# contrast variance by randomroot and fixedroot

*Cecile Ane and Qing Yu(Sabrina)*

*October 6th, 2016*

## Focus of the report

In this report, we use `Rstudio` with version 3.3.1 and `l1ou` package with version 1.28. our goal is to find out the reason that the covariance matrices generated by two root models are the same. Another problem is that variance of the contrast is not as expected, which I will explain later in the report. I checked the formula given in the `phylolm` package for `transf_branch_length` function and found out the fomulas were correctly used and generated the same result as expected. The tree should be ultrametric. t is the time from root to node. Under "OUrandomRoot", t is transformed to $\exp(-2*\alpha*(T-t))$, where T is now the mean root-to-tip distance. Under "OUfixedRroot", t is transformed to $\exp(-2*\alpha*(T-t))(1-exp(-2*\alpha*t))$ (as shown from the last session). So, the potential issue we need to deal with is that the covariance matrix we get from `sqrt_OU_covariance` function is not divided by $2*\alpha$. Mohammad and I are still working on fixing the bug.

## 1. Potential bug regarding to the same covariance matrix of the tree generated by two models

```
#truealpha=0.695680760276337
#sigma2=0.070414558609387
truealpha=0.6957
```

Let us look at the covariance matrix generated by two root models. `tij` represents the time spent from the root to the node. `dij` represents two times the time spent from the node to the tip. The height of the tree is the maximum number of `tij`. `vrandom` represents the covariance matrix generated by OUrandomRoot model. `vfix` represents the covariance matrix generated by OUfixedRoot model.

```
options(digits=4)
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))
```

Then, we can also get transpose of square root of the inverse matrix of covariance and the square root of the covariance matrix . 'Dtf' represents the transpose of square root of the inverse matrix of covariance generated by OUfixedRoot model. 'Bf' represents the square root of the covariance matrix generated by OUfixedRoot model.

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

Dtf  = t(REf$sqrtInvSigma)
Bf   = REf$sqrtSigma
```

'Dtr' represents the transpose of square root of the inverse matrix of covariance generated by OUrandomRoot model. 'Br' represents the square root of the covariance matrix generated by OUrandomRoot model.

```
REr = sqrt_OU_covariance(lizard.tree, alpha=truealpha,
                         root.model = "OUrandomRoot", normalize.tree.hight=T,
                         check.order=F, check.ultrametric=F)
Dtr  = t(REr$sqrtInvSigma)
Br   = REr$sqrtSigma
```

In the last version of `l1ou` (1.27), there were a bug due to the function `normalized_tree`. For the tree has a root edge, the root edge was not used in the funtion. In addition, the distance from the root to all tips did not include the length of the root.edge. Due to this bug, the square root of the covariance matrix generated by the two models are the same except for the last column which is the covariance of the root with other edges. Now, the bug has been fixed and the square root of the covariance matrix generated by the two models are different.

```
all.equal(Bf[,1:99],Br[,1:99])
```

```
## [1] "Mean relative difference: 0.1332"
```

## 2. Potential bug on variance of the contrast which should be sigma2.

We expect to see $D\Sigma D^T = I_n$.Since `D` is equal to $V^{-1/2}$ and $DD^T$ should be $V^{-1}$, so $D\Sigma D^T = I_n$.

Since this matrix is big, we just take a look at the diagonal of this matrix and to see if other number off the diagonal is 0.

Now, lets take a look at the $D\Sigma D^T$ generated by OUrandomRoot Model.

```
Ind_random=Dtr%*%vrandom%*%t(Dtr)
diag(Ind_random)
```

```
##   [1] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [11] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [21] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [31] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [41] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [51] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [61] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [71] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [81] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
##  [91] 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187 0.7187
```

```
1/(2*truealpha)
```

```
## [1] 0.7187
```

```
# Now, the diagonal of the matrix is 1.
diag(Ind_random*(2*truealpha))
```

```
##   [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 1 1 1
##  [36] 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 1 1
##  [71] 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
```

$\gamma = \sigma^2/(2\alpha)$ is the stationary variance: the variance of the OU process $(Y_t)_t$ when $t$ goes to $\infty$.In this case, the diagonal of the $D\Sigma D^T$ is equal to $1/(2\alpha)$, which is suppose to be 1. They become 1 when I multiply them by $(2\alpha)$.

**Check if the number off the diagonal is 0.**

```
Ind_rep=Ind_random
diag(Ind_rep)=NA
max(abs(Ind_rep),na.rm =T)
```

```
## [1] 1.316e-10
```

The numbers which are not on the diagonal are all zeros.

Furthermore, $BB^T$ should be equal to $\Sigma$ and$B = D^{-1^T}$ .However, they are different.

```
vrandom_sup=Br%*%t(Br)
all.equal(vrandom,vrandom_sup)
```

```
## [1] "Attributes: < Length mismatch: comparison on first 1 components >"
## [2] "Mean relative difference: 0.3914"
```

```
all.equal(t(solve(t(Dtr))),Br)
```

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

. Now, let us take a look at the $D\Sigma D^T$ generated by OUfixeDtroot Model.

```
Ind_fix=Dtf%*%vfix%*%t(Dtf)
diag(Ind_fix)
```

```
##   [1] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [11] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [21] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [31] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [41] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [51] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [61] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [71] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [81] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
##  [91] 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399 0.5399
```

```
(1-exp(-2*truealpha*treeheight))/(2*truealpha)
```

```
## [1] 0.5399
```

```
# Now, the diagonal of the matrix is 1.
diag(Ind_fix*(2*truealpha)/(1-exp(-2*truealpha*treeheight)))
```

```
##   [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 1 1 1
##  [36] 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 1 1
##  [71] 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
```

In this case, the diagonal of the $D\Sigma D^T$ is equal to $(1 - \exp(-2 * \alpha * T))/(2 * \alpha)$, which is suppose to be 1.

**Check if the number off the diagonal is 0.**

```
Ind_rep=Ind_fix
diag(Ind_rep)=NA
max(abs(Ind_rep),na.rm =T)
```

```
## [1] 9.888e-11
```

The numbers which are not on the diagonal are all zeros.

Furthermore, $BB^T$ should be equal to $\Sigma$ and $B = D^{-1^T}$. However, they are different.

```
vfix_sup=Bf%*%t(Bf)
all.equal(vfix,vfix_sup)
```

```
## [1] "Attributes: < Length mismatch: comparison on first 1 components >"
## [2] "Mean relative difference: 0.8521"
```

```
all.equal(t(solve(t(Dtf))),Bf)
```

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

We can see for covariance matrix generated by both model, their values do not match with $DD^T$ and $BB^T$. So, there may be bug inside `sqrt_OU_covariance` for generating `D` and `B`.