Handling Metadata in RNeXML

Carl Boettiger Scott Chamberlain Rutger Vos Hilmar Lapp

Writing NeXML metadata

The `add_basic_meta()` function takes as input an existing `nexml` object (like the other `add_` functions, if none is provided it will create one), and at the time of this writing any of the following parameters: `title`, `description`, `creator`, `pubdate`, `rights`, `publisher`, `citation`. Other metadata elements and corresponding parameters may be added in the future.

Load the packages and data:

```
library('RNeXML')
library('geiger')
data(bird.orders)
```

Create an `nexml` object for the phylogeny `bird.orders` and add appropriate metadata:

```
birds <- add_trees(bird.orders)
birds <- add_basic_meta(
  title = "Phylogeny of the Orders of Birds From Sibley and Ahlquist",

  description = "This data set describes the phylogenetic relationships of the
     orders of birds as reported by Sibley and Ahlquist (1990). Sibley
     and Ahlquist inferred this phylogeny from an extensive number of
     DNA/DNA hybridization experiments. The ``tapestry'' reported by
     these two authors (more than 1000 species out of the ca. 9000
     extant bird species) generated a lot of debates.

     The present tree is based on the relationships among orders. The
     branch lengths were calculated from the values of Delta T50H as
     found in Sibley and Ahlquist (1990, fig. 353).",

  citation = "Sibley, C. G. and Ahlquist, J. E. (1990) Phylogeny and
     classification of birds: a study in molecular evolution. New
     Haven: Yale University Press.",

  creator = "Sibley, C. G. and Ahlquist, J. E.",
    nexml=birds)
```

Instead of a literal string, citations can also be provided in R's `bibentry` type, which is the one in which R package citations are obtained:

```
birds <- add_basic_meta(citation = citation("ape"), nexml = birds)
```

A citation to a published paper with a Digital Object Identifier (DOI) can be provided in the form of the DOI, which the package `knitcitations` [@Boettiger_2014] can turn into a formatted citation using. As an example, to add the citation information of the paper that generated the `geospiza` phylogeny included in the `geiger` package:

```
library("knitcitations")
geiger_nex <- add_basic_meta(citation = bib_metadata("10.2307/2408428"))
```

Taxonomic identifiers

The `taxize_nexml()` function uses the R package `taxize` [@Chamberlain_2013] to check each taxon label against the NCBI database. If a unique match is found, a metadata annotation is added to the taxon providing the NCBI identification number to the taxonomic unit.

```
birds <- taxize_nexml(birds, "NCBI")
```

If no match is found, the user is warned to check for possible typographic errors in the taxonomic labels provided. If multiple matches are found, the user will be prompted to choose between them.

Custom metadata extensions

We can get a list of namespaces along with their prefixes from the `nexml` object:

```
prefixes <- get_namespaces(birds)
prefixes["dc"]
```

```
                                dc
"http://purl.org/dc/elements/1.1/"
```

We create a `meta` element containing this annotation using the `meta` function:

```
modified <- meta(property = "prism:modificationDate", content = "2013-10-04")
```

We can add this annotation to our existing `birds` NeXML file using the `add_meta()` function. Because we do not specify a level, it is added to the root node, referring to the NeXML file as a whole.

```
birds <- add_meta(modified, birds)
```

The built-in vocabularies are just the tip of the iceberg of established vocabularies. Here we add an annotation from the `skos` namespace which describes the history of where the data comes from:

```
history <- meta(property = "skos:historyNote",
  content = "Mapped from the bird.orders data in the ape package using RNeXML")
```

Because `skos` is not in the current namespace list, we add it with a url when adding this meta element. We also specify that this annotation be placed at the level of the `trees` sub-node in the NeXML file.

```
birds <- add_meta(history,
                  birds,
                  level = "trees",
                  namespaces = c(skos = "http://www.w3.org/2004/02/skos/core#"))
```

For finer control of the level at which a `meta` element is added, we will manipulate the `nexml` R object directly using S4 sub-setting, as shown in the supplement.

Much richer metadata annotation is possible. Later we illustrate how metadata annotation can be used to extend the base NeXML format to represent new forms of data while maintaining compatibility with any NeXML parser. The `RNeXML` package can be easily extended to support helper functions such as `taxize_nexml` to add additional metadata without imposing a large burden on the user.

Reading NeXML metadata

A call to the `nexml` object prints some metadata summarizing the data structure:

```
birds
```

```
A nexml object representing:
     1 phylogenetic tree blocks, where:
     block 1 contains 1 phylogenetic trees
     44 meta elements
     0 character matrices
     23 taxonomic units
 Taxa:   Struthioniformes, Tinamiformes, Craciformes, Galliformes, Anseriformes, Turniciformes ...

 NeXML generated by RNeXML using schema version: 0.9
 size: 370.7 Kb
```

We can extract all metadata pertaining to the NeXML document as a whole (annotations of the XML root node, `<nexml>`) with the command

```
meta <- get_metadata(birds)
```

This returns a named list of available metadata. We can see the kinds of metadata recorded from the names (showing the first 4):

```
names(meta)[1:4]
```

```
[1] "dc:title"                      "dc:creator"
[3] "dc:description"                "dcterms:bibliographicCitation"
```

and can ask for a particular element using the standard list sub-setting mechanism (i.e. either the name of an element or its numeric position),

```
meta[["dc:title"]]
```

```
[1] "Phylogeny of the Orders of Birds From Sibley and Ahlquist"
```

All metadata terms must belong to an explicit *namespace* or vocabulary that allows a computer to interpret the term precisely. The prefix (before the `:`) indicates to which vocabulary the term belongs, e.g. `dc` in this case. The `get_namespaces` function tells us the definition of the vocabulary using a link:

```
prefixes <- get_namespaces(birds)
prefixes["dc"]
```

```
                               dc
"http://purl.org/dc/elements/1.1/"
```

Common metadata can be accessed with a few dedicated functions:

```
get_citation(birds)
```

```
Sibley, C. G. and Ahlquist, J. E. (1990) Phylogeny and
     classification of birds: a study in molecular evolution. New
     Haven: Yale University Press. Paradis E, Claude J and Strimmer K (2004). "APE: analyses of
phylogenetics and evolution in R language." _Bioinformatics_,
*20*, pp. 289-290.
```

```
get_taxa(birds)
```

```
 [1] "Struthioniformes" "Tinamiformes"     "Craciformes"
 [4] "Galliformes"      "Anseriformes"     "Turniciformes"
 [7] "Piciformes"       "Galbuliformes"    "Bucerotiformes"
[10] "Upupiformes"      "Trogoniformes"    "Coraciiformes"
[13] "Coliiformes"      "Cuculiformes"     "Psittaciformes"
[16] "Apodiformes"      "Trochiliformes"   "Musophagiformes"
[19] "Strigiformes"     "Columbiformes"    "Gruiformes"
[22] "Ciconiiformes"    "Passeriformes"
```

Which returns text from the otu element labels, typically used to define taxonomic names, rather than text from explicit meta elements.

We can also access metadata at a specific level (or use `level=all` to extract all meta elements in a list). Here we show only the first few results:

```
otu_meta <- get_metadata(birds, level="otu")
otu_meta[1:4]
```

```
$`tc:toTaxon`
[1] "http://ncbi.nlm.nih.gov/taxonomy/8798"

$`tc:toTaxon`
[1] "http://ncbi.nlm.nih.gov/taxonomy/8802"

$`tc:toTaxon`
[1] "http://ncbi.nlm.nih.gov/taxonomy/8976"

$`tc:toTaxon`
[1] "http://ncbi.nlm.nih.gov/taxonomy/8976"
```