# phruta: scrapping genbank and assembling phylogenetic trees [*]

**Cristian Román-Palacios** *School of Information, University of Arizona, Tucson, Arizona 85721, USA.*
*ORCiD : 0000-0003-1696-4886*

Current methodological practices for assembling phylogenetic trees often recur to sequence data stored in GenBank. However, understanding molecular and taxonomic availability in GenBank is generally not very straightforward. For instance, the genetic makeup of datasets available in GenBank can strongly differ between genera even within the same family. Similarly, the taxonomic information associated with sequence data in GenBank can be outdated, relative to other databases that mainly focus on the taxonomic side. phruta, a newly developed R package, is designed to improve the user experience and access to information to genetic data stored in GenBank. By using phruta, users are able to (1) quantitatively explore the molecular makeup of particular clades with information in GenBank, (2) assemble curated multi-gene molecular datasets with retrieved and local sequences, and (3) run basic phylogenetic talks, all within R. The structure of the functions implemented in phruta, designed as a workflow, aim to allow users to assemble simple workflows for particular talks, which are in turn expected to increase reproducibility when assembling phylogenies. This paper provides a brief overview on the performance and workflow associated with phruta.

*Keywords*: R package, Phylogenetics, Reproducibility, Workflow

## *Background*

## *The phruta R package*

The `phruta` package is designed to simplify the basic phylogenetic pipeline in `R`. `phruta` is designed to allow scientists from different backgrounds to assemble molecular databases or phylogenies for particular taxa with as minimal complexity and maximal reproducibility as possible. All the code in `phruta` is run within the same software (`R`) and data from intermediate steps are either stored to the environment or exported locally to independent folders. In general, `phruta` is able to (1) find potentially (phylogenetically) relevant gene regions for a given set of taxa based on GenBank, (2) retrieve gene sequences and curate taxonomic information from the same database, (3) combine downloaded and local gene sequences, and (4) perform sequence alignment, phylogenetic inference, and basic tree dating tasks.

[*]Replication files are available on the author's Github account (http://github.com/cromanpa). **Current version**: September 01, 2022; **Corresponding author**: cromanpa94@arizona.edu.

*Alternatives to phruta*

Similar functionalities for assembling curated molecular datasets for phylogenetic analyses can be found in `phylotaR` and SuperCRUNCH. However, note that `phylotaR` is limited to downloading and curating sequences (e.g. doesn't align sequences). Similarly, `SuperCRUNCH` only curates local sequences. `phruta` is closer to the `SUPERSMART` and its "new" associated `R` workflow `SUPERSMARTR`. However, most of the applications in the different packages that are part of `SUPERSMARTR` are simplified in `phruta`. Standalone applications that might resemble `phruta` could include MEGA and geneious. However, analyses in these two alternatives are either poorly reproducible or not available to everyone (e.g. geneious has a paid version). `phruta` aims to address these issues.

*phruta in a nutshell*

The current release of `phruta` includes a set of eight major functions. All eight functions form a pipeline within `phruta` to output a time-calibrated phylogeny. However, users interested in using their own files at any stage can run each function independently. Note that all the functions for which their primary output are sequences (aligned or unaligned) are listed under `sq.*`. All the files that output phylogenies (time-calibrated or not) are listed under `tree.*`.

- First, the distribution of gene sampled for a given organism or set of taxa can be explored using the `acc.gene.sampling` function. This function will return a table that summarizes either the distribution of genes sampled for the search term in general or specifically across species.

- Second, given a list of target organisms, users can retrieve a list of accession numbers that are relevant to their search using `acc.table.retrieve()`. Instead of directly downloading sequences from genbank (see `sq.retrieve.direct()` below), retrieving accession numbers allow users to have more control over the sequences that are being used in the analyses. Note that users can also curate the content of the dataset obtained using `sq.retrieve.direct()`.

- Third, users should download gene sequences. Sequences can be download using the `sq.retrieve.indirec` from the accession numbers retrieved before using the `acc.table.retrieve()` function. This is the preferred option within `phruta`. Additionally, users can directly download gene sequences using the `sq.retrieve.direct()` function. Both `sq.retrieve.indirect()` and `sq.retrieve.direct()` functions save gene sequences in `fasta` files that will be located in a new directory named `0.Sequences`.

- Fourth, `sq.add()` allows users to include local sequences to those retrieved from genbank in the previous step. This function saves all the resulting `fasta` files in two directories, combined sequences in `0.Sequences` and local sequences in `0.AdditionalSequences` (originally downloaded sequences are moved to `0.0.OriginalDownloaded` at this step). Note that `sq.add()` is optional.

- Fifth, the `sq.curate()` function filters out unreliable sequences based on information listed in genbank (e.g. PREDICTED) and on taxonomic information provided by the user. Specifically, this function retrieves taxonomic information from the Global Biodiversity Information Facility (GBIF) database's taxonomic backbone (see alternatives in the advanced vignette to `phruta`). If a given species belongs to a non-target group, this species is dropped from the analyses. This function automatically corrects taxonomy and renames sequences.

- Sixth, `sq.aln()` performs multiple sequence alignment on `fasta` files. Currently, `phruta` uses the `DECIPHER` R package, here. This package allows for adjusting sequence orientation and masking (removing ambiguous sites).

- Seventh, the `tree.raxml()` function allows users to perform tree inference under `RAxML` for sequences in a given folder. This is a wrapper to `ips::raxml()` and each of the arguments can be customized. The current release of `phruta` can manage both partitioned and unpartitioned analyses. Starting and constrained trees are allowed.

- Eight, `tree.dating()` enables users to perform time-calibrations of a given phylogeny using `geiger::congruify.phylo()`. `phruta` includes a basic set of comprehensively sampled, time-calibrated phylogenies that are used to extract secondary calibrations for the target phylogeny. Note that sampling in those phylogenies can be examined using `data(SW.phruta)`. Please make sure you have at least **two** groups in common with each of the phylogenies. Similarly, users can choose to run either `PATHd-8` or `treePL`.

*Assembling a molecular dataset for target taxa in phruta*

Let's learn how `phruta` works by assembling a molecular dataset at the species level for a few bird clades. For this tutorial, we will focus on assembling a phylogeny for the new world Quails (family Odontophoridae), including nearly 34 classified in 10 genera. We will use the Phasianidae as an outgroup, with special emphasis in the genus Polyplectron (8 species). Luckily, we will be able to examine how similar is the resulting tree for the Odontophoridae to existing phylogenies (Crowe et al. 2006a, b; Cohen et al. 2012; Hosner et al. 2015).

So far, we have decided the taxonomic makeup of our analyses. For this, we could simply check the taxonomic sampling of previous studies (Crowe et al. 2006a, b; Cohen et al. 2012; Hosner et al. 2015). For instance, [review sampling in each of those]. From here we could choose to either use the same sampling previous studies have used or try to figure out what genes are well sampled in GenBank. Although `phruta` is able to accommodate for both options, we will simply find well sampled genes in GenBank. Let's now load `phruta`!

```
library(phruta)
```

Let's now look for the gene regions that are extensively sampled in our target taxa. For this, we will use the `gene.sampling.retrieve()` function in `phruta`. The resulting `data.frame`, named `gs.seqs` in this example, will contain the list of full names for genes sampled in GenBank for the target taxa.

```
gs.seqs <- gene.sampling.retrieve(organism = c("Odontophoridae", "Polyplectron"),
                                  speciesSampling = TRUE)
```

For the search terms, `phruta` was able to retrieve the names for r nrow(gs.seqs) gene regions. The frequency estimates per gene are based on inter-specific sampling @ref(tab:topGenes).

The `gene.sampling.retrieve()` provides an estimate of the number of species in genbank (matching the taxonomic criteria of the search term) that have sequences for a given gene region. However, this estimate is only as good as the annotations for genes deposited in genbank.

From here, we will generate a preliminary summary of the accession numbers sampled for the combination of target taxa and gene regions. In fact, not all these accession numbers are expected to be in the final (curated) molecular dataset. Using the `acc.table.retrieve()` function, we will assemble a species-level summary of accession numbers (hence the `speciesLevel = TRUE` argument). For simplicity, this tutorial will focus on sampling gene regions that are sampled in >30% of the species (`targetGenes` data.frame).

```
targetGenes <- gs.seqs[gs.seqs$PercentOfSampledSpecies > 20,]
acc.table <- acc.table.retrieve(
            clades  = "Odontophoridae",
            species = "Polyplectron",
            genes = targetGenes$Gene,
            speciesLevel = TRUE
          )
```

The `acc.table` object is a `data.frame` that is later on going to be used for downloading locally the relevant gene sequences. In this case, the dataset includes the following information (@ref(tab:AccN)):

Now, since we're going to retrieve sequences from genbank using an existing preliminary accession numbers table, we will use the `sq.retrieve.indirect()` function in `phruta`. Please note that there are two versions of `sq.retrieve.*` in `phruta` . The one that we're using in this tutorial, `sq.retrieve.indirect()`, retrieves sequences "indirectly" because it necessarily follows the initial step of generating a table summarizing target accession numbers (see the `acc.table.retrieve()` function above). I present the information in this vignette using `sq.retrieve.indirect()` instead of `sq.retrieve.direct()` because the first function is way more flexible. Specifically, it allows for correcting issues *prior* to downloading/retrieving any sequence. For instance, you can add new sequences, species, populations to the resulting data.frame from `acc.table.retrieve()`. Additionally, you could even manually assemble your own dataset of accession numbers to be retrieved using `sq.retrieve.indirect()`. Instead, `sq.retrieve.direct()` does its best to directly (i.e. without input from the user) retrieve sequences for a target set of taxa and set of gene regions. In short, you should be able to catch errors using `sq.retrieve.indirect()` but mistakes will be harder to spot and fix if you're using `sq.retrieve.direct()`.

Now, we still need to retrieve all the sequences from the accessions table generated using acc.table. Note that since we have specified download.sqs = FALSE, the sequences are returned in a list that is stored in your global environment. If we decide to download the sequences to our working directory using download.sqs = TRUE, phruta will write all the resulting fasta files into a newly created folder 0.Sequences located in our working directory. The latter option is covered in the "To export or not export 'phruta' outputs" vignette.

```
sqs.downloaded <- sq.retrieve.indirect(acc.table = acc.table, download.sqs = FALSE)
```

We're now going to make sure that we include only sequences that are reliable and from species that we are actually interested in analyzing. We're going to use the sq.curate() function for this. We will provide a list of taxonomic criteria to filter out incorrect sequences (filterTaxonomicCriteria argument). For instance, we could simply provide a vector of the genera that we're interested in analyzing. This vector must have a length of 1, with all the target genera being separated with | (e.g. "Callipepla|Colinus|Dendrortyx"). For now, we will assume that all of the species we downloaded are relevant to the analyses (i.e. filterTaxonomicCriteria = [AZ]). Note that, since we're not downloading anything to our working directory, we need to pass our downloaded sequences (sqs.downloaded object generated above using the sq.retrieve.indirect() function) to the sqs.object argument in sq.curate().

```
sqs.curated <- sq.curate(filterTaxonomicCriteria = '[AZ]',
                         kingdom = 'animals',
                         sqs.object = sqs.downloaded,
                         removeOutliers = FALSE)
```

Running the sq.curate() function will create an object of class list that includes (1) the curated sequences with original names, (2) the curated sequences with species-level names (renamed_* prefix), (3) the accession numbers table (AccessionTable), and (4) a summary of taxonomic information for all the species sampled in the files (Taxonomy.csv). First, the newly updated accession numbers table (@ref(tab:tw))

We can also check out the taxonomic information associated with each species in our dataset @ref(tab:tw2)

Finally, we can generate a table summarizing the overlap between species-level sampling in the molecular dataset @ref(tab:tw3):

Now, we'll align the sequences that we just curated. For this, we just use `sq.aln()` with default parameters. We're again passing the output from `sq.curate()`, `sqs.curated`, using the `sqs.object` argument in `sq.aln()`.

```
sqs.aln <- sq.aln(sqs.object = sqs.curated)
```

The resulting multiple sequence alignments will be saved to `sqs.aln()` object, a list or alignments. For each of the gene regions, we will have access to the original alignment (`Aln.Original`), the masked one (`Aln.Masked`), and information on the masking process.

Note that we could use these resulting alignments to infer phylogenies. We cover these steps within `phruta` in another vignette: "Phylogenetics with the `phruta` R package". For now, let's wrap up and plot one of our cool alignments. Let's first check the raw alignments.

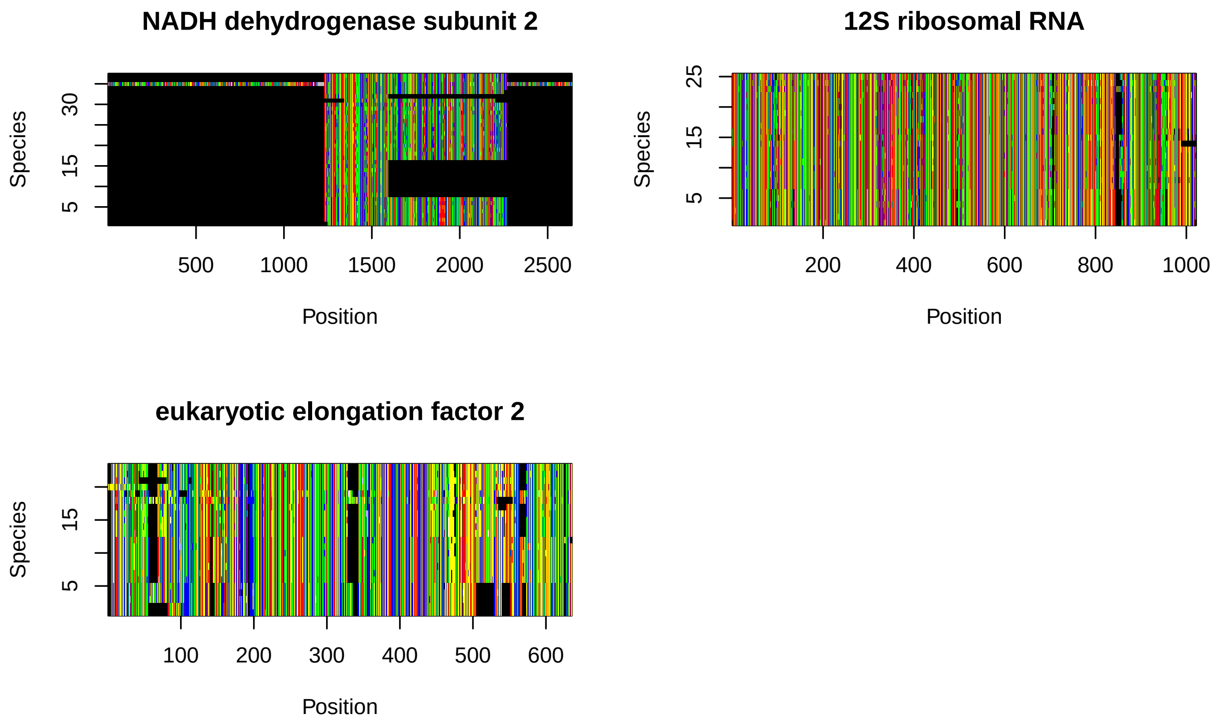

Figure 1: A figure showing raw alignments

Now, the masked alignments…

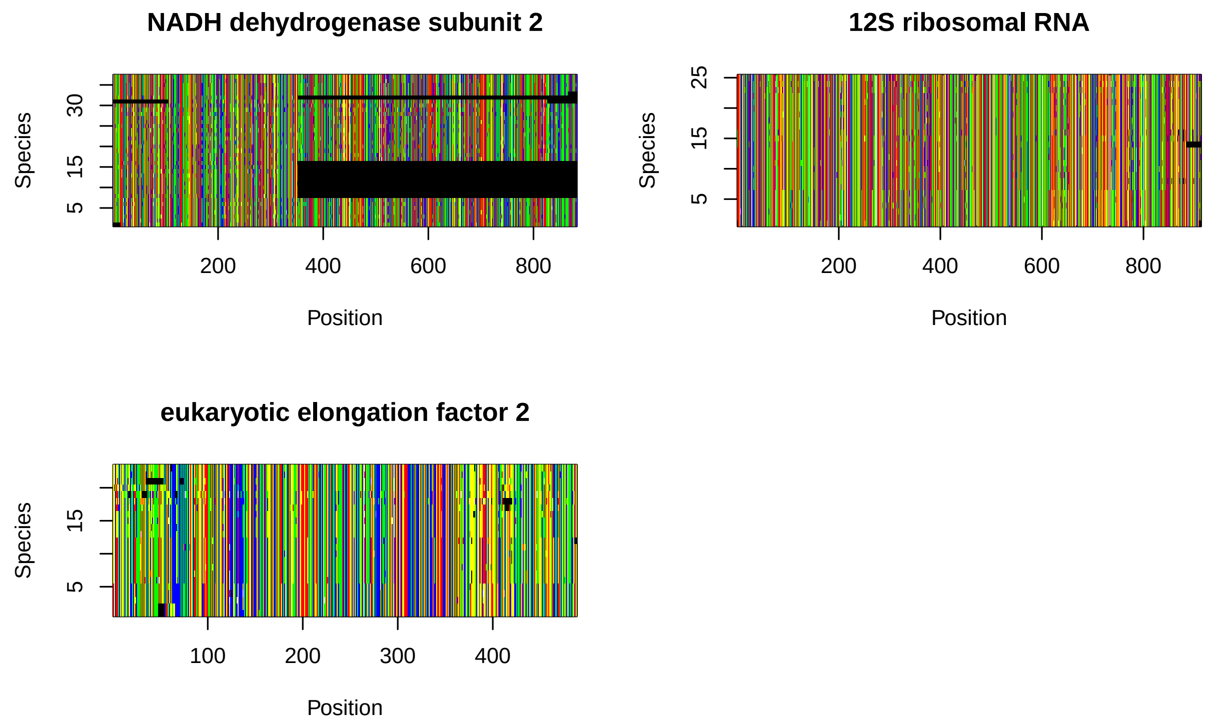

Figure 2: A figure showing curated alignments

Basic phylogenetics with phruta

### *Phylogenetic inference with* `phruta` *and* `RAxML`

Phylogenetic inference is conducted using the `tree.raxml()` function. We need to indicate where the aligned sequences are located (`folder` argument), the patterns of the files in the same folder (`FilePatterns` argument; “`Masked_`” in our case). We’ll run a total of 100 boostrap replicates and set the outgroup to “Manis_pentadactyla”.

```
tree.raxml(folder='2.Alignments',
           FilePatterns= 'Masked_',
           raxml_exec='raxmlHPC',
           Bootstrap=100,
           outgroup ="Manis_pentadactyla")
```

The trees are saved in `3.Phylogeny`. Likely, the bipartitions tree, “RAxML_bipartitions.phruta”, is the most relevant. `3.Phylogeny` also includes additional `RAxML`-related input and output files.

Finally, let's perform tree dating in our phylogeny using secondary calibrations extracted from Scholl and Wiens (2016). This study curated potentially the most comprenhensive and reliable set of trees to summarize the temporal dimension in evolution across the tree of life. In `phruta`, the trees from Scholl and Wiens (2016) were renamed to match taxonomic groups.

### *Tree dating in* `phruta`

Tree dating is performed using the `tree.dating()` function in `phruta`. We have to provide the name of the folder containing the `1.Taxonomy.csv` file created in `sq.curate()`. We also have to indicate the name of the folder containing the `RAxML_bipartitions.phruta` file. We will scale our phylogeny using `treePL`.

```
tree.dating(taxonomyFolder="1.CuratedSequences",
            phylogenyFolder="3.Phylogeny",
            scale='treePL')
```

Running this line will result in a new folder `4.Timetree`, including the different time-calibrated phylogenies obained (if any) and associated secondary calibrations used in the analyses. We found only a few overlapping calibration points (family-level constraints):

Here's the resulting time-calibrated phylogeny. The whole process took ~20 minutes to complete on my computer (16 gb RAM, i5).

Advanced methods with phruta As explained in the brief intro to `phruta`, the `sq.curate()` function is primarily designed to curate taxonomic datasets using `gbif`. Alto `gbif` is extremely fast and efficient, it is largely designed to deal with animals and plants. If you're interested in using the `gbif` backbone for curating sequence regardless of the kingdom use the following approach:

```
taxonomy.retrieve(species_names=c("Felis_catus", "PREDICTED:_Vulpes",
                  "Phoca_largha", "PREDICTED:_Phoca" ,
                  "PREDICTED:_Manis" , "Felis_silvestris" , "Felis_nigripes"),
                  database='gbif', kingdom=NULL)
```

Note that the `kingdom` argument is set to `NULL`. However, as indicated in the first vignette, `gbif` is efficient for retrieving accurate taxonomy when we provide details on the `kingdom`. Given that all the species we're interested in are animals, we could just use:

```
taxonomy.retrieve(species_names=c("Felis_catus", "PREDICTED:_Vulpes",
                  "Phoca_largha", "PREDICTED:_Phoca" ,
                  "PREDICTED:_Manis" , "Felis_silvestris" , "Felis_nigripes"),
                  database='gbif', kingdom='animals')
```

We could also do the same for plants by using `plants` instead of `animals` in the `kingdom` argument. Now, what if we were interested in following other databases to retrieve taxonomic information for the species in our database? The latest version of `phruta` allow users to select the desired database. The databases follow the `taxize::classification()` function. Options are: `ncbi`, `itis`, `eol`, `tropicos`, `nbn`, `worms`, `natserv`, `bold`, `wiki`, and `pow`. Please select only one. Note that the `gbif` option in `taxize::classification()` is replaced by the internal `gbif` in `phruta`.

Now, let's assume that we were interested in curating our database using `itis`:

```
taxonomy.retrieve(species_names=c("Felis_catus", "PREDICTED:_Vulpes",
                  "Phoca_largha", "PREDICTED:_Phoca" ,
                  "PREDICTED:_Manis" , "Felis_silvestris" , "Felis_nigripes"),
                  database='itis')
```

Using alternative databases is sometimes desirable. Please make sure you review which the best database is for your target group is before selecting one.

## *Creating taxonomic constraints in `phruta`*

For different reasons, phylogenetic analyses sometimes require of tree constraints. `phruta` can automatically generate trees in accordance to taxonomy and a backbone topology. We divide constraint trees into two: (1) ingroup+outgroup and (2) particular clades.

*ingroup + outgroup*

In this constraint type, `phruta` will create monophyletic groups for each of the taxonomic groups in the database (for selected target columns). Finally, it will generate tree with the same topology provided in the `Topology` argument. The user will provide the species names of the outgroup taxa as a vector of string that should fully match the names in the taxonomy file.

```
tree.constraint(
               taxonomy_folder = "1.CuratedSequences",
               targetColumns = c("kingdom", "phylum", "class", "order",
                                 "family", "genus", "species_names"),
               Topology = "((ingroup), outgroup);",
               outgroup = "Manis_pentadactyla"
 )
```

*Particular clades*

In this constraint type, `phruta` will create a constraint tree for particular clades. For instance, let's assume that we only need to create a tree constraining the monophyly within two genera and their sister relationships:

```
tree.constraint( taxonomy_folder = "1.CuratedSequences",
                targetColumns = c("kingdom", "phylum", "class",
                                  "order", "family", "genus", "species_names"),
                Topology = "((Felis), (Phoca));"
 )
```

Note that the key aspect here is the `Topology` argument. It is a newick tree.

*Running PartitionFinder in `phruta`*

With the current version of `phruta`, users are able to run `PartitionFinder` v1 within `R`. For this, users should provide the name of the folder where the alignments are stored, a particular pattern in the file names (masked in our case), and which models will be run in PartitionFinder. This function will download PartitionFinder, generate the input files, and run it all within R. The output files will be in a new folder within the working directory.

```
sq.partitionfinderv1(folderAlignments = "2.Alignments",
                    FilePatterns = "Masked",
                    models = "all"
 )
```

Unfortunately, the output files are not integrated with the current phruta pipeline. This will be part of a new release. However, users can still perform gene-based partitioned analyses within RAxML or can use PartitionFinder's output files to inform their own analyses outside phruta.

*Partitioned analyses in RAxML*

Users can now run partitioned analyses in RAxML within phruta. This approach is implemented by setting the partitioned argument in tree.raxml to TRUE. For now, partitions are based on the genes are being analyzed. The same model is used to analyze each partition. More details on partitioned analyses can be customized by passing arguments in ips::raxml.

```
tree.raxml(folder = "2.Alignments", FilePatterns = "Masked",
           raxml_exec = "raxmlHPC", Bootstrap = 100,
           outgroup = "Manis_pentadactyla",
           partitioned=T
)
```

*Identifying rogue taxa*

phruta can help users run RogueNaRok implemented in the Rogue R package. Users can then examine whether rogue taxa should be excluded from the analyses. tree.roguetaxa() uses the bootstrap trees generated using the tree.raxml() function along with the associated best tree to identify rogue taxa.

```
tree.roguetaxa(folder = "3.Phylogeny")
```

Reproducibility with phruta Users can choose to share the script they used to run the analyses (e.g. assemble their molecular dataset) and the associated workspace.

Performance

http://jboyd.net/Taxo/Odontophoridae.pdf

Hosner, P.A., E.L. Braun, and R.T. Kimball (2015a), Land connectivity changes and global cooling shaped the colonization history and diversification of New World quail (Aves: Galliformes: Odontophoridae), J. Biogeogr. 42, 1883-1895.

Crowe, T.M., R.C.K. Bowie, P. Bloomer, T.G. Mandiwana, T.A.J. Hedderson, E. Randi, S. Pereira, and J. Wakeling (2006a), Phylogenetics, biogeography and classification of, and character evolution in, gamebirds (Aves: Galliformes): Effects of character exclusion, data partitioning and missing data, Cladistics 22, 495-532.

Crowe, T.M., P. Bloomer, E. Randi, V. Lucchini, R. Kimball, E. Braun, and J.G. Groth (2006b), Supra-generic cladistics of landfowl (Order Galliformes), Acta Zool. Sinica 52, S358-S361.

Cohen, C., J.L. Wakeling, T.G. Mandiwana-Neudani, E. Sande, C. Dranzoa, T.M. Crowe, and R.C.K. Bowie (2012), Phylogenetic affinities of evolutionarily enigmatic African galliforms: the Stone Partridge Ptilopachus petrosus and Nahan's Francolin Francolinus nahani, and support for their sister relationship with New World quails, Ibis 154, 768-780.

Table 1: Top seven genes sampled at the species level for GenBank sequences assigned to Odontophoridae and Polyplectron.

| Gene | Sampled in N species | PercentOfSampledSpecies |
|---|---:|---:|
| NADH dehydrogenase subunit 2 | 90 | 98.90110 |
| 12S ribosomal RNA | 27 | 29.67033 |
| eukaryotic elongation factor 2 | 25 | 27.47253 |
| NADH dehydrogenase subunit 5 | 18 | 19.78022 |
| cytochrome b | 16 | 17.58242 |
| cytochrome oxidase subunit 1 | 10 | 10.98901 |

Table 2: Accession numbers for species in Odontophoridae, our ingroup, and Polyplectron, outgroup genys.

| Species | Ti |
|---|---|
| Callipepla californica | Callipepla californica voucher MMNH<USA-MN>:180 NADH dehydrogenase s |
| Callipepla gambelii | Callipepla gambelii voucher MZFC:TIB14 NADH dehydrogenase subunit 2 (ND |
| Colinus virginianus | Colinus virginianus tRNA-Leu gene, partial sequence; NADH dehydrogenase su |
| Colinus cristatus | Colinus cristatus NADH dehydrogenase subunit 2 (ND2) gene, complete cds; m |
| Colinus nigrogularis | Colinus nigrogularis NADH dehydrogenase subunit 2 (ND2) gene, ND2-NADH |
| Dendrortyx barbatus | Dendrortyx barbatus NADH dehydrogenase subunit 2 (ND2) gene, ND2-NADH |
| Philortyx fasciatus | Philortyx fasciatus NADH dehydrogenase subunit 2 (ND2) gene, partial cds; mi |
| Rhynchortyx cinctus | Rhynchortyx cinctus NADH dehydrogenase subunit 2 (ND2) gene, complete cds |
| Cyrtonyx montezumae | Cyrtonyx montezumae NADH dehydrogenase subunit 2 (ND2) gene, complete |
| Oreortyx pictus | Oreortyx pictus NADH dehydrogenase subunit 2 (ND2) gene, complete cds; mit |
| Odontophorus leucolaemus | Odontophorus leucolaemus NADH dehydrogenase subunit 2 (ND2) gene, comp |
| Odontophorus speciosus | Odontophorus speciosus NADH dehydrogenase subunit 2 (ND2) gene, complet |
| Odontophorus erythrops | Odontophorus erythrops NADH dehydrogenase subunit 2 (ND2) gene, complet |
| Odontophorus guttatus | Odontophorus guttatus NADH dehydrogenase subunit 2 (ND2) gene, complete |
| Odontophorus gujanensis | Odontophorus gujanensis NADH dehydrogenase subunit 2 (ND2) gene, comple |
| Odontophorus capueira | Odontophorus capueira NADH dehydrogenase subunit 2 (ND2) gene, complete |
| Odontophorus stellatus | Odontophorus stellatus NADH dehydrogenase subunit 2 (ND2) gene, complete |
| Dactylortyx thoracicus | Dactylortyx thoracicus NADH dehydrogenase subunit 2 (ND2) gene, complete c |
| Dendrortyx macroura | Dendrortyx macroura NADH dehydrogenase subunit 2 (ND2) gene, complete c |
| Callipepla squamata | Callipepla squamata NADH dehydrogenase subunit 2 (ND2) gene, complete cds |
| Callipepla douglasii | Callipepla douglasii NADH dehydrogenase subunit 2 (ND2) gene, complete cds |
| Colinus leucopogon | Colinus leucopogon sclateri voucher UCLA:31817 NADH dehydrogenase subur |
| Dendrortyx leucophrys | Dendrortyx leucophrys voucher AMNH:388928 NADH dehydrogenase subunit |
| Cyrtonyx ocellatus | Cyrtonyx ocellatus voucher AMNH:143653 NADH dehydrogenase subunit 2 (N |
| Odontophorus balliviani | Odontophorus balliviani voucher AMNH:820368 NADH dehydrogenase subuni |
| Odontophorus columbianus | Odontophorus columbianus voucher AMNH:472695 NADH dehydrogenase sub |
| Odontophorus strophium | Odontophorus strophium voucher AMNH:181791 NADH dehydrogenase subur |
| Odontophorus melanonotus | Odontophorus melanonotus voucher AMNH:123807 NADH dehydrogenase sub |
| Odontophorus hyperythrus | Odontophorus hyperythrus voucher AMNH:107639 NADH dehydrogenase sub |
| Odontophorus melanotis | Odontophorus melanotis melanotis voucher AMNH:186995 NADH dehydroger |
| Polyplectron inopinatum | Polyplectron inopinatum NADH dehydrogenase subunit 2 (ND2) gene, complet |
| Polyplectron napoleonis | Polyplectron napoleonis NADH dehydrogenase subunit 2 (ND2) gene, complete |
| Polyplectron chalcurum | Polyplectron chalcurum NADH dehydrogenase subunit 2 (ND2) gene, complete |
| Polyplectron bicalcaratum | Polyplectron bicalcaratum NADH dehydrogenase subunit 2 (ND2) gene, comple |
| Polyplectron malacense | Polyplectron malacense NADH dehydrogenase subunit 2 (ND2) gene, complete |
| Polyplectron germaini | Polyplectron germaini NADH dehydrogenase subunit 2 (ND2) gene, complete c |
| Polyplectron katsumatae | Polyplectron katsumatae NADH dehydrogenase subunit 2 (ND2) gene, partial c |
| Cyrtonyx montezumae | Cyrtonyx montezumae 12S ribosomal RNA gene, partial sequence; mitochondri |
| Dactylortyx thoracicus | Dactylortyx thoracicus 12S ribosomal RNA gene, partial sequence; mitochondria |
| Oreortyx pictus | Oreortyx pictus 12S ribosomal RNA gene, partial sequence; mitochondrial |
| Odontophorus erythrops | Odontophorus erythrops 12S ribosomal RNA gene, partial sequence; mitochond |
| Odontophorus gujanensis | Odontophorus gujanensis 12S ribosomal RNA gene, partial sequence; mitochon |
| Odontophorus stellatus | Odontophorus stellatus 12S ribosomal RNA gene, partial sequence; mitochondri |
| Odontophorus capueira | Odontophorus capueira 12S ribosomal RNA gene, partial sequence; mitochondr |
| Odontophorus speciosus | Odontophorus speciosus 12S ribosomal RNA gene, partial sequence; mitochond |
| Odontophorus leucolaemus | Odontophorus leucolaemus 12S ribosomal RNA gene, partial sequence; mitocho |
| Odontophorus balliviani | Odontophorus balliviani 12S ribosomal RNA gene, partial sequence; mitochond |
| Dendrortyx macroura | Dendrortyx macroura 12S ribosomal RNA gene, partial sequence; mitochondria |

Table 3: 4. Preliminary accession number table

| OriginalNames | AccN | Species | file |
|---|---|---|---|
| MZ476322 Callipepla californica | MZ476322 | Callipepla_californica | NADH dehydrogenase s |
| MZ476314 Callipepla gambelii | MZ476314 | Callipepla_gambelii | NADH dehydrogenase s |
| EU166949 Colinus virginianus | EU166949 | Colinus_virginianus | NADH dehydrogenase s |
| AF222544 Colinus cristatus | AF222544 | Colinus_cristatus | NADH dehydrogenase s |
| KR732857 Colinus nigrogularis | KR732857 | Colinus_nigrogularis | NADH dehydrogenase s |
| KR732856 Dendrortyx barbatus | KR732856 | Dendrortyx_barbatus | NADH dehydrogenase s |
| KR732855 Philortyx fasciatus | KR732855 | Philortyx_fasciatus | NADH dehydrogenase s |
| KR732850 Rhynchortyx cinctus | KR732850 | Rhynchortyx_cinctus | NADH dehydrogenase s |
| KR732849 Cyrtonyx montezumae | KR732849 | Cyrtonyx_montezumae | NADH dehydrogenase s |
| KR732848 Oreortyx pictus | KR732848 | Oreortyx_pictus | NADH dehydrogenase s |
| KR732847 Odontophorus leucolaemus | KR732847 | Odontophorus_leucolaemus | NADH dehydrogenase s |
| KR732846 Odontophorus speciosus | KR732846 | Odontophorus_speciosus | NADH dehydrogenase s |
| KR732845 Odontophorus erythrops | KR732845 | Odontophorus_erythrops | NADH dehydrogenase s |
| KR732844 Odontophorus guttatus | KR732844 | Odontophorus_guttatus | NADH dehydrogenase s |
| KR732843 Odontophorus gujanensis | KR732843 | Odontophorus_gujanensis | NADH dehydrogenase s |
| KR732842 Odontophorus capueira | KR732842 | Odontophorus_capueira | NADH dehydrogenase s |
| KR732841 Odontophorus stellatus | KR732841 | Odontophorus_stellatus | NADH dehydrogenase s |
| KR732840 Dactylortyx thoracicus | KR732840 | Dactylortyx_thoracicus | NADH dehydrogenase s |
| KR732839 Dendrortyx macroura | KR732839 | Dendrortyx_macroura | NADH dehydrogenase s |
| KR732838 Callipepla squamata | KR732838 | Callipepla_squamata | NADH dehydrogenase s |
| KR732837 Callipepla douglasii | KR732837 | Callipepla_douglasii | NADH dehydrogenase s |
| KC556543 Colinus leucopogon | KC556543 | Colinus_leucopogon | NADH dehydrogenase s |
| KC556066 Dendrortyx leucophrys | KC556066 | Dendrortyx_leucophrys | NADH dehydrogenase s |
| KC556060 Cyrtonyx ocellatus | KC556060 | Cyrtonyx_ocellatus | NADH dehydrogenase s |
| KC556524 Odontophorus balliviani | KC556524 | Odontophorus_balliviani | NADH dehydrogenase s |
| KC556517 Odontophorus columbianus | KC556517 | Odontophorus_columbianus | NADH dehydrogenase s |
| KC556515 Odontophorus strophium | KC556515 | Odontophorus_strophium | NADH dehydrogenase s |
| KC556513 Odontophorus melanonotus | KC556513 | Odontophorus_melanonotus | NADH dehydrogenase s |
| KC556512 Odontophorus hyperythrus | KC556512 | Odontophorus_hyperythrus | NADH dehydrogenase s |
| KC556507 Odontophorus melanotis | KC556507 | Odontophorus_melanotis | NADH dehydrogenase s |
| EF569482 Polyplectron inopinatum | EF569482 | Polyplectron_inopinatum | NADH dehydrogenase s |
| EF569481 Polyplectron napoleonis | EF569481 | Polyplectron_napoleonis | NADH dehydrogenase s |
| EF569480 Polyplectron chalcurum | EF569480 | Polyplectron_chalcurum | NADH dehydrogenase s |
| EF569479 Polyplectron bicalcaratum | EF569479 | Polyplectron_bicalcaratum | NADH dehydrogenase s |
| DQ768268 Polyplectron malacense | DQ768268 | Polyplectron_malacense | NADH dehydrogenase s |
| DQ768266 Polyplectron germaini | DQ768266 | Polyplectron_germaini | NADH dehydrogenase s |
| KC778823 Polyplectron katsumatae | KC778823 | Polyplectron_katsumatae | NADH dehydrogenase s |
| KR732830 Cyrtonyx montezumae | KR732830 | Cyrtonyx_montezumae | 12S ribosomal RNA |
| KR732829 Dactylortyx thoracicus | KR732829 | Dactylortyx_thoracicus | 12S ribosomal RNA |
| KR732828 Oreortyx pictus | KR732828 | Oreortyx_pictus | 12S ribosomal RNA |
| KR732827 Odontophorus erythrops | KR732827 | Odontophorus_erythrops | 12S ribosomal RNA |
| KR732826 Odontophorus gujanensis | KR732826 | Odontophorus_gujanensis | 12S ribosomal RNA |
| KR732825 Odontophorus stellatus | KR732825 | Odontophorus_stellatus | 12S ribosomal RNA |
| KR732824 Odontophorus capueira | KR732824 | Odontophorus_capueira | 12S ribosomal RNA |
| KR732823 Odontophorus speciosus | KR732823 | Odontophorus_speciosus | 12S ribosomal RNA |
| KR732822 Odontophorus leucolaemus | KR732822 | Odontophorus_leucolaemus | 12S ribosomal RNA |
| KR732821 Odontophorus balliviani | KR732821 | Odontophorus_balliviani | 12S ribosomal RNA |
| KR732820 Dendrortyx macroura | KR732820 | Dendrortyx_macroura | 12S ribosomal RNA |
| KR732819 Philortyx fasciatus | KR732819 | Philortyx_fasciatus | 12S ribosomal RNA |

Table 4: 4. Preliminary accession number table

| kingdom | phylum | class | order | family | genus | species |
|---|---|---|---|---|---|---|
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla | Callipepla californica |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla | Callipepla gambelii |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus | Colinus virginianus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus | Colinus cristatus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus | Colinus nigrogularis |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dendrortyx | Dendrortyx barbatus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Philortyx | Philortyx fasciatus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Rhynchortyx | Rhynchortyx cinctus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Cyrtonyx | Cyrtonyx montezumae |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Oreortyx | Oreortyx pictus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus leucolaemus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus speciosus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus erythrops |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus guttatus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus gujanensis |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus capueira |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus stellatus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dactylortyx | Dactylortyx thoracicus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dendrortyx | Dendrortyx macroura |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla | Callipepla squamata |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla | Callipepla douglasii |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus | Colinus leucopogon |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dendrortyx | Dendrortyx leucophrys |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Cyrtonyx | Cyrtonyx ocellatus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus balliviani |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus columbianus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus strophium |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus melanonotus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus hyperythrus |
| Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus | Odontophorus melanotis |
| Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron | Polyplectron inopinatum |
| Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron | Polyplectron napoleonis |
| Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron | Polyplectron chalcurum |
| Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron | Polyplectron bicalcaratum |
| Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron | Polyplectron malacense |
| Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron | Polyplectron germaini |
| Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron | Polyplectron katsumatae |

Table 5: 4. Preliminary accession number table

| species_names | kingdom | phylum | class | order | family | genus |
|---|---|---|---|---|---|---|
| Callipepla_californica | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla |
| Callipepla_douglasii | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla |
| Callipepla_gambelii | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla |
| Callipepla_squamata | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Callipepla |
| Colinus_cristatus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus |
| Colinus_leucopogon | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus |
| Colinus_nigrogularis | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus |
| Colinus_virginianus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Colinus |
| Cyrtonyx_montezumae | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Cyrtonyx |
| Cyrtonyx_ocellatus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Cyrtonyx |
| Dactylortyx_thoracicus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dactylortyx |
| Dendrortyx_barbatus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dendrortyx |
| Dendrortyx_leucophrys | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dendrortyx |
| Dendrortyx_macroura | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Dendrortyx |
| Odontophorus_balliviani | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_capueira | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_columbianus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_erythrops | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_gujanensis | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_guttatus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_hyperythrus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_leucolaemus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_melanonotus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_melanotis | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_speciosus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_stellatus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Odontophorus_strophium | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Odontophorus |
| Oreortyx_pictus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Oreortyx |
| Philortyx_fasciatus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Philortyx |
| Polyplectron_bicalcaratum | Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron |
| Polyplectron_chalcurum | Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron |
| Polyplectron_germaini | Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron |
| Polyplectron_inopinatum | Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron |
| Polyplectron_katsumatae | Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron |
| Polyplectron_malacense | Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron |
| Polyplectron_napoleonis | Animalia | Chordata | Aves | Galliformes | Phasianidae | Polyplectron |
| Rhynchortyx_cinctus | Animalia | Chordata | Aves | Galliformes | Odontophoridae | Rhynchortyx |