CONGRUIFICATION: SUPPORT FOR TIME SCALING LARGE PHYLOGENETIC TREES

1. Getting started

Congruification is a nearly automated procedure that resolves all possible secondary calibrations of a *reference* tree that can be used to scale another *target* tree to units of time. Aside from having available a trustworthy timetree (referred to as the *reference*), congruification requires that the tree to be scaled (the *target*) have branch lengths in units of change (e.g., expected substitutions according to a molecular evolutionary model). So long as certain assumptions are reasonable to make for a given dataset, the procedure is quite forgiving in its requirement of sample overlap between the *reference* and *target*. Two imperfectly overlapping trees may be used in congruification if one supplies a linkage table defining the evolutionary relationship(s) between sampled tips of the two trees. For instance, a *reference* tree can be given for which tips represent families and a *target* that samples species from within these families. All higher taxa provided in the linkage table are assumed to be monophyletic, and proper use of a linkage table is illustrated below. The effect of violation of this monophyly assumption is not dire, only that fewer data from the *reference* tree will be exploited for divergence dates.

If working along from this vignette and using the R console, the user can copy and paste each fragment of code (preceded by a prompt arrow in the vignette, >). The user will need to ensure that the latest version GEIGER is installed (version $\geq$ 2): with a functional internet connection, this can be achieved by entering `install.packages("geiger")` at the R console. The package as well as its dependencies will be downloaded and installed locally.

The primary function used in this vignette is **congruify.phylo**. The example in the documentation pages associated with the **congruify.phylo()** function shows one instance of congruification requiring a linkage table. This example can be run in the R console with:

```
example(congruify.phylo)
```

If run, output from the example will appear at the R console. In this example, a family-level phylogeny of salamanders (Zhang and Wake 2008) is used to scale a species-level tree of the same group of amphibians. It is noteworthy that the function **example** is generally useful in exploring the features and functions of many packages. If a help file exists for a particular function and if the function is demonstrated within the documentation pages, the **example** function can be used to execute any examples found within the help pages. To find the help page associated with a function, precede the function name with ?. At the command prompt in the R console, try the following function that can aid taxonomically-informed node labeling of phylogenies:

```
?gbresolve
```

2. Assembling data

This vignette focuses on time-scaling the very large amphibian tree of Pyron and Wiens (2011). The tree that we'll use to provide divergences times (i.e., our time-scaled *reference* tree from Roelants et al. 2007) samples broadly from 152 species of frogs, salamanders, and caecilians. Both trees that we'll use for congruification are found within the GEIGER package as part of a data object, `amphibia`. We'll start by manipulating the *reference* tree in such a way as to maximize the amount of information mapped from *reference* to *target*.

```
> require(geiger)
> amph=get(data(amphibia)) ## load data and reassign to an object 'amph'
> rl=amph$roelants ## Roelants et al. (2007) chronogram
```

In order to exploit as many divergence events as possible, we should like to know the broadest lineage exemplified by each tip in the *reference* tree. In creating an exemplar tree, we will relabel the tips of the *reference* based on which major lineages are represented by single samples in the tree. This, for instance, has the effect of allowing the node corresponding to the most recent common ancestor of *Ambystoma* and *Dicamptodon* to be resolved in the *target* if *any* species within these genera are sampled within the *target*. Without relabeling, exactly the same species occurring in the *reference* as are found in the *target* must be present for this node to be matchable

Date: Id: congruify.Rnw 114 2012-12-30 14:22:36Z jeastman processed with geiger 1.99-3 in R version 3.0.0 (2013-04-03) on April 11, 2013.

between the trees. Relabeling prevents this. Certainly, the validity in relabeling tips rests on the assumption that these salamander genera, *Ambystoma* and *Dicamptodon*, are each monophyletic. In these cases, there is overwhelming support for monophyly. In order to generate the exemplar *reference* tree, we thus require accurate taxonomic information for all tips in that tree as well as those sampled in the *target*.

A taxonomic table may be user-supplied or it can be created on the fly. If we do not supply our own taxonomic table, the function **gbresolve** is used to resolve the taxonomy for the queried taxa. A downloaded and internally assembled update of that taxonomic resource can be forced through the argument `update=TRUE`. Let's run a quick example where we use **gbresolve** with the preexisting version of the taxonomic resource:

```
> gbresolve(c("Liua", "Hynobius", "Ranodon", "Andrias", "Siren"), rank="order", within="Caudata")
```

```
         genus      family             superfamily         suborder
Liua     "Liua"     "Hynobiidae"       "Cryptobranchoidea" ""
Hynobius "Hynobius" "Hynobiidae"       "Cryptobranchoidea" ""
Ranodon  "Ranodon"  "Hynobiidae"       "Cryptobranchoidea" ""
Andrias  "Andrias"  "Cryptobranchidae" "Cryptobranchoidea" ""
Siren    "Siren"    "Sirenidae"        ""                  "Sirenoidea"
         order
Liua     "Caudata"
Hynobius "Caudata"
Ranodon  "Caudata"
Andrias  "Caudata"
Siren    "Caudata"
```

This function has returned the NCBI taxonomic hierarchy for the five salamander genera that we queried. While GEIGER allows some flexibility with the format of taxonomic tables, functions within the package always expect the most exclusive groups to be leftmost in the taxonomic tables passed to a function. As seen above, missing values may be handled through `NA` or an empty space.

We will rely on the NCBI taxonomy database to generate our exemplar tree using the function **subset.phylo**. We will prefer the rank of genus in our exemplar tree.

We'll execute the relabeling of our `reference` phylogeny using a taxonomic table created from the NCBI taxonomy:

```
> trl=gbresolve(rl, within="Amphibia", rank=c("genus", "family"))$tax
> ex=subset(rl, tax=trl, rank="genus")
> ## show original and new tip labels of the exemplar phylogeny
> print(cbind(original=rl$tip.label[1:6], exemplar=ex$tip.label[1:6]))
```

```
     original                    exemplar
[1,] "Rhacophorus_malabaricus"   "Rhacophorus"
[2,] "Philautus_wynaadensis"     "Pseudophilautus"
[3,] "Mantidactylus_ulcerosus"   "Mantidactylus"
[4,] "Boophis_xerophilus"        "Boophis"
[5,] "Laliostoma_labrosa"        "Laliostoma"
[6,] "Meristogenys_kinabaluensis" "Meristogenys"
```

The output directly above shows us that we've relabeled the phylogeny of Roelants et al. (2007) based on tips that represent more expansive lineages (genera). The first column (`original`) is the set of original tip labels; the second column (`exemplar`) is the set of relabeled tips of the exemplar tree. For instance, we find that one of the true tree frogs (*Rhacophorus malabaricus*) is the only sampled member of the genus as is *Boophis xerophilus* in its own genus. The species *Philautus wynaadensis* has been assigned to the genus *Pseudophilautus*. Note that the exemplar tree has fewer tips than the original tree: where genera were determined to be monophyletic, all but a single representative for the genus were pruned from the tree. It was likely noticed that warnings were issued regarding the (non)-monophyly of the genus *Ichthyophis* and the lack of information for *Uraeotyphlus malabaricus*. In such cases, relevant taxa are left unpruned and the original tip labels are left intact.

We'll plot the exemplar phylogeny just created.

```
> plot.phylo(ladderize(ex, right=FALSE), cex=0.25, label.offset=3, x.lim=c(-10, 450))
> axisPhylo(cex.axis=0.75)
```

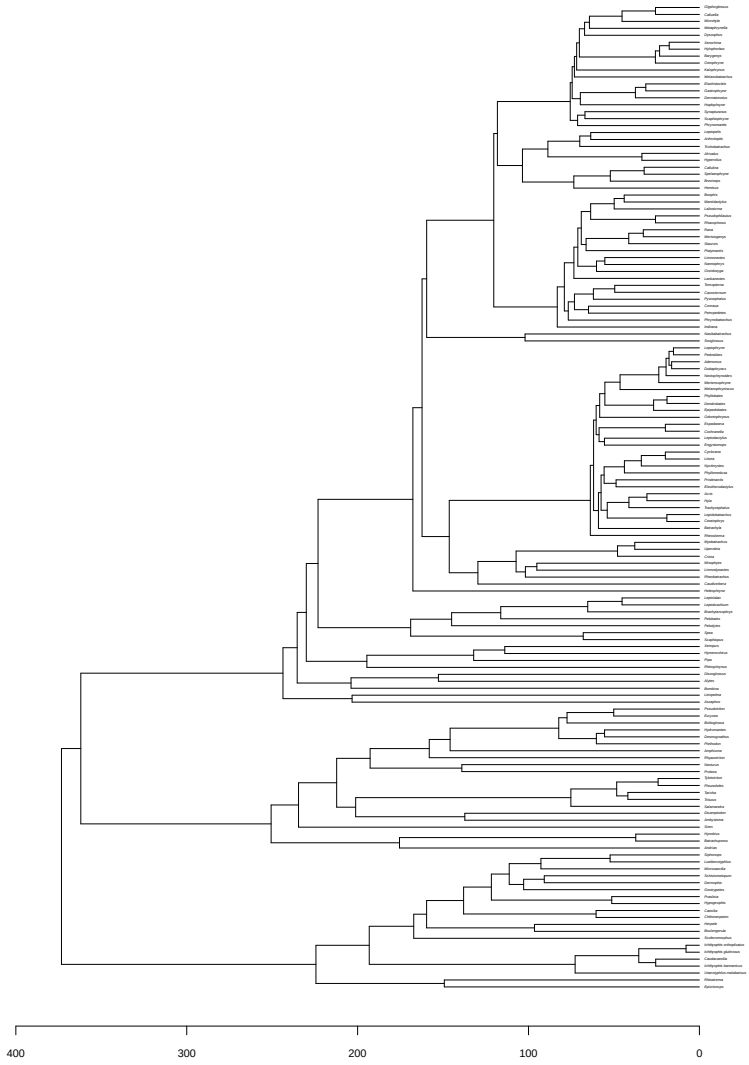


FIGURE 1. Exemplar phylogeny of Roelants et al. (2007) with scale in millions of years

Having converted the *reference* tree into an exemplar phylogeny, we can set that tree aside for the moment. Let's collect our phylogram that we should like to scale to absolute time. We'll use the tree by Pyron and Wiens (2011), involving over 2800 amphibian species. We'll need to resolve the taxonomy of tips in the tree in order to congruify.

```
> pw=ladderize(amph$pyronwiens, right=FALSE) ## Pyron and Wiens (2011) phylogram
> ## resolve taxonomy of tips (restricted to within amphibians)
> tmp=gbresolve(pw, within="Amphibia", rank=c("genus", "class"))
> tax=tmp$tax
```

Let's take a quick look at the first few lines of the taxonomic table that results:

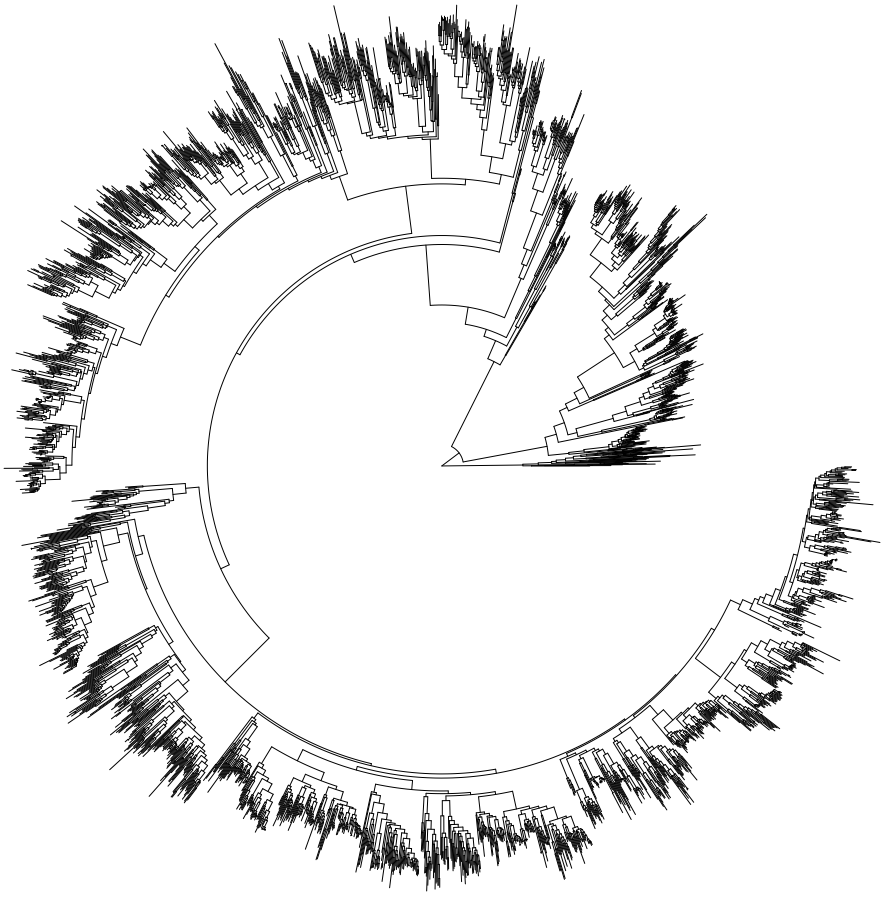
```
> head(tax[,c("family", "order")])
```

	family	order
Ichthyophis_tricolor	"Ichthyophiidae"	"Gymnophiona"
Ichthyophis_bannanicus	"Ichthyophiidae"	"Gymnophiona"
Caudacaecilia_asplenia	"Ichthyophiidae"	"Gymnophiona"
Ichthyophis_glutinosus	"Ichthyophiidae"	"Gymnophiona"
Ichthyophis_orthoplicatus	"Ichthyophiidae"	"Gymnophiona"
Ichthyophis_bombayensis	"Ichthyophiidae"	"Gymnophiona"

The taxonomic table (*tax*) will be critical in the congruification step below; this table links sampled tips in the *target* tree to the sampled tips of our *reference* tree.

Note that **gbresolve** has also resolved node labels for the higher taxa found within the created *tax* object. These labels could be plotted in addition to the tree structure. Let's plot the phylogram of Pyron and Wiens (2011), using just the tree structure:

```
> plot.phylo(pw, type="fan", show.tip=FALSE, edge.width=0.2, no.margin=TRUE)
```



3. Tree scaling

We're now prepared to congruify the amphibian tree, given that we have the *reference*, *target*, and taxonomic hierarchy for the *target*. These objects are in memory and references by the following names in bold: the exemplar-labelled Roelants et al. (2007) timetree (*reference*: `ex`), the Pyron and Wiens (2011) tree (*target*: `pw`), and the taxonomic table resolved for the same phylogeny (`tax`):

```
> res=congruify.phylo(reference=ex, target=pw, taxonomy=tax, scale=NA)
```

If no `scale` method is supplied to **congruify.phylo**, the function simply returns a table of matched nodes between the *target* and *reference*. Currently, the only canned scaling method within **congruify.phylo** is `PATHd8`, which requires that this rate-smoothing utility is accessible from the PATH available in the R console. Try `Sys.getenv("PATH")` to see if `PATHd8` is installed in a location in which R expects executables to be found. If R's PATH appears empty or is incompatible with where `PATHd8` is installed, you may need to modify the R PATH with `Sys.setenv(PATH="/path/to/executables")` where the folder containing `PATHd8` replaces `/path/to/executables`.

Using the result from above, we'll write out an infile for use in treePL via **write.treePL**. Doing so will generate two files (`amphibia.infile` and `amphibia.intree`) in our working directory. treePL could then be used to scale the Pyron and Wiens (2011) *target*.

```
> cal=res$calibrations
> ## options ('opts') can be user-specific
> write.treePL(pw, cal, base="amphibia", nsites=12712, opts=list(smooth=0.1, nthreads=2,
+         opt=1, optad=1, thorough=TRUE))
[1] "amphibia.infile"
attr(,"method")
[1] "treePL"
```

The above `opts` used in **write.treePL** select the methods used by treePL and set the smoothing parameter to a value of 0.1.

4. Summarizing

Let's look at the first lines of the table of (106) resolved calibrations:

```
> head(cal[,-which(names(cal)=="MRCA")]) ## excluding the hash keys in the first column
    MaxAge  MinAge                      taxonA                          taxonB
1  373.306 373.306     Epicrionops_marmoratus    Cryptobranchus_alleganiensis
2  224.390 224.390    Ichthyophis_bombayensis          Epicrionops_marmoratus
3  193.201 193.201    Ichthyophis_bombayensis Crotaphatrema_tchabalmbaboensis
10 167.162 167.162        Herpele_squalostoma Crotaphatrema_tchabalmbaboensis
11 159.630 159.630 Chthonerpeton_indistinctum             Herpele_squalostoma
12 137.976 137.976     Gegeneophis_seshachari      Chthonerpeton_indistinctum
```

Let's print the example of the congruified tree with labeled nodes. For convenience, the result obtained from treePL is stored with data that comes with geiger, so we can load that time-scaled tree. The user is nevertheless encouraged to attempt using treePL on his or her own with the files `amphibia.infile` and `amphibia.intree`. The function **nodelabel.phylo** provides a solution for resolving node labels based on a taxonomic table. If the tree is inconsistent with the delimitation of a taxon within the given table, the node most consistent with the delimitation can be labelled with quotes if `strict=FALSE`.

```
> phy=amph$congruified
> ## exclude genus, tribe, and subfamily from the labels (the first three columns of 'tax')
> phy=nodelabel.phylo(phy, tax[,-match(c("genus", "tribe", "subfamily"), colnames(tax))], strict=TRUE)
> plot.phylo(phy, show.tip=FALSE, edge.width=0.1, no.margin=TRUE, x.lim=c(-10,450))
> axisPhylo(cex.axis=0.75)
> nodelabels(phy$node.label, frame="none", col="lightskyblue", cex=0.85)
```

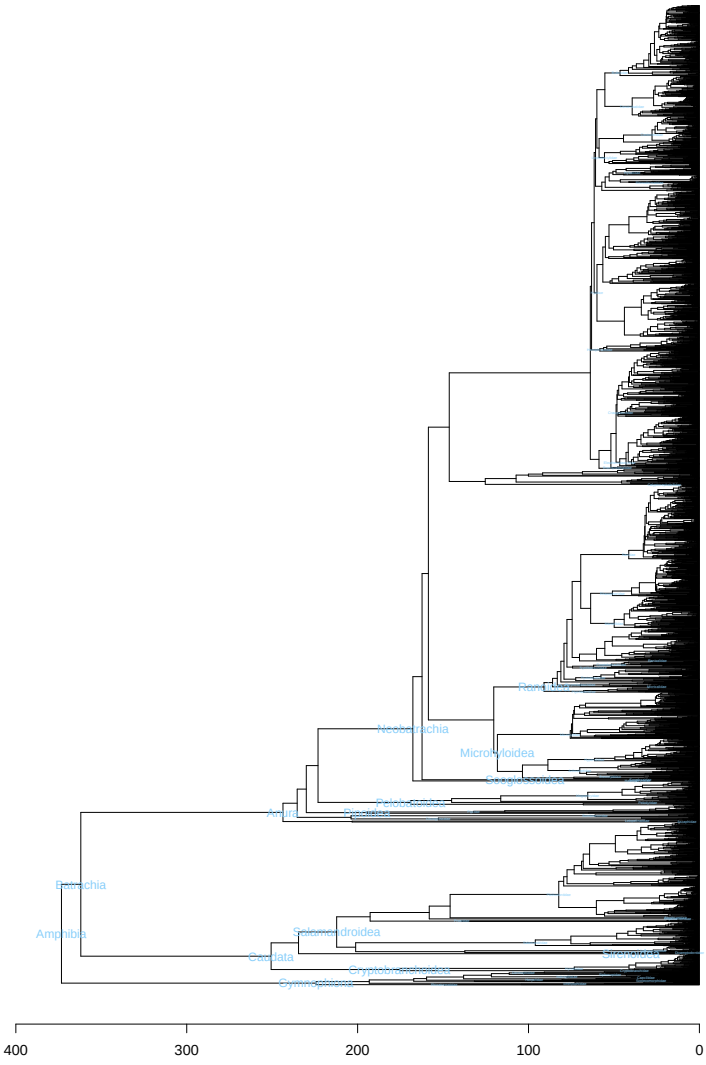


FIGURE 2. Congruified phylogeny of Pyron and Wiens (2011) given dates from Roelants et al. (2007) with scale in millions of years

We might create a custom function that can display the distribution of constraints around the tree. Among other reasons, this will give us a sense of whether we're dealing with rogue taxa that prevent full exploitation of the dates in Roelants et al. (2007). To write the function, one could open a 'New Document' from within the R console. This function will take two arguments (the tree and the table of calibrations) and will return a nodewise vector that will indicate whether a given node was matched.

```
> mrcaID=function(phy, cal){
+         cal=as.matrix(cal)
+         res=sapply(1:nrow(cal), function(idx){ ## loop over rows
+                         tips=cal[idx, c("taxonA", "taxonB")] ## fetch spanning taxa
+                         return(geiger:::.mrca(tips, phy)) ## MRCA of spanning taxa (node ID)
+         })
+         N=Ntip(phy)
+         n=Nnode(phy)
+         nn=integer(N+n) ## create empty vector of same length as branches in tree
+         nn[res]=1 ## identify nodes that appear within calibrations
+         nn=nn[-c(1:N)] ## exclude tip branches
+         return(nn) ## return vector (ordered from first to last internal node in the tree)
+ }
> vec=mrcaID(phy, cal) ## get vector for calibrated nodes
> sum(vec)==nrow(cal) ## check on whether the function is working appropriately
[1] TRUE
> plot.phylo(phy, type="fan", show.tip=FALSE, edge.width=0.1)
> ## plot box at node only if calibrated
> nodelabels(text=NULL, cex=ifelse(vec==0, NA, 2), frame="n", bg="white", pch=21)
```

We see that the calibrated nodes are fairly well distributed across the phylogeny, and most of the calibrations are relatively deep in time (as we should expect from a sparsely sampled *reference* tree). The presence of rogue taxa (i.e., taxa whose placement is widely inconsistent between the *reference* and *target*) would be indicated where calibrations are lacking (but expected) within a large subtree in the *target*.

To reiterate what we have accomplished in this vignette, we've developed a single scaling of a large species-level amphibian tree (Pyron and Wiens 2011) using dates derived from an exemplar-sampled timetree of Roelants et al. (2007). We used taxonomic data from NCBI and the congruification method to resolve concordant nodes between these two trees. We lastly used TREEPL to scale the congruified tree to time. If we were to conduct further analyses, we would ideally be able to express our uncertainty in the timing of splits and topology of both trees. A reasonable means of doing so would be to iterate congruification across many estimates of both trees (e.g., sampling from the posterior distributions of trees for both datasets). While we have used a single *reference* tree in the demonstrated example, one could easily extend the method to involve many *reference* trees from which each temporal constraint (i.e., a pair of *MinAge* and *MaxAge* for a given node) represents a true range of plausible dates.

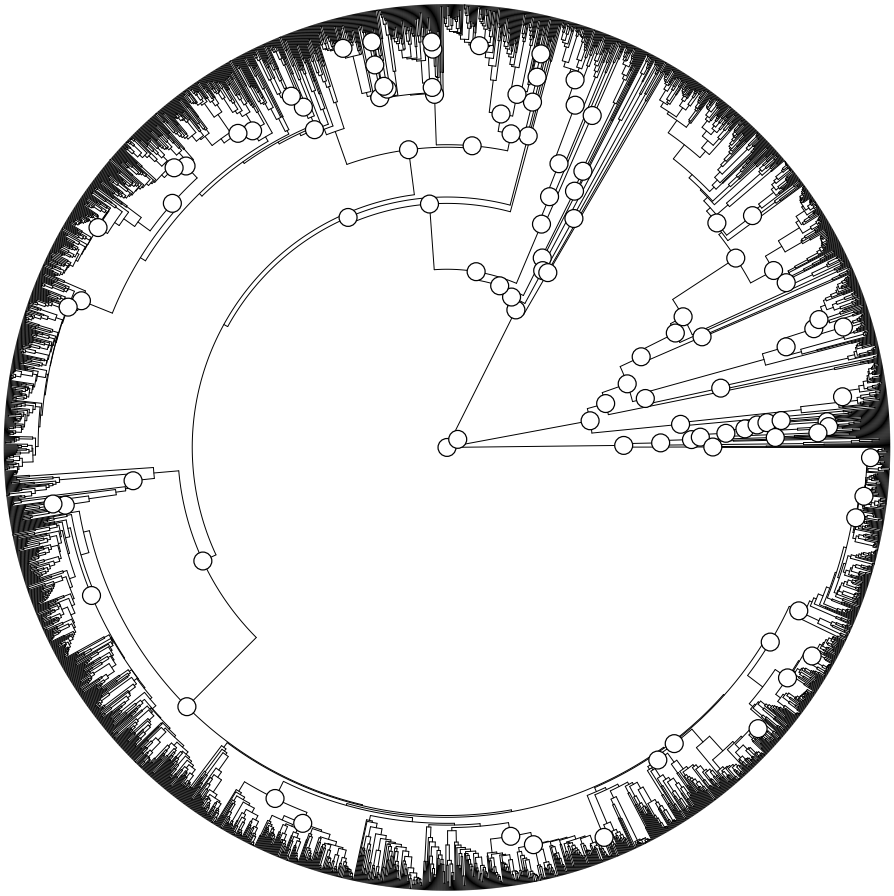


FIGURE 3. Distribution of secondary calibrations derived from Roelants et al. (2007)