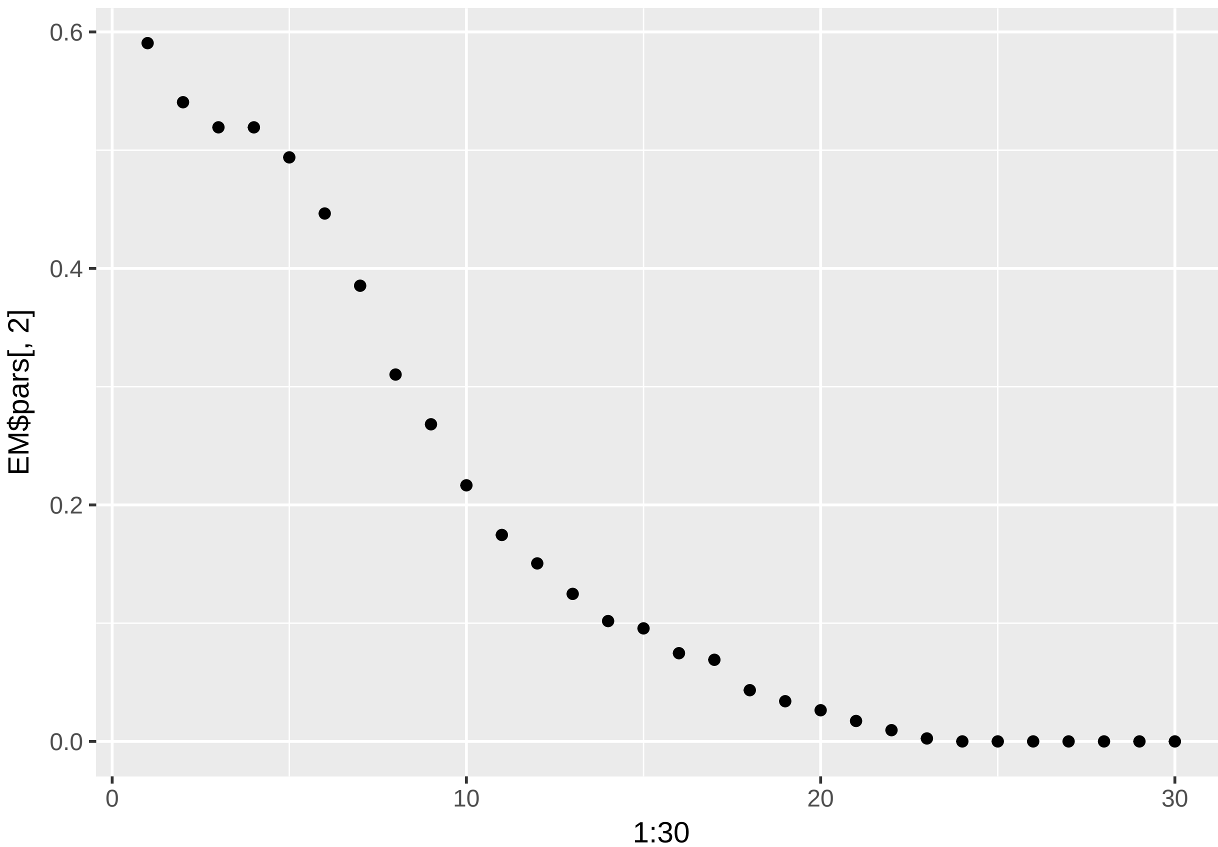
```
seed = 2
s = sim_phyl(seed = seed)
```

```r
s2 = phylo2p(drop.fossil(s$newick))
EM = EM_phylo(wt=s2$wt, init_par = c(1.8,0.6,80),n_trees = 10, parallel = T, impsam = T,tol=0.01)
```

```
## [1] "iteration # 1 :"
## [1]  1.8  0.6 80.0
## [1] "iteration # 2 :"
## [1]  1.5834220  0.5905331 77.6105193
## [1] "iteration # 3 :"
## [1]  1.4291909  0.5405425 81.3083535
## [1] "iteration # 4 :"
## [1]  1.212287  0.519334 82.479472
## [1] "iteration # 5 :"
## [1]  1.0917879  0.5193045 79.3334363
## [1] "iteration # 6 :"
## [1]  0.9702550  0.4939131 72.5812872
## [1] "iteration # 7 :"
## [1]  0.8297322  0.4464494 70.2456213
## [1] "iteration # 8 :"
## [1]  0.8176419  0.3853682 56.5270803
## [1] "iteration # 9 :"
## [1]  0.7947964  0.3102458 58.0017412
## [1] "iteration # 10 :"
## [1]  0.8538630  0.2681941 48.8146855
## [1] "iteration # 11 :"
## [1]  0.6967765  0.2165949 44.6041770
## [1] "iteration # 12 :"
## [1]  0.6986264  0.1745985 42.2884480
## [1] "iteration # 13 :"
## [1]  0.6600497  0.1505484 37.0518471
## [1] "iteration # 14 :"
## [1]  0.7681743  0.1247759 34.3403000
## [1] "iteration # 15 :"
## [1]  0.7514510  0.1017828 35.3987811
## [1] "iteration # 16 :"
## [1]  0.66315753  0.09560791 33.72809494
## [1] "iteration # 17 :"
## [1]  0.43471195  0.07460211 39.03937816
## [1] "iteration # 18 :"
## [1]  0.37336383  0.06904496 41.18437134
## [1] "iteration # 19 :"
## [1]  0.3405108  0.0433248 45.1742095
## [1] "iteration # 20 :"
## [1]  0.30355250  0.03399919 49.50185404
## [1] "iteration # 21 :"
## [1]  0.26353496  0.02636614 60.15299355
## [1] "iteration # 22 :"
## [1]  0.25293303  0.01729457 71.26147701
## [1] "iteration # 23 :"
## [1]  0.237063033  0.009514001 72.642360197
## [1] "iteration # 24 :"
## [1]  0.225212317  0.002516605 91.162910575
## [1] "iteration # 25 :"
## [1] 2.225726e-01 9.252049e-06 9.224708e+01
## [1] "iteration # 26 :"
```

```
## [1] 2.225563e-01 7.873884e-17 9.225862e+01
## [1] "iteration # 27 :"
## [1] 2.225563e-01 1.250353e-17 9.225870e+01
## [1] "iteration # 28 :"
## [1] 2.225562e-01 3.755188e-17 9.225877e+01
## [1] "iteration # 29 :"
## [1] 2.225562e-01 2.148811e-18 9.225882e+01
## [1] "iteration # 30 :"
## [1] 2.225562e-01 2.148811e-18 9.225883e+01
```

```r
qplot(1:30,EM$pars[,2])
```




```r
EM$pars
```

```
##              [,1]         [,2]     [,3]
##  [1,] 1.5834220 5.905331e-01 77.61052
##  [2,] 1.4291909 5.405425e-01 81.30835
##  [3,] 1.2122874 5.193340e-01 82.47947
##  [4,] 1.0917879 5.193045e-01 79.33344
##  [5,] 0.9702550 4.939131e-01 72.58129
##  [6,] 0.8297322 4.464494e-01 70.24562
##  [7,] 0.8176419 3.853682e-01 56.52708
##  [8,] 0.7947964 3.102458e-01 58.00174
##  [9,] 0.8538630 2.681941e-01 48.81469
## [10,] 0.6967765 2.165949e-01 44.60418
## [11,] 0.6986264 1.745985e-01 42.28845
## [12,] 0.6600497 1.505484e-01 37.05185
## [13,] 0.7681743 1.247759e-01 34.34030
```

```
## [14,] 0.7514510 1.017828e-01 35.39878
## [15,] 0.6631575 9.560791e-02 33.72809
## [16,] 0.4347119 7.460211e-02 39.03938
## [17,] 0.3733638 6.904496e-02 41.18437
## [18,] 0.3405108 4.332480e-02 45.17421
## [19,] 0.3035525 3.399919e-02 49.50185
## [20,] 0.2635350 2.636614e-02 60.15299
## [21,] 0.2529330 1.729457e-02 71.26148
## [22,] 0.2370630 9.514001e-03 72.64236
## [23,] 0.2252123 2.516605e-03 91.16291
## [24,] 0.2225726 9.252049e-06 92.24708
## [25,] 0.2225563 7.873884e-17 92.25862
## [26,] 0.2225563 1.250353e-17 92.25870
## [27,] 0.2225562 3.755188e-17 92.25877
## [28,] 0.2225562 2.148811e-18 92.25882
## [29,] 0.2225562 2.148811e-18 92.25883
## [30,] 0.2225562 2.148811e-18 92.25885
```

```
stat=NULL
seed = 2
s = sim_phyl(seed = seed)
phylo1 = s$newick.extant
for(i in 1:30){
  expe = expectedLTT2(pars=EM$pars[i,])
  wt = c(expe$bt[1],diff(expe$bt))
  p = list(wt=wt,E=rep(1,length(expe$bt)),n=expe$Ex)
  phylo2 = p2phylo(p)
  ltt = ltt_stat(phylo1,phylo2)
  stat[i] = ltt
}
```

```
 qplot(1:30,stat)
```

Simulations March 2017

Given a reconstructed phylogenetic tree

```
phylo <- sim_phyl(seed=3)$newick.extant
```

The new proposed method corresponds to the following steps

1. Set initial parameters for λ_0 and K

```
pars = c(2,80)
```

2. Given the parameters, find μ^* such that minimizes ltt_stat

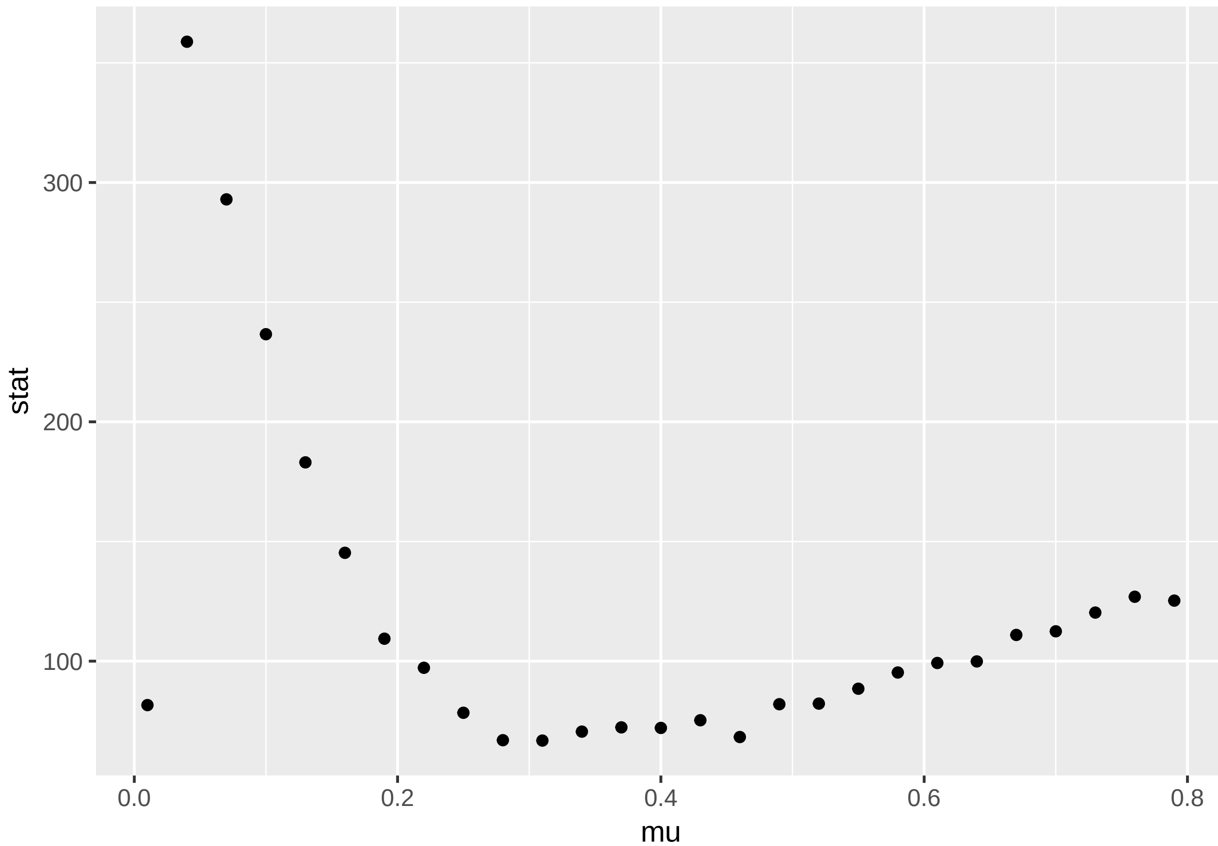
```
mu=0.2
# this does not work at all:
#pars = subplex(par = mu, fn = ltt_mu, phylo=phylo, prior_pars = pars)$par
#let´s visualize the ltt_mu curve
pp = proc.time()
mu = seq(0.01, 0.8, by=0.03)
stat=NULL
for(i in 1:length(mu)){
  stat[i] = ltt_mu(mu = mu[i], phylo = phylo, prior_pars = pars)
```

```
}
print(pp - proc.time())
```

```
##     user   system  elapsed
## -319.924   -0.044 -320.332
```

and

```
qplot(mu,stat)
```




```
nmu = mean(mu[stat < quantile(stat,0.05)])
nmu
```

```
## [1] 0.295
```

3. Once we have the new μ we can update λ and K fixing μ.
4. Once we have the new λ and K we go to step 2 to re-calculate μ
5. We repeat this algorithm until convergence.
