Constrast_mean_var

Qing Yu(Sabrina)

February 5, 2017

Goal of the study

In this report, our goal is to check the distribution of contrasts and check the correctness of covariance matrix generated by fixed root model.

simluation parameters

We used the lizard tree from the `phylolm` package and associated trait data on these lizard species (just the first trait, which is the first PC axis from a PCA), to find parameter values that are realistic. This trait was analyzed to estimate the shifts in trait evolution using the function `estimate_shift_configuration` from the `l1ou` package. The analysis takes about 2 minutes, so the results were saved in an R data file:

```
eModel <- estimate_shift_configuration(lizard$tree, lizard$Y)
save(eModel,  file = "eModel_2_5.RData")
```

Next, variables are set up to prepare the simulation.

```
load("eModel_2_5.RData")
data(lizard.tree, lizard.traits)
lizard <- adjust_data(lizard.tree, lizard.traits[,1])
```

```
## the new tree is normalized: each tip at distance 1 from the root.
## new Y: matrix of size 100 x 1
```

```
truealpha=eModel$alpha
y0=eModel$intercept
truetheta = y0 + l1ou:::convert_shifts2regions(eModel$tree,
                          eModel$shift.configuration, eModel$shift.values)
nShifts   = length(eModel$shift.configuration) # Total number of shifts
n_tips=length(eModel$tree$tip.label) # Total number of tips
sigma2=eModel$sigma2
shiftnode= eModel$tree$edge[eModel$shift.configuration,1]-n_tips # internal nodes with shift
othernode=(1:(n_tips-1))[-shiftnode] # Other nodes: those without a true shift
```

The tree has 100 tips and 8 nodes with a shift at one of its child edges, numbered $72, 51, 41, 63, 93, 61, 84, 18$ (`shiftnode`). At these shifts, the optimal value changed by: -2.91, -2.97, -5.5, -2.39, -2.61, -1.91, -3.7, -3.49 (shift values). The other parameters were estimated to be $\alpha = 0.606898$ (`truealpha`) and $\sigma^2 = 0.0625187$ and an "intercept" `y0` of 0.2488096. Together with the changes, this intercept gives the following optimal values `truetheta`, one for each edge: $\theta =$ 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, -2.36, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, -3.45, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, -1.66, -1.66, -1.66, -1.66, -4.57, -4.57, -4.57, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -1.66, -4.05, -1.66, -1.66, -1.66, 0.25, 0.25, 0.25, -2.72, -2.72, -2.72, -2.72, -2.72, -2.72, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, -2.72, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, 0.25, -5.25, 0.25, -3.24, 0.25.

We used this model to simulate new data using the `rTraitCont` function from the `phylolm` package. Below, `RE` is the result of the function `sqrt_OU_covariance`, which calculates the square-root of the phylogenetic covariance matrix with a recursive algorithm, which traverses the tree once. `C.IH` is the inverse square root of the phylogenetic covariance matrix, and `C.H` is the square-root of the phylogenetic covariance matrix. Finally, `YY` contains the contrasts at all nodes. These matrices and contrasts were obtained using the true value of α, the same value used to simulate the data. This is an ideal situation when α is known without error.

Simulation of Y using `rtraitCont`

```
n_sim=100000
Y_table=matrix(nrow=n_sim, ncol=n_tips, data=NA)
for (i in 1:n_sim) {
  Y  <- rTraitCont(eModel$tree, "OU", theta=truetheta,
                   alpha=truealpha,
                   sigma=sqrt(eModel$sigma2), root.value=y0)
  Y_table[i,]=Y
}
save(Y_table, file="Y_table_2_5.RData")
```

The covariance matrix generated by Y is the same as the covariance matrix generated by fixed root from the definition.

Checking whether mu from eModel and mean of tips from Y are significant different

```
load("Y_table_2_5.RData")
truemu=as.matrix(apply(Y_table,2,mean))
#cbind(eModel$mu,truemu)
head(eModel$mu)
```

```
##           [,1]
## [1,] 0.2488096
## [2,] 0.2488096
## [3,] 0.2488096
## [4,] 0.2488096
## [5,] 0.2488096
## [6,] 0.2488096
```

```
head(truemu)
```

```
##           [,1]
## [1,] 0.2485854
## [2,] 0.2478851
## [3,] 0.2483680
## [4,] 0.2493011
## [5,] 0.2482424
## [6,] 0.2481869
```

```
2*sqrt(eModel$sigma2/n_sim) ## 2*standard error
```

```
## [1] 0.001581375
```

```r
all.equal(eModel$mu,truemu, scale=1)
```

```
## [1] "Mean absolute difference: 0.0004177774"
```

```r
all.equal(eModel$mu,truemu,tolerance= 2*sqrt(eModel$sigma2/n_sim), scale=1)
```

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

mu from eModel and mean of tips from Y are not significant different

Compute the observed covariance matrix of Y

```r
cov_table=matrix(nrow=n_tips,ncol=n_tips,data=NA)
for (noderow in 1:n_tips){
  for(nodecol in 1:n_tips){
    sum=0
    for(sim in 1:n_sim){
      sum=(Y_table[sim,noderow]-eModel$mu[noderow])*(Y_table[sim,(nodecol)]-eModel$mu[nodecol])+sum
    }
    cov_table[noderow,nodecol]=sum/n_sim
  }
}
save(cov_table,file="Cov_table_2_5.RData")
```

Calculate the true covariance matrix with fixed root model

```r
tij=vcv(lizard.tree)  # the time spent on each edge
treeheight=max(tij)
dij=2*(treeheight-tij)
##vrandom=1/(2*truealpha)*exp(-truealpha*dij) # Sigma generated by OUrandomRoot model
vfix=1/(2*truealpha)*exp(-truealpha*dij)*(1-exp(-2*truealpha*tij))
```

Check agreement between observed and true covariance matrix of Y

```r
load("Cov_table_02_05.RData")
# matrix to be compared with
true_cov=vfix*eModel$sigma2
matequal <- function(x, y)
    all(abs(x-y)<=4*sqrt(1/n_sim)*eModel$sigma2*max(diag(vfix)))

matequal(true_cov,cov_table)
```

```
## [1] FALSE
```

```r
sqrt(1/n_sim)*eModel$sigma2*max(diag(vfix))
```

```
## [1] 0.0001144927
```

```r
all.equal(true_cov,cov_table,scale=1)
```

```
## [1] "Attributes: < Length mismatch: comparison on first 1 components >"
## [2] "Mean absolute difference: 9.341627e-05"
```

```r
all.equal(true_cov,cov_table,scale=1,tolerance=4*sqrt(1/n_sim)*eModel$sigma2*max(diag(vfix)))
```

```
## [1] "Attributes: < Length mismatch: comparison on first 1 components >"
```

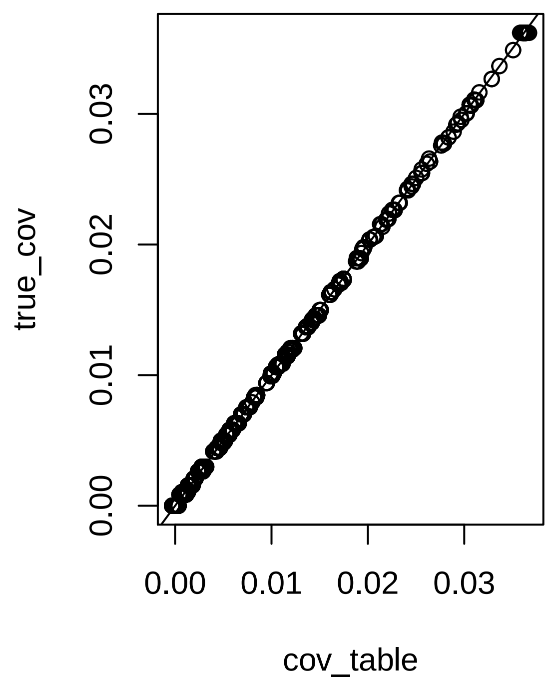
```r
true_cov[1:6,1:6]
```

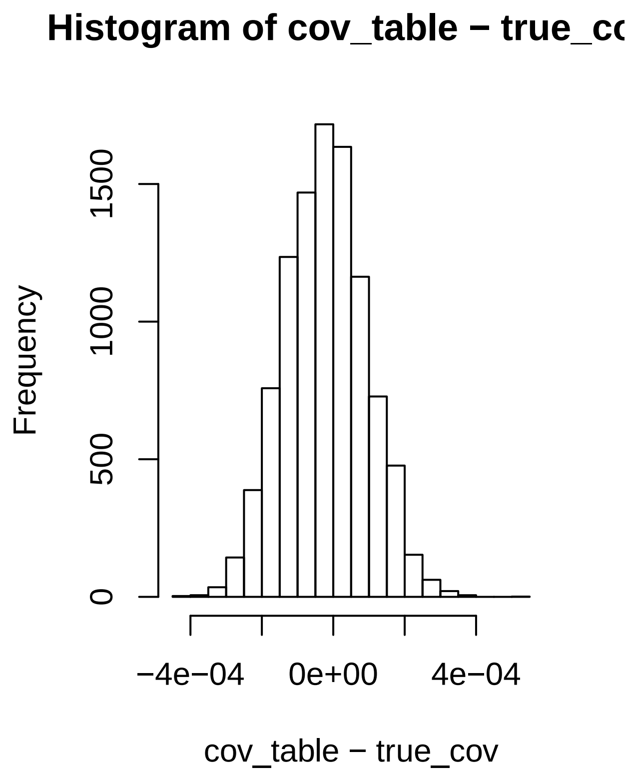
```
##                    ahli     allogus rubribarbus       imias     sagrei
## ahli        0.036205756 0.028639694 0.023190653 0.009937628 0.00630642
## allogus     0.028639694 0.036205756 0.023190653 0.009937628 0.00630642
## rubribarbus 0.023190653 0.023190653 0.036205756 0.009937628 0.00630642
## imias       0.009937628 0.009937628 0.009937628 0.036205756 0.00630642
## sagrei      0.006306420 0.006306420 0.006306420 0.006306420 0.03620576
## bremeri     0.006306420 0.006306420 0.006306420 0.006306420 0.02237467
##                bremeri
## ahli        0.00630642
## allogus     0.00630642
## rubribarbus 0.00630642
## imias       0.00630642
## sagrei      0.02237467
## bremeri     0.03620576
```

```r
cov_table[1:6,1:6]
```

```
##             [,1]        [,2]        [,3]        [,4]        [,5]
## [1,] 0.036729559 0.028899400 0.023311163 0.010051731 0.006421107
## [2,] 0.028899400 0.036206617 0.023221863 0.010086038 0.006436202
## [3,] 0.023311163 0.023221863 0.036095669 0.009966072 0.006466176
## [4,] 0.010051731 0.010086038 0.009966072 0.036252477 0.006567512
## [5,] 0.006421107 0.006436202 0.006466176 0.006567512 0.036003414
## [6,] 0.006327824 0.006347724 0.006456822 0.006468972 0.022192240
##             [,6]
## [1,] 0.006327824
## [2,] 0.006347724
## [3,] 0.006456822
## [4,] 0.006468972
## [5,] 0.022192240
## [6,] 0.035968925
```

```r
layout(matrix(1:2,1,2))
plot(cov_table,true_cov); abline(a=0, b=1)
hist(cov_table-true_cov)
```





The covariance matrix generated by Y is not significantly different from true covariance matrix of fixed root.

square root of covariance matrix

We first calculate it with `sqrt_OU_covariance`, then check that it is in agreement with the true covariance `vfix` calculated earlier.

```
REf = sqrt_OU_covariance(lizard.tree, alpha=truealpha,
                         root.model = "OUfixedRoot",normalize.tree.height=T ,
                         check.order=F, check.ultrametric=F)

Dtf   = t(REf$sqrtInvSigma)
Bf    = REf$sqrtSigma
all.equal(Dtf%*%Bf,diag(100), scale=1)
```

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

```
Ind_fix=Dtf%*%vfix%*%t(Dtf)
all.equal(Ind_fix,diag(100), scale=1)
```

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

"sqrt_OU_covariance" function caculates the square root of covariance matrix correctly.

Calculate Contrasts

```
contrast_table=matrix(nrow=n_sim, ncol=n_tips, data=NA)
for (i in 1:nrow(Y_table)) {
  contrast_table[i,]=Dtf%*%(Y_table[i,] - eModel$mu)
}
save(contrast_table, file="contrast_table_2_5.RData")
```

Check whether contrast mean is equal to 0

```
load('contrast_table_2_5.RData')
head(colMeans(contrast_table))
```

```
## [1]  0.0002936259  0.0005437081 -0.0011580854 -0.0014622974  0.0013248416
## [6] -0.0001489877
```

```
max(abs(colMeans(contrast_table)))
```

```
## [1] 0.001822459
```

```
colMeans(contrast_table)
```

```
##   [1]  2.936259e-04  5.437081e-04 -1.158085e-03 -1.462297e-03  1.324842e-03
##   [6] -1.489877e-04 -1.082437e-04  3.153187e-04  1.092892e-03 -3.949626e-04
##  [11] -6.906195e-04 -3.593490e-04 -4.397628e-04  1.502403e-03  6.004795e-04
##  [16]  1.532167e-04 -4.803428e-04  1.086476e-03 -1.124537e-03  3.234783e-04
##  [21] -2.055787e-04  1.240912e-03  5.511492e-04  1.923877e-04 -1.438383e-04
##  [26]  1.655792e-03 -6.682487e-04 -1.707682e-03 -1.133258e-03  3.482009e-04
##  [31]  3.939282e-04 -4.521101e-05  7.937092e-04 -7.300490e-04 -5.816772e-04
##  [36]  2.410313e-04  1.580529e-04  7.753134e-04  1.357171e-03 -3.434553e-04
##  [41] -1.062812e-03 -8.143168e-04  9.046964e-04  3.582746e-04 -1.822459e-03
##  [46] -7.308433e-04 -1.134198e-03  1.404343e-05  7.893330e-04 -3.609898e-04
##  [51]  9.241683e-04 -8.765625e-04 -4.132106e-04 -3.312742e-04  3.813810e-04
##  [56] -3.807294e-04 -7.968394e-04 -7.978669e-05  1.270336e-04 -4.466186e-04
##  [61]  7.032605e-04 -4.764006e-04  6.274901e-04 -1.207996e-03 -9.376294e-04
##  [66]  2.209574e-04 -2.858502e-04 -1.292109e-03 -5.827875e-04  1.752016e-04
##  [71]  2.847665e-04 -1.321285e-03 -8.194454e-04  6.150952e-05 -7.989488e-04
##  [76] -1.200301e-03  4.047343e-04  5.518316e-05  4.542254e-04  4.538062e-04
##  [81] -3.905190e-04  4.750033e-05  5.819572e-04  1.007289e-03  3.284101e-04
##  [86]  4.645462e-05  8.556082e-04  3.988658e-04 -2.834460e-04  3.224534e-04
##  [91] -4.945792e-04  1.423454e-03 -2.226035e-04 -1.192475e-03  7.468512e-04
##  [96]  2.900539e-04 -3.529061e-04 -1.232649e-03 -4.025506e-04 -2.648512e-04
```

The mean of all contrasts are all very close to zeros.

Check whether contrast variance is equal to sigma2

```
ss=function(x){
  sum(x^2)/length(x)
}
convar=apply(contrast_table,2,ss)
convar
```

```
##   [1] 0.06226439 0.06258969 0.06228206 0.06185538 0.06231495 0.06242140
##   [7] 0.06256224 0.06219234 0.06216682 0.06240147 0.06222383 0.06200548
##  [13] 0.06282155 0.06245075 0.06279975 0.06258275 0.06251883 0.06247997
##  [19] 0.06232777 0.06258758 0.06236507 0.06259616 0.06267557 0.06242355
##  [25] 0.06229275 0.06260808 0.06221274 0.06243247 0.06235870 0.06259370
##  [31] 0.06246769 0.06242847 0.06288057 0.06255914 0.06274826 0.06257278
##  [37] 0.06292527 0.06257065 0.06299845 0.06257913 0.06272465 0.06208273
##  [43] 0.06248409 0.06191161 0.06240520 0.06221114 0.06242529 0.06244330
##  [49] 0.06259906 0.06223723 0.06279803 0.06269686 0.06249820 0.06199016
##  [55] 0.06256578 0.06227071 0.06227814 0.06239336 0.06227895 0.06250814
##  [61] 0.06252155 0.06262676 0.06236870 0.06257509 0.06268641 0.06331554
##  [67] 0.06237474 0.06266784 0.06249402 0.06252180 0.06290999 0.06252153
##  [73] 0.06268941 0.06249738 0.06239644 0.06212852 0.06292058 0.06233895
##  [79] 0.06295462 0.06280390 0.06239385 0.06251193 0.06269901 0.06250260
##  [85] 0.06262733 0.06293786 0.06249215 0.06245859 0.06262924 0.06207798
##  [91] 0.06244999 0.06254036 0.06251645 0.06248630 0.06202160 0.06285781
##  [97] 0.06236344 0.06288143 0.06249279 0.06212581
```

```r
mean(convar)
```

```
## [1] 0.06249321
```

```r
eModel$sigma2
```

```
## [1] 0.06251866
```

The variance of contrasts is the same as sigma square.