Calculation of Association Statistics from Extant Species Only

August 11, 2022

Contents

Overview **2**
Data and Input Requirements 2

Analysis Walkthrough **2**
Reading in Trees and Calculating Relative Evolutionary Rates 2
Binary Traits 2
Continuous Traits 3
Categorical Traits 4

Conclusion **5**

This walkthrough demonstrates how to perform an RERconverge analysis using only the data at the tips of the tree, skipping the phylogenetic inference step of a typical RERconverge analysis. This walkthrough builds on existing RERconverge objects. First time users should first read the "RERconverge Analysis Walkthrough" vignette for information on installation, setup, and getting started.

Overview

Typically, we recommend including ancestral states in an RERconverge analysis because incorporating evolutionary information can strengthen the statistical power of the analysis. However, there may be phenotypes in which ancestral states are not as informative and could add noise to the results. In that case, we present a method for calculating association statistics between relative evolutionary rates and phenotype values using only the extant species in the tree.

Data and Input Requirements

The required inputs are as follows:

1. Phylogenetic trees of the same format described in the "RERconverge Analysis Walkthrough" vignette.
2. Species-labeled phenotype values

- The species labels MUST match the tree tip labels that will be used in `getAllResiduals` to calculate the relative evolutionary rates (RERs)
- a named vector of binary, continuous, or categorical trait values

Analysis Walkthrough

Reading in Trees and Calculating Relative Evolutionary Rates

Refer to the "RERconverge Analysis Walkthrough" vignette to learn how to read in gene trees using `readTrees` and calculate evolutionary rates using `getAllResiduals`.

Running the code below will read in some example trees that come with the RERconverge package that we will use for this walkthrough.

```
# check RERconverge is properly installed
library(RERconverge)

# find where the package is located on your machine
rerpath = find.package('RERconverge')

# read in the trees with the given file name
toytreefile = "subsetMammalGeneTrees.txt"
toyTrees=readTrees(paste(rerpath,"/extdata/",toytreefile,sep=""), max.read = 200)

# calculate the relative evolutionary rates with getAllResiduals
RERmat = getAllResiduals(toyTrees)
```

Binary Traits

We will continue our analysis for binary traits. First, we define foreground species for the marine binary phenotype and generate a named phenotype vector.

```
# define the foreground species
marineextantforeground = c("Walrus","Seal","Killer_whale","Dolphin","Manatee")
```

```
# make a phenotype vector for the species in the tree
# the phenotype values must be numeric (0 and 1 instead of TRUE and FALSE)
marinephenvals = rep(0, length(toyTrees$masterTree$tip.label))
names(marinephenvals) = toyTrees$masterTree$tip.label
# set the foreground species to true
marinephenvals[marineextantforeground] = 1
```

Finally, we calculate statistics using `getAllCorExtantOnly` which takes the following as input:

- `RERmat`: The RER matrix returned by `getAllResiduals`.
- `phenvals`: the named phenotype vector with names matching those used to calculated RERs in `getAllResiduals`.
- `method`: set to `"k"` for binary traits to calculate Kendall rank coefficients, `"p"` for continuous traits to use a Pearson correlation, and `"aov"` or `"kw"` for categorical traits to use an ANOVA or Kruskal Wallis test respectively.
- `min.sp`: The minimum number extant species in the gene tree for that gene to be included in the analysis.
- `min.pos`: The minimum number of extant foreground species in the gene tree for that gene to be included in the analysis.
- `winsorizeRER/winsorizetrait`: pulls the most extreme N values (default N=3) in both the positive and negative tails to the value of the N+1 most extreme value. This process mitigates the effect of extreme outliers before calculating correlations.

```
# set method to k to use a Kendall rank test since this is a binary phenotype
cors = getAllCorExtantOnly(RERmat, marinephenvals, method = "k")

# view the top results
head(cors[order(cors$P),])
```

```
##                 Rho  N           P     p.adj
## ANO2      0.3067162 58 0.004924203 0.3149587
## DNAH6    -0.2765208 62 0.008690358 0.3149587
## AK124326 -0.3464217 39 0.009807976 0.3149587
## ATP2A1    0.2870640 55 0.010455033 0.3149587
## BICC1    -0.2827014 56 0.010923423 0.3149587
## BDH1      0.2827014 56 0.010923423 0.3149587
```

For further analysis of the gene results, such as calculating functional enrichments, refer to the "RERconverge Analysis Walkthrough" vignette.

Continuous Traits

We will follow much the same steps for continuous traits as we did for binary traits. First, ensure that you followed the instructions above for reading in the gene trees and calculating relative evolutionary rates.

Next, we will load in some example data provided by RERconverge for the mammal body weight phenotype and calculate association statistics.

```
# load in the example data
data("logAdultWeightcm")

# set mehtod to p to use a Pearson correlation since this is a continuous phenotype
cors = getAllCorExtantOnly(RERmat, phenvals = logAdultWeightcm, method = "p")
```

```
# view the top results
head(cors[order(cors$P),])
```

```
##                 Rho  N            P       p.adj
## DNAH7    -0.5349841 59 1.269896e-05 0.002488997
## ADAMTSL4 -0.4827883 59 1.076584e-04 0.010550520
## DNAH6    -0.4010073 61 1.361267e-03 0.058734961
## AL833346 -0.4962212 38 1.532360e-03 0.058734961
## ATP2A1    0.4189829 54 1.614352e-03 0.058734961
## ADH7      0.4574178 44 1.798009e-03 0.058734961
```

For further analysis of the gene results, such as calculating functional enrichments, refer to the "RERconverge Analysis Walkthrough" vignette.

Categorical Traits

Once again, we will follow very similar steps for categorical traits as for both binary and continuous traits. First, ensure that you followed the instructions above for reading in the gene trees and calculating relative evolutionary rates.

Next, we will load in some example data provided by RERconverge for the sleep pattern phenotype and calculate association statistics.

```
# load in the example data
data("sleepPattern")

# set mehtod to kw to use a Kruskal Wallis test since this is a categorical phenotype
cors = getAllCorExtantOnly(RERmat, phenvals = sleepPattern, method = "kw")
```

Finally, we can view the results for all categories or for the pairwise comparisons between categories.

```
# the first element of cors is a table of association statistics for the Kruskal Wallis or ANOVA test a
all_categories_results = cors[[1]]
# view top results
head(all_categories_results[order(all_categories_results$P),])
```

```
##               Rho  N           P     p.adj
## ANKRD26  14.67979 57 0.002111780 0.3885675
## BMP10    12.31649 52 0.006374024 0.4369840
## ATR      11.42420 58 0.009639911 0.4369840
## B4GALNT2 11.39681 54 0.009762762 0.4369840
## ANKRD18B 10.60382 43 0.014072884 0.4369840
## ALKBH6   10.55971 54 0.014361252 0.4369840
```

```
# the second element of cors is a list of tables of pairwise comparisons between categories
pairwise_tests = cors[[2]]
names(pairwise_tests)
```

```
## [1] "CATHEMERAL - CREPUSCULAR" "CATHEMERAL - DIURNAL"
## [3] "CREPUSCULAR - DIURNAL"    "CATHEMERAL - NOCTURNAL"
## [5] "CREPUSCULAR - NOCTURNAL"  "DIURNAL - NOCTURNAL"
```

```
# view top results of pairwie test between diurnal and nocturnal species
head(pairwise_tests[[6]][order(pairwise_tests[[6]]$P),])
```

```
##                Rho           P     p.adj
## ANKRD26  -3.352167 0.004810895 0.8852047
## ATR       3.105270 0.011406305 1.0000000
```

```
## BMP10    -2.909463 0.021722999 1.0000000
## AMMECR1L -2.853599 0.025936248 1.0000000
## B4GALNT2 -2.726903 0.038359097 1.0000000
## ASIC3     2.689644 0.042916954 1.0000000
```

Conclusion

This concludes the walkthrough on how to calculate association statistics between phenotype and relative evolutionary rates with only the extant species in the tree.