Distribution of phylogenetic contrasts under OU model

Cecile Ane and Qing (Sabrina) Yu

July 24, 2016

Goal of the study

In this report, my goal is to check the distribution of phylogenetic contrasts that are used in the bootstrap procedure in the `l1ou` package. We found that for some nodes, the contrasts' mean was far away from zero. We found the same variance for contrasts on all nodes, but with an unexpected value, so I investigated a few potential reasons for this. I did not identify a specific reason at this point, but the conclusion is that we may need to fix the contrast calculation for the bootstrap in the `l1ou` package.

I used code from the bootstrap procedure to simulate contrast values. Then I calculated the mean and standard deviation of the contrast values at each node. The conclusions above held for all nodes, whether nodes had a shift on child edge or not.

Simulation procedure

I 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). I analyzed this trait to estimate the shifts in trait evolution using the function `estimate_shift_configuration` from `l1ou` package. The analysis takes a little bit of time (about 2 minutes), so I saved the results in an R data file:

```
data(lizard.tree, lizard.traits)
lizard <- adjust_data(lizard.tree, lizard.traits[,1])
eModel <- estimate_shift_configuration(lizard$tree, lizard$Y)
save(eModel,  file = "eModel.RData")
```

The estimated shift configuration can be loaded quickly now:

```
load("eModel.RData")
```

Next I set up variables to prepare the simulation.

```
truealpha=eModel$alpha
y0=eModel$intercept
truetheta   = l1ou:::convert_shifts2regions(eModel$tree,
                                            eModel$shift.configuration, eModel$shift.values)+y0
nShifts   = length(eModel$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 # indices of internal nodes with shift
othernode=(1:(n_tips-1))[-shiftnode] # Other nodes: those without a true shift
n_sim=100000
```

The tree has 100 tip and 8 nodes that were detected with a shift, numbered $72, 51, 41, 63, 93, 61, 84, 18$. On these edges, the optimal values changed by (shift magnitudes):

$-0.8427272, -0.883897, -1.125207, -0.7715222, -0.8376818, -0.7117536, -1.5501462, -0.8134879$. The other parameters were estimated to be $\alpha = 0.606898$ and $\sigma^2 = 0.0625187$ and an "intercept" of 0.2488096. Together with the changes, this intercept gives the following optimal values, one for each edge:

$\theta =$ 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, -0.589, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, -1.301, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, -0.463, -0.463, -0.463, -0.463, -1.306, -1.306, -1.306, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -0.463, -1.234, -0.463, -0.463, -0.463, 0.249, 0.249, 0.249, -0.635, -0.635, -0.635, -0.635, -0.635, -0.635, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, -0.635, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, 0.249, -0.876, 0.249, -0.565, 0.249.

Now I 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.

```r
# simulate y based on this configuration
n_sim=100000
boot_table=matrix(nrow = n_sim, ncol=n_tips,data=0)
RE = sqrt_OU_covariance(eModel$tree, alpha=truealpha,
                        root.model = eModel$l1ou.options$root.model,
                        check.order=F, check.ultramteric=F)
C.IH  = t(RE$sqrtInvSigma)
C.H   = RE$sqrtSigma
for (i in 1:n_sim) {
  Y  <- rTraitCont(eModel$tree, "OU", theta=truetheta,
                 alpha=truealpha,
                 sigma=sqrt(eModel$sigma2), root.value=y0)
  YY    = C.IH%*%(Y - eModel$mu)
  boot_table[i,]=YY
}
save(boot_table, file="boot_table.RData")
```

The simulation of 10^5 data sets took 1 min and I saved all the results in `boot_table.RData` file.

Results: visualizations of contrast distributions

Contrasts on shift nodes

We first focus on the distribution, mean and standard deviations of contrasts associated with shift nodes, that is, nodes for which one child edge has a shift.

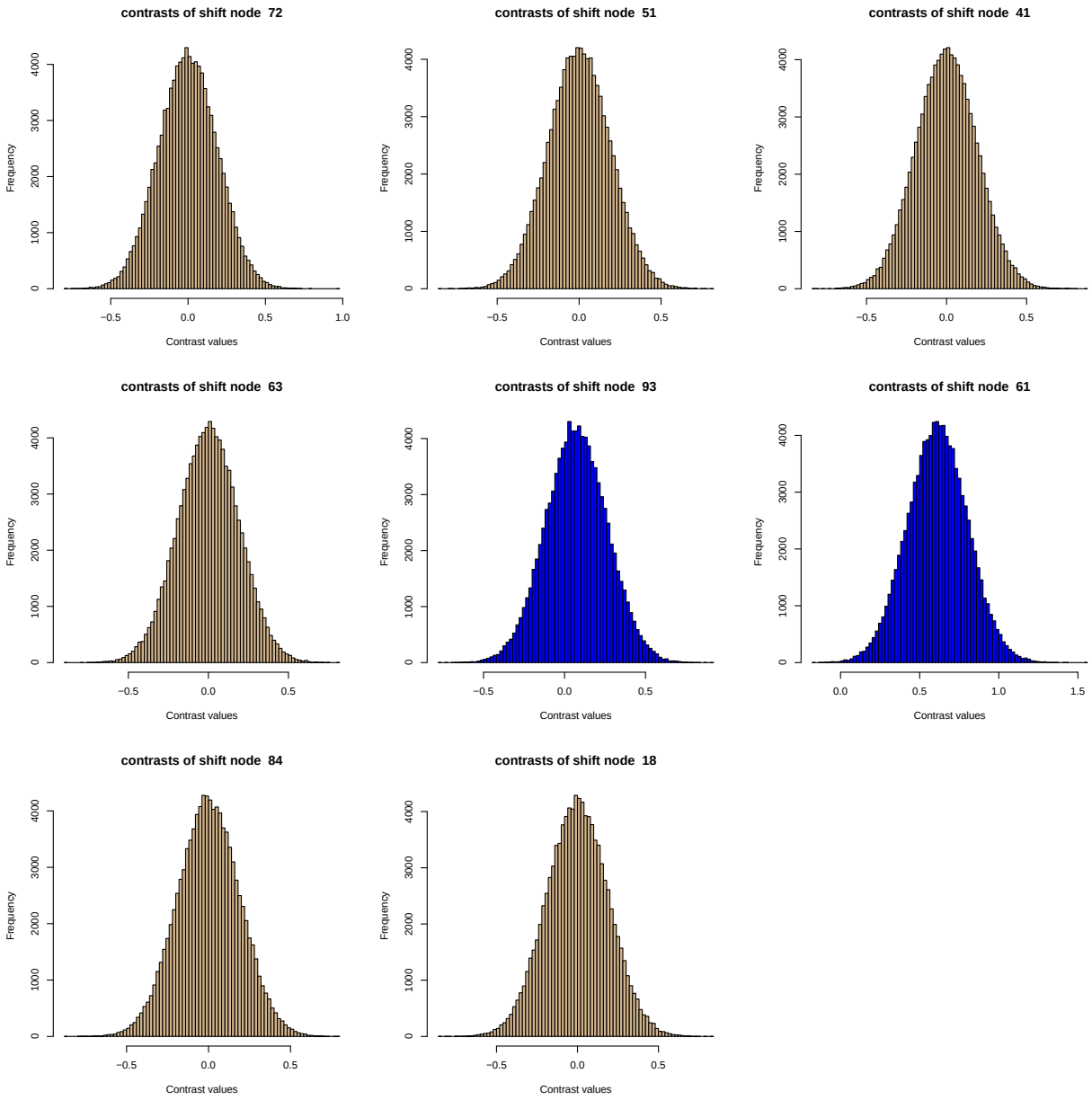
```r
# mean and variance of internal nodes with shifts
meanlist= colMeans(boot_table)[shiftnode] # mean contrasts of all shift nodes
options(digits=4)
meanlist
```

```
## [1] -0.0002017  0.0005686 -0.0003760  0.0004852  0.0736395  0.6129731
## [7]  0.0009355 -0.0001960
```

```
sdlist=apply(boot_table,2,sd)[shiftnode] # sd contrasts of all shift nodes
sdlist
```

```
## [1] 0.1893 0.1902 0.1907 0.1901 0.1904 0.1905 0.1903 0.1903
```

```
layout(matrix(1:9,ncol=3,nrow=3, byrow=TRUE))
for (j in shiftnode) {
  mycol="tan"
  if (j %in% c(93,61)){ mycol="blue"}
  hist(boot_table[,j],col=mycol,xlab="Contrast values", main=
         paste("contrasts of shift node " , j),breaks=100)
}
```



All the histograms are bell-shaped distribution with its peak close to 0 except for the 5th and 6th shiftnode (in blue). The mean of node 61 is 0.613, which is far away from 0 relative to its standard deviation. Accordingly, the proportion of contrast values below 0 is only 6.2×10^{-4} (instead of 0.5). The mean of node 93 is 0.0736, which is also far away from 0, but less so. The proportion of contrast values below 0, for all shift nodes, are:

```
apply(boot_table,2,function(v){sum(v<0)/n_sim})[shiftnode]
```

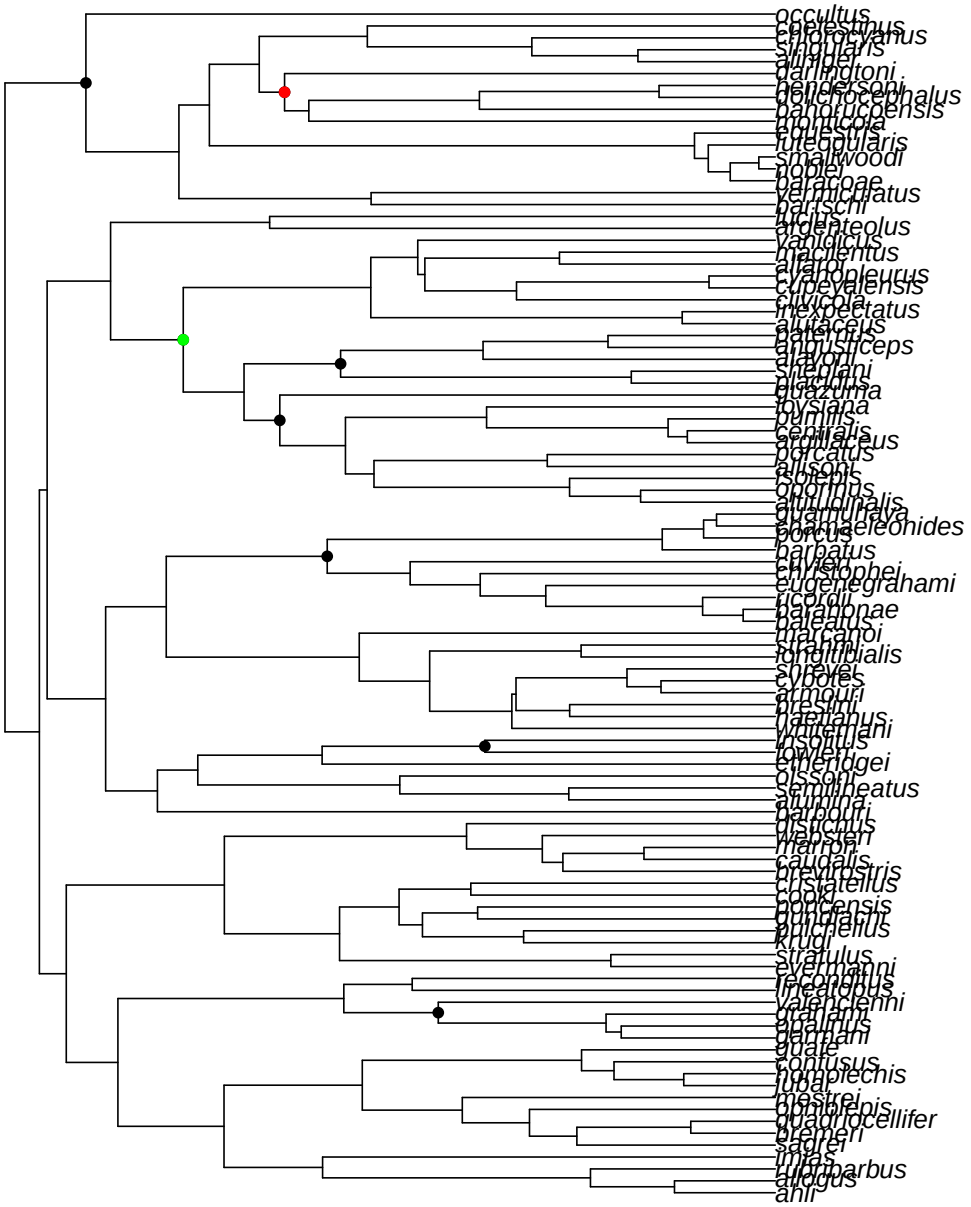
```
## [1] 0.50105 0.49910 0.50047 0.49683 0.34936 0.00062 0.49879 0.50025
```

Let's visualizing the 2 nodes with non-zero mean contrasts on the tree:

```
layout(matrix(1, 1))
plot(eModel$tree,main="Shift nodes on the tree with red and green
```

```
    representing the shiftnodes whose contrasts are far away from zero")
nodelabels("",shiftnode+n_tips,pch=16,frame="none")
nodelabels("",shiftnode[5]+n_tips,pch=16,col="red",frame="none") # Add n_tips
# for the indices of shiftnodes since we need to include the external node for the tree.
nodelabels("",shiftnode[6]+n_tips,pch=16,col="green",frame="none")
```

Shift nodes on the tree with red and green representing the shiftnodes whose contrasts are far away from zero



The two nodes with contrasts far away from 0 are close to the root and I do not understand the reason that their means are far away from zero.

Mean, SD and distribution of contrasts on non-shift nodes

```
# mean and variance of the other internal nodes,
# also proportion of contrasts that are <0. Expected: 0.5
pro0other=apply(boot_table,2,function(v){sum(v<0)/n_sim})[othernode] # The proportion of contrasts
#which is below zero for all the non-shift nodes.
apply(boot_table,2,sd)[othernode] # The standard deviations of contrasts for all non-shift nodes.
```

```
##  [1] 0.1906 0.1902 0.1907 0.1906 0.1907 0.1905 0.1907 0.1907 0.1904 0.1900
## [11] 0.1902 0.1900 0.1895 0.1908 0.1906 0.1896 0.1904 0.1908 0.1915 0.1906
## [21] 0.1900 0.1897 0.1892 0.1910 0.1898 0.1900 0.1901 0.1898 0.1894 0.1903
## [31] 0.1909 0.1906 0.1894 0.1905 0.1904 0.1905 0.1903 0.1900 0.1910 0.1907
## [41] 0.1908 0.1904 0.1907 0.1906 0.1902 0.1902 0.1908 0.1900 0.1904 0.1905
## [51] 0.1899 0.1902 0.1905 0.1903 0.1901 0.1903 0.1903 0.1902 0.1904 0.1902
## [61] 0.1903 0.1908 0.1902 0.1900 0.1898 0.1905 0.1901 0.1898 0.1904 0.1906
## [71] 0.1899 0.1900 0.1897 0.1904 0.1904 0.1906 0.1900 0.1909 0.1905 0.1907
## [81] 0.1900 0.1902 0.1901 0.1904 0.1902 0.1902 0.1903 0.1900 0.1900 0.1902
## [91] 0.1902
```

```
convar = mean(apply(boot_table,2,var))
round(colMeans(boot_table)[othernode], 3) # The mean of contrasts for all non-shift nodes
```

```
##  [1] -0.001  0.000 -0.001 -0.001  0.001  0.000  0.000 -0.001  0.000  0.000
## [11]  0.001 -0.876  0.000  0.001  0.000  0.000  0.000 -0.001  0.000  0.000
## [21]  0.000  0.000  0.000  0.001  0.000  0.000  0.001  0.000  0.000  0.000
## [31]  0.000 -0.001  0.001  0.000  0.000 -0.001 -0.410  0.000  0.000 -0.001
## [41] -0.001  0.000  0.000  0.000 -0.001  0.000  0.000 -0.001 -0.638  0.000
## [51]  0.000 -0.291  0.000  0.001 -0.001  0.000  0.000  0.000  0.000  0.001
## [61]  0.002  0.282  0.000 -0.001 -0.137  0.000  0.000  0.001  0.000 -0.550
## [71] -0.002  0.000  0.000  0.000 -0.523  0.000  0.206 -0.001 -0.667 -0.174
## [81] -0.102 -0.158 -0.247  0.845 -0.091  0.010  0.473 -0.785 -0.320 -0.348
## [91] -0.142
```

Mean seems way off for some edges - especially for some edges close to the tips. The shift nodes seem like cluster together. Mean seems correct (equals to 0) for other edges. SD seems incorrect, but sd is equal for all edges. The distribution of contrasts seems normal for edges.

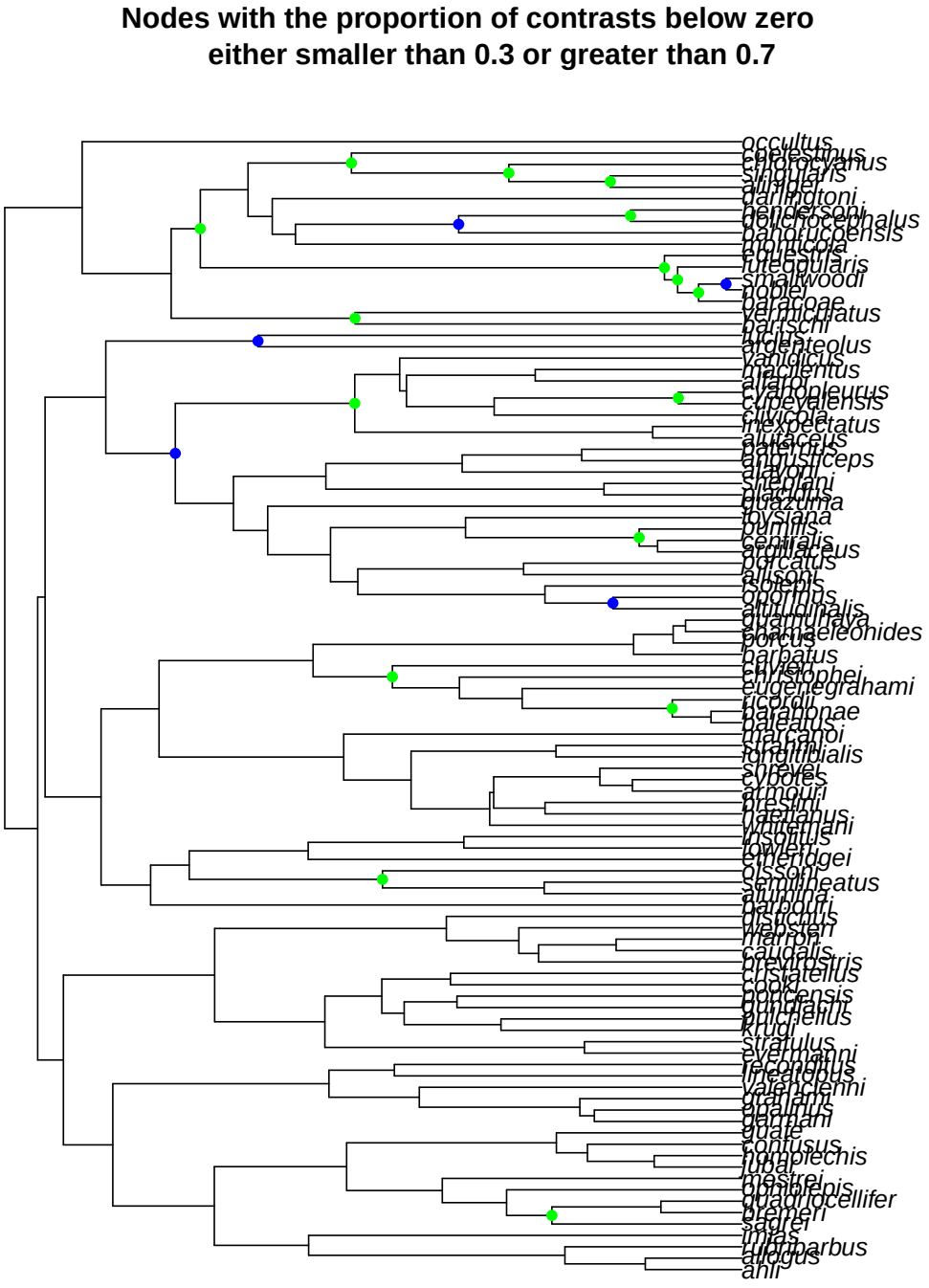
There are 0.1758 of othernodes who have high proportion of contrasts below zero. The standard deviation is all around `convar`.

We saw `convar=` 0.0362 which represents the mean of all variances of contrasts for non-shift nodes is different from `eModel$sigma2=` 0.0625. Most of the mean contrasts are around 0 but there are still quite a few mean contrasts which are far away from zero.

visualize weird nodes in the tree: both with or without shifts

```
prop.below0 = apply(boot_table,2,function(v){sum(v<0)/n_sim}) # The proportion of contrasts
# which is below zero for all nodes.
weird0.3 = which(prop.below0 < .3) # The indices of nodes with proportion
# below zero that is smaller than 0.3.
```

```r
weird0.7= which( prop.below0 > .7) # The indices of nodes with proportion
# below zero that is greater than 0.7.
plot(eModel$tree, main="Nodes with the proportion of contrasts below zero
     either smaller than 0.3 or greater than 0.7")
nodelabels("",weird0.3+n_tips,pch=16,col="blue",frame = "none")
nodelabels("",weird0.7+n_tips,pch=16,col="green",frame = "none")
mtext("Blue nodes with 30% or less of contrasts below zero.
      Green nodes with 70% or more of contrasts below zero.",side=1,pch=16,line=2)
```



Blue nodes with 30% or less of contrasts below zero.
Green nodes with 70% or more of contrasts below zero.

The plot shows that all the nodes which have the proportion of contrasts below 0 either greater than 0.7 or smaller than 0.3. Some nodes cluster together and are close to the tips. Others are separate from each other.

Why is the mean incorrect?

Trying to ind out where the problem is for the incorrect mean on some edges. potential problems:

1. bug or incorrect use of the simulation function
2. bug in calculation of contrasts
3. ?

Calculations to see if item 1 is a problem:

```
# checking if we have the correct mu
range(abs(
  rTraitCont(eModel$tree, "OU", theta=truetheta,
             alpha=100000, sigma=sqrt(eModel$sigma2),
             root.value = y0) - eModel$mu))
```

```
## [1] 4.350e-06 1.208e-03
```

```
range(abs(
  rTraitCont(eModel$tree, "OU", theta=truetheta,
             alpha=10000000, sigma=sqrt(eModel$sigma2),
             root.value = y0) - eModel$mu))
```

```
## [1] 4.456e-07 1.437e-04
```

When I change α from 100,000 to 10,000,000, the difference between Y and the true mu is getting smaller. The bigger the α is, the stronger the driving force to Y to be equal to the true mu.

```
mu = rep(y0,n_tips)
range(abs(
  rTrait(n=1, phy=eModel$tree, model="OU",
         list(optimal.value=y0, sigma2=eModel$sigma2,
              alpha=100000,
              ancestral.state=y0)) - mu))
```

```
## [1] 0.0000123 0.0014392
```

```
range(abs(
  rTraitCont(eModel$tree, "OU", theta=y0,
             alpha=100000, sigma=sqrt(eModel$sigma2),
             root.value = y0) - mu))
```

```
## [1] 0.0000122 0.0014415
```

I compare the Y that is generated by `rTrait` in phylolm package with Y that is generated by `rTraitCont` from l1ou package. I guess the reason that they are different is that rTraitCont has approximations.

Conclusion: NO. It is not the problem from the simulation because `rTraitcont` gave the correct mean.

Why is the variance incorrect?

Could it be a wrong choice in the model for the value at the root of the tree, when calculataing the square-root of the covariance matrix?

```
# checking if we have the correct transformation
RE = sqrt_OU_covariance(eModel$tree, alpha=truealpha,
                        root.model = eModel$l1ou.options$root.model)
newtree = reorder(eModel$tree, "post")
all(newtree$edge == eModel$tree$edge)
```

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

```
all(newtree$tip.label == eModel$tree$tip.label)
```

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

```
C.IH  = t(RE$sqrtInvSigma)
C.H   = RE$sqrtSigma

RE1 = sqrt_OU_covariance(eModel$tree, alpha=truealpha,root.model = "OUfixedRoot")
RE2 = sqrt_OU_covariance(eModel$tree, alpha=truealpha, root.model="OUrandomRoot")
all(RE1$sqrtInvSigma==RE2$sqrtInvSigma)
```

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

Interpreration

The variances of all contrasts are all roughly equal, around 0.0362. Why this particular value instead of the expected 0.0625?

$\gamma = \sigma^2/(2\alpha)$ is the stationary variance: the variance of the OU process (Y_t) when t is close to ∞. After a fixed time of evolution T, $\gamma(1 - \exp(-2\alpha T))$ is the variance of Y_t if the root value Y_0 is fixed with variance 0.

Is the contrast variance equal to $\sigma^2(1 - \exp(-2\alpha T))$ where $T = 1$ is the height of the tree?

```
(1-exp(-2*truealpha*1)) #  factor that differs between OU fixed root vs. random root
```

```
## [1] 0.7029
```

```
convar / eModel$sigma2
```

```
## [1] 0.5792
```

No.The values are different.

Is the variance of the contrasts, as currently calculated by the `l1ou` bootstrap function `sqrt_OU_covariance` equal to the stationary variance γ?

```
eModel$sigma2/ (2*truealpha)
```

```
## [1] 0.05151
```

```
convar
```

```
## [1] 0.03621
```

No. The values are different.

Is the contrast variance equal to $\gamma(1 - \exp(-2\alpha T))$?

```
eModel$sigma2 * (1-exp(-2*truealpha*1)) / (2*truealpha)
```

```
## [1] 0.03621
```

```
convar
```

```
## [1] 0.03621
```

Yes, almost. It looks like. This gives us an idea to fix the contrast calculation for the bootstrap in `l1ou` package.