Introduction to phyx.js

Written by Gaurav Vaidya. Last updated August 24, 2021

This tutorial provides an introduction to the phyx.js library, and shows you how it can be used to read a Phyx file, check it for validity, examine phyloreferences, phylogenies and specifiers, and describe how to convert the file into RDF.

Starting with Phyx files

Phyx files are digitized clade definitions in a JSON-LD format. While you can use an editor like Klados to create Phyx files, you can also write one by yourself as a JSON document.

```
 1  var alligatoridae_brochu2003 = {
 2      "@context": "https://www.phyloref.org/phyx.js/context/v1.0.0/phyx.json",
 3
 4      // Phylogeny from Brochu 2003: https://doi.org/10.1146/annurev.earth
 5      //    .31.100901.141308
        "phylogenies": [{
 6          "newick": "(Parasuchia,(rauisuchians,Aetosauria,(sphenosuchians,(
                protosuchians,(mesosuchians,(Hylaeochampsa,Aegyptosuchus,
                Stomatosuchus,(Allodaposuchus,('Gavialis gangeticus',(('Diplocynodon
                 ratelii',('Alligator mississippiensis','Caiman crocodilus')
                Alligatoridae)Alligatoroidea,('Tomistoma schlegelii',('Osteolaemus
                tetraspis','Crocodylus niloticus')Crocodylinae)Crocodylidae)
                Brevirostres)Crocodylia))Eusuchia)Mesoeucrocodylia)Crocodyliformes)
                Crocodylomorpha));",
 7          "source": {
 8              "type": "article",
 9              "title": "Phylogenetic approaches toward crocodylian history",
10              "authors": [{
11                  "firstname": "Christopher",
12                  "middlename": "A.",
13                  "lastname": "Brochu"
14              }],
15              "year": 2003,
16              "figure": "1",
17              "identifier": [{
18                  "type": "doi",
19                  "id": "10.1146/annurev.earth.31.100901.141308"
20              }],
21              "journal": {
22                  "name": "Annual Review of Earth and Planetary Sciences",
23                  "volume": "31",
24                  "pages": "357--397",
25                  "identifier": [{ "type": "eISSN", "id": "1545-4495" }]
26              }
27          }
28      }],
29
30      // Clade definition from Brochu 2003: Alligatoridae
```

```
31      "phylorefs": [{
32          "label": "Alligatoridae",
33          "scientificNameAuthorship": { "bibliographicCitation": "(Cuvier 1807)
                " },
34          "phylorefType": "phyloref:PhyloreferenceUsingMinimumClade",
35          "definition": "Alligatoridae (Cuvier 1807).\n\nLast common ancestor
                of Alligator mississippiensis and Caiman crocodilus and all of its
                descendents.",
36          "definitionSource": {
37              "bibliographicCitation": "Brochu (2003) Phylogenetic approaches
                    toward crocodylian history. Annual Review of Earth and
                    Planetary Sciences 31:1, 357-397. doi: 10.1146/annurev.earth
                    .31.100901.141308"
38          },
39          "internalSpecifiers": [
40              {
41                  "@type": "http://rs.tdwg.org/ontology/voc/TaxonConcept#
                        TaxonConcept",
42                  "hasName": {
43                      "@type": "http://rs.tdwg.org/ontology/voc/TaxonName#
                            TaxonName",
44                      "nomenclaturalCode": "http://rs.tdwg.org/ontology/voc/
                            TaxonName#ICZN",
45                      "label": "Caiman crocodilus Linnaeus, 1758",
46                      "nameComplete": "Caiman crocodilus",
47                      "genusPart": "Caiman",
48                      "specificEpithet": "crocodilus"
49                  }
50              }, {
51                  "@type": "http://rs.tdwg.org/ontology/voc/TaxonConcept#
                        TaxonConcept",
52                  "hasName": {
53                      "@type": "http://rs.tdwg.org/ontology/voc/TaxonName#
                            TaxonName",
54                      "nomenclaturalCode": "http://rs.tdwg.org/ontology/voc/
                            TaxonName#ICZN",
55                      "label": "Alligator mississippiensis (Daudin, 1802)",
56                      "nameComplete": "Alligator mississippiensis",
57                      "genusPart": "Alligator",
58                      "specificEpithet": "mississippiensis"
59                  }
60              }
61          ]
62      }]
63  }
```

Validating a Phyx document using JSON Schema

We publish a JSON Schema with phyx.js, which can be used to validate that a Phyx document is correctly formed. We use Ajv, a JSON Schema validator for JavaScript. Note that we use the copy of the context file that is included in this repository, but you can also download it from our website.

```
 1  var fs = require('fs');
 2  var Ajv = require('ajv');
 3
 4  // Configure Ajv.
 5  var ajv = new Ajv({
 6    allErrors: true, // Display all error messages, not just the first.
 7  });
 8
 9  // We use the JSON Schema included with this repository, but you can download
        the
10  // Phyx JSON Schema from https://www.phyloref.org/phyx.js/context/v1.0.0/schema
        .json
11  var validator = ajv.compile(JSON.parse(fs.readFileSync('../docs/context/v1.0.0/
        schema.json')));
12
13  // Attempt to validate the Brochu 2003 example file.
14  var result = validator(alligatoridae_brochu2003);
15  console.log(`Is alligatoridae_brochu2003 valid? ${result}`);
16  console.log('Errors:', validator.errors || 'none');
17
18  // Let's make an invalid copy of the Brochu 2003 example file to make sure this
         is working.
19  var alligatoridae_brochu2003copy = {...alligatoridae_brochu2003};
20  delete alligatoridae_brochu2003copy['@context'];  // @context is required for
        successful validation
21
22  var result = validator(alligatoridae_brochu2003copy);
23  console.log(`Is alligatoridae_brochu2003copy valid? ${result}`);
24  console.log('Errors:', validator.errors);
```

```
 1  Is alligatoridae_brochu2003 valid? true
 2  Errors: none
 3  Is alligatoridae_brochu2003copy valid? false
 4  Errors: [
 5    {
 6      keyword: 'required',
 7      dataPath: '',
 8      schemaPath: '#/required',
 9      params: { missingProperty: '@context' },
10      message: "should have required property '@context'"
11    }
12  ]
```

Examining phyloreferences, taxonomic units and taxon concepts

The phyx.js library was built in order to simplify the process of working with individual components of Phyx documents, and to facilitate the conversion of a Phyx document into OWL. The library consists of a series of wrappers, each of which wraps part of the document. For example, we can wrap every specifier that is a taxonomic unit using the TaxonomicUnitWrapper.

This provides a number of convenience methods: for example, .internalSpecifiers and .

`externalSpecifiers` will always return lists, whether or not these are defined in the underlying phyloreference (if undefined, the methods return empty lists). There is also a `.specifiers` method that lists both internal and external specifiers.

Furthermore, taxonomic units that are taxon concepts can be wrapped by a TaxonConceptWrapper, which have methods for accessing the "complete name" (i.e. the monomial, binomial or trinomial name) and the nomenclatural code.

```
 1  // Load the Phyx library.
 2  var phyx = require('..');
 3
 4  // List all the phyloreferences along with their specifiers.
 5  alligatoridae_brochu2003.phylorefs.forEach(phyloref => {
 6      let wrappedPhyloref = new phyx.PhylorefWrapper(phyloref);
 7
 8      console.log(wrappedPhyloref.label);
 9
10      // Extract the "complete name" and the nomenclatural code short name for
            each specifier that is a taxonomic unit.
11      wrappedPhyloref.internalSpecifiers.forEach(specifier => {
12          let wrappedSpecifier = new phyx.TaxonomicUnitWrapper(specifier);
13          if (wrappedSpecifier.taxonConcept) {
14              let wrappedTaxonConcept = new phyx.TaxonConceptWrapper(
                    wrappedSpecifier.taxonConcept);
15              console.log(` - Internal: ${wrappedTaxonConcept.nameComplete} (${
                    wrappedTaxonConcept.nomenCodeDetails.shortName})`);
16          } else {
17              console.log(` - Internal: ${wrappedSpecifier.label}`);
18          }
19      });
20
21      // Note that the phyloref doesn't have an `externalSpecifiers` key, but the
            wrapper provides it as an empty list
22      // for ease of use.
23      wrappedPhyloref.externalSpecifiers.forEach(specifier => {
24          let wrappedSpecifier = new phyx.TaxonomicUnitWrapper(specifier);
25          if (wrappedSpecifier.taxonConcept) {
26              let wrappedTaxonConcept = new phyx.TaxonConceptWrapper(
                    wrappedSpecifier.taxonConcept);
27              console.log(` - External: ${wrappedTaxonConcept.nameComplete} (${
                    wrappedTaxonConcept.nomenCodeDetails.shortName})`);
28          } else {
29              console.log(` - External: ${wrappedSpecifier.label}`);
30          }
31      });
32
33      console.log();
34  });
```

```
 1  Alligatoridae
 2   - Internal: Caiman crocodilus (ICZN)
 3   - Internal: Alligator mississippiensis (ICZN)
```

Examining phylogenies

Phylogenies are stored in JSON files as Newick strings, but the PhylogenyWrapper can be used to look at internal and terminal node labels and to translate the Newick string into a JSON structure for use by downstream programs.

```
 1  var phylogeny = alligatoridae_brochu2003.phylogenies[0];
 2  console.log(`The phylogeny is represented by the Newick string: ${phylogeny.
        newick}`);
 3  console.log();
 4
 5  // Display internal and external nodes.
 6  var wrappedPhylogeny = new phyx.PhylogenyWrapper(phylogeny);
 7  console.log(`This consists of the following nodes:\n - Internal nodes: ${
        wrappedPhylogeny.getNodeLabels('internal').join(', ')}`);
 8  console.log(` - Terminal nodes: ${wrappedPhylogeny.getNodeLabels('terminal').
        join(', ')}`);
 9  console.log();
10
11  // Convert the Newick string into a JSON structure for examination.
12  console.log(`Newick string as a JSON structure: ${JSON.stringify(phyx.
        PhylogenyWrapper.getParsedNewick(phylogeny.newick), undefined, 2)}`);
13  console.log();
```

```
 1  The phylogeny is represented by the Newick string: (Parasuchia,(rauisuchians,
        Aetosauria,(sphenosuchians,(protosuchians,(mesosuchians,(Hylaeochampsa,
        Aegyptosuchus,Stomatosuchus,(Allodaposuchus,('Gavialis gangeticus',(('
        Diplocynodon ratelii','Alligator mississippiensis','Caiman crocodilus')
        Alligatoridae)Alligatoroidea,('Tomistoma schlegelii',('Osteolaemus tetraspis
        ','Crocodylus niloticus')Crocodylinae)Crocodylidae)Brevirostres)Crocodylia))
        Eusuchia)Mesoeucrocodylia)Crocodyliformes)Crocodylomorpha));
 2
 3  This consists of the following nodes:
 4   - Internal nodes: Alligatoridae, Alligatoroidea, Crocodylinae, Crocodylidae,
         Brevirostres, Crocodylia, Eusuchia, Mesoeucrocodylia, Crocodyliformes,
         Crocodylomorpha
 5   - Terminal nodes: Parasuchia, rauisuchians, Aetosauria, sphenosuchians,
         protosuchians, mesosuchians, Hylaeochampsa, Aegyptosuchus, Stomatosuchus,
         Allodaposuchus, Gavialis gangeticus, Diplocynodon ratelii, Alligator
         mississippiensis, Caiman crocodilus, Tomistoma schlegelii, Osteolaemus
         tetraspis, Crocodylus niloticus
 6
 7  Newick string as a JSON structure: {
 8    "json": {
 9      "children": [
10        {
11          "children": [
12            {
13              "label": "Crocodylomorpha",
14              "children": [
15                {
16                  "label": "Crocodyliformes",
17                  "children": [
18                    {
```

```
19                    "label": "Mesoeucrocodylia",
20                    "children": [
21                      {
22                        "label": "Eusuchia",
23                        "children": [
24                          {
25                            "children": [
26                              {
27                                "label": "Crocodylia",
28                                "children": [
29                                  {
30                                    "label": "Brevirostres",
31                                    "children": [
32                                      {
33                                        "label": "Crocodylidae",
34                                        "children": [
35                                          {
36                                            "label": "Crocodylinae",
37                                            "children": [
38                                              {
39                                                "label": "Crocodylus niloticus"
                                                  ,
40                                                "name": "Crocodylus niloticus"
41                                              },
42                                              {
43                                                "label": "Osteolaemus tetraspis
                                                  ",
44                                                "name": "Osteolaemus tetraspis"
45                                              }
46                                            ],
47                                            "name": "Crocodylinae"
48                                          },
49                                          {
50                                            "label": "Tomistoma schlegelii",
51                                            "name": "Tomistoma schlegelii"
52                                          }
53                                        ],
54                                        "name": "Crocodylidae"
55                                      },
56                                      {
57                                        "label": "Alligatoroidea",
58                                        "children": [
59                                          {
60                                            "label": "Alligatoridae",
61                                            "children": [
62                                              {
63                                                "label": "Caiman crocodilus",
64                                                "name": "Caiman crocodilus"
65                                              },
66                                              {
67                                                "label": "Alligator
                                                  mississippiensis",
68                                                "name": "Alligator
                                                  mississippiensis"
69                                              }
70                                            ],
```

```
 71                                          "name": "Alligatoridae"
 72                                        },
 73                                        {
 74                                          "label": "Diplocynodon ratelii",
 75                                          "name": "Diplocynodon ratelii"
 76                                        }
 77                                      ],
 78                                      "name": "Alligatoroidea"
 79                                    }
 80                                  ],
 81                                  "name": "Brevirostres"
 82                                },
 83                                {
 84                                  "label": "Gavialis gangeticus",
 85                                  "name": "Gavialis gangeticus"
 86                                }
 87                              ],
 88                              "name": "Crocodylia"
 89                            },
 90                            {
 91                              "label": "Allodaposuchus",
 92                              "name": "Allodaposuchus"
 93                            }
 94                          ]
 95                        },
 96                        {
 97                          "label": "Stomatosuchus",
 98                          "name": "Stomatosuchus"
 99                        },
100                        {
101                          "label": "Aegyptosuchus",
102                          "name": "Aegyptosuchus"
103                        },
104                        {
105                          "label": "Hylaeochampsa",
106                          "name": "Hylaeochampsa"
107                        }
108                      ],
109                      "name": "Eusuchia"
110                    },
111                    {
112                      "label": "mesosuchians",
113                      "name": "mesosuchians"
114                    }
115                  ],
116                  "name": "Mesoeucrocodylia"
117                },
118                {
119                  "label": "protosuchians",
120                  "name": "protosuchians"
121                }
122              ],
123              "name": "Crocodyliformes"
124            },
125            {
126              "label": "sphenosuchians",
```

```
127                    "name": "sphenosuchians"
128                  }
129                ],
130                "name": "Crocodylomorpha"
131              },
132              {
133                "label": "Aetosauria",
134                "name": "Aetosauria"
135              },
136              {
137                "label": "rauisuchians",
138                "name": "rauisuchians"
139              }
140            ]
141          },
142          {
143            "label": "Parasuchia",
144            "name": "Parasuchia"
145          }
146        ]
147      }
148  }
```

Accessing citations

Another example of a wrapper that can be used for wrapping a part of a Phyx file is the CitationWrapper. This can be used to wrap citations anywhere in the Phyx file to provide a full bibliographic citation for the citation.

```
1  var wrappedPhylogenyCitation = new phyx.CitationWrapper(
       alligatoridae_brochu2003.phylogenies[0].source);
2  console.log(`The source of the phylogeny in this Phyx document is: ${
       wrappedPhylogenyCitation.toString()}`);
3
4  var wrappedPhylorefCitation = new phyx.CitationWrapper(alligatoridae_brochu2003
       .phylorefs[0].definitionSource);
5  console.log(`The definition source of the phyloreference in this Phyx document
       is: ${wrappedPhylorefCitation.toString()}`);
```

```
1  The source of the phylogeny in this Phyx document is: Christopher A. Brochu
       (2003) Phylogenetic approaches toward crocodylian history Annual Review of
       Earth and Planetary Sciences 31:357--397  fig 1 doi: 10.1146/annurev.earth
       .31.100901.141308
2  The definition source of the phyloreference in this Phyx document is: Brochu
       (2003) Phylogenetic approaches toward crocodylian history. Annual Review of
       Earth and Planetary Sciences 31:1, 357-397. doi: 10.1146/annurev.earth
       .31.100901.141308
```

Converting a Phyx document into OWL

A Phyx document (which is in JSON-LD format) can be converted into OWL/RDF in the form of N-Quads by using the `PhyxWrapper` to wrap the entire Phyx document. A base URL can be specified.

```
1  nQuads = new phyx.PhyxWrapper(alligatoridae_brochu2003).toRDF('http://example.
       org/phyx-tutorial#');
2  nQuads.then(nq => console.log(nq.slice(0, 2500) + '...'));
```

```
1  <http://example.org/phyx-tutorial#phylogeny0> <http://ontology.phyloref.org/
       phyloref.owl#newick_expression> "(Parasuchia,(rauisuchians,Aetosauria,(
       sphenosuchians,(protosuchians,(mesosuchians,(Hylaeochampsa,Aegyptosuchus,
       Stomatosuchus,(Allodaposuchus,('Gavialis gangeticus',(('Diplocynodon ratelii
       ',('Alligator mississippiensis','Caiman crocodilus')Alligatoridae)
       Alligatoroidea,('Tomistoma schlegelii',('Osteolaemus tetraspis','Crocodylus
       niloticus')Crocodylinae)Crocodylidae)Brevirostres)Crocodylia))Eusuchia)
       Mesoeucrocodylia)Crocodyliformes)Crocodylomorpha));" .
2  <http://example.org/phyx-tutorial#phylogeny0> <http://purl.obolibrary.org/obo/
       CDAO_0000148> <http://example.org/phyx-tutorial#phylogeny0_node0> .
3  <http://example.org/phyx-tutorial#phylogeny0> <http://purl.org/dc/terms/source>
       _:b165 .
4  <http://example.org/phyx-tutorial#phylogeny0> <http://www.w3.org/1999/02/22-rdf
       -syntax-ns#type> <http://ontology.phyloref.org/phyloref.owl#
       ReferencePhylogenyEvidence> .
5  <http://example.org/phyx-tutorial#phylogeny0> <http://www.w3.org/2000/01/rdf-
       schema#isDefinedBy> _:b0 .
6  <http://example.org/phyx-tutorial#phylogeny0_node0> <http://purl.obolibrary.org
       /obo/CDAO_0000149> <http://example.org/phyx-tutorial#phylogeny0_node1> .
7  <http://example.org/phyx-tutorial#phylogeny0_node0> <http://purl.obolibrary.org
       /obo/CDAO_0000149> <http://example.org/phyx-tutorial#phylogeny0_node29> .
8  <http://example.org/phyx-tutorial#phylogeny0_node0> <http://purl.obolibrary.org
       /obo/CDAO_0000200> <http://example.org/phyx-tutorial#phylogeny0> .
9  <http://example.org/phyx-tutorial#phylogeny0_node0> <http://www.w3.org
       /1999/02/22-rdf-syntax-ns#type> <http://purl.obolibrary.org/obo/CDAO_0000140
       > .
10 <http://example.org/phyx-tutorial#phylogeny0_node10> <http://ontology.phyloref.
       org/phyloref.owl#has_Sibling> <http://example.org/phyx-tutorial#
       phylogeny0_node13> .
11 <http://example.org/phyx-tutorial#phylogeny0_node10> <http://purl.obolibrary.
       org/obo/CDAO_0000149> <http://example.org/phyx-tutorial#phylogeny0_node11> .
12 <http://example.org/phyx-tutorial#phylogeny0_node10> <http://purl.obolibrary.
       org/obo/CDAO_0000149> <http://example.org/phyx-tutorial#phylogeny0_node12> .
13 <http://example.org/phyx-tutorial#phylogeny0_node10> <http://purl.obolibrary.
       org/obo/CDAO_0000179> <http://example.org/phyx-tutorial#phylogeny0_node9> .
14 <http://example.org/phyx-tutorial#phylogeny0_node10> <http://purl.obolibrary.
       org/obo/CDAO_0000187> _:b50 .
15 <http://example.org/phyx-tutorial#phylogeny0_node10> <http://purl.obolibrary.
       org/obo/CDAO_00002...
```

```
1  // We can convert these N-Quads into RDF/Turtle for easier interpretation.
2  // We only convert a few N-Quads as a demonstration.
3  var N3 = require('n3');
4  var { Readable } = require("stream")
5
```

```
 6  nQuads.then(nq => {
 7      someNqs = nq.slice(0, 1071)
 8
 9      var streamParser = new N3.StreamParser(),
10      inputStream = Readable.from([someNqs]),
11      streamWriter = new N3.StreamWriter();
12
13      inputStream.pipe(streamParser);
14      streamParser.pipe(streamWriter);
15      streamWriter.pipe(process.stdout);
16  });
17
18  undefined;
```

Navigating a Phyx document as a JSON file

Most phyx.js wrappers have been designed to help interpret the more complex parts of a Phyx file, such as the phyloreferences, specifiers, phylogenies, citations and the entire Phyx document. However, since every Phyx document is also a JSON document, much of the information in the Phyx document can be accessed sufficiently easily using standard JSON libraries. In some cases, as in the demonstration below, this requires more complex code than would be necessary by using the phyx.js wrappers.

```
 1  // List all the phylorefs in a Phyx document.
 2  alligatoridae_brochu2003copy.phylorefs.forEach((phyloref, index) => {
 3      console.log(`- Phyloref ${index + 1}. ${phyloref.label}:`);
 4      (phyloref.internalSpecifiers || []).forEach(specifier => {
 5          console.log(`  - Internal specifier: ${(specifier.hasName || {}).
              nameComplete}`);
 6      });
 7      (phyloref.externalSpecifiers || []).forEach(specifier => {
 8          console.log(`  - External specifier: ${(specifier.hasName || {}).
              nameComplete}`);
 9      });
10      console.log();
11  });
```

```
 1  <http://example.org/phyx-tutorial#phylogeny0> <http://ontology.phyloref.org/
        phyloref.owl#newick_expression> "(Parasuchia,(rauisuchians,Aetosauria,(
        sphenosuchians,(protosuchians,(mesosuchians,(Hylaeochampsa,Aegyptosuchus,
        Stomatosuchus,(Allodaposuchus,('Gavialis gangeticus',(('Diplocynodon ratelii
        ',('Alligator mississippiensis','Caiman crocodilus')Alligatoridae)
        Alligatoroidea,('Tomistoma schlegelii',('Osteolaemus tetraspis','Crocodylus
        niloticus')Crocodylinae)Crocodylidae)Brevirostres)Crocodylia))Eusuchia)
        Mesoeucrocodylia)Crocodyliformes)Crocodylomorpha));";
 2      <http://purl.obolibrary.org/obo/CDAO_0000148> <http://example.org/phyx-
          tutorial#phylogeny0_node0>;
 3      <http://purl.org/dc/terms/source> _:b0_b165;
 4      a <http://ontology.phyloref.org/phyloref.owl#ReferencePhylogenyEvidence>;
 5      <http://www.w3.org/2000/01/rdf-schema#isDefinedBy> _:b0_b0.
 6  - Phyloref 1. Alligatoridae:
 7    - Internal specifier: Caiman crocodilus
```

```
8     - Internal specifier: Alligator mississippiensis
```

Looking up phyloreferences on the Open Tree of Life

An included script, `resolve.js`, can be used to resolve a phyloreference on the Open Tree of Life. This can be executed from the command line by running:

```
1  $ npm run resolve test/examples/correct/brochu_2003.json
```

However, this script can also be invoked from within Node.js.

```
1  var child_process = require('child_process');
2
3  child = child_process.spawnSync('../bin/resolve.js', ['../test/examples/correct
       /brochu_2003.json'], {
4    encoding: 'utf-8',
5    stdio: 'pipe',
6  });
7  results = JSON.parse(child.output.join('\n'));
8
9  Object.keys(results).forEach(key => {
10     console.log(`- ${key}:`);
11     values = results[key];
12     values.forEach(value => {
13         if('status' in value) {
14             resolved = value['resolved'];
15             console.log(`  - Resolved ${value['cladeType']} phylorefence`);
16             console.log(`    to: ${resolved['@id']} (label: ${resolved['label'
                   ]})`);
17         } else if ('error' in value) {
18             console.log(`  - Could not resolve: ${value['error']}`);
19         } else {
20             console.log(`  - Unable to interpret: ${JSON.stringify(value,
                   undefined, 2)}`);
21         }
22     })
23  });
```

```
1  - Alligatoridae:
2    - Resolved minimum phylorefence
3      to: https://tree.opentreeoflife.org/opentree/argus/opentree13.4@ott195670 (
           label: Alligatoridae)
4  - Alligatorinae:
5    - Resolved maximum phylorefence
6      to: https://tree.opentreeoflife.org/opentree/argus/opentree13.4@ott151255 (
           label: Alligatorinae)
7  - Caimaninae:
8    - Resolved maximum phylorefence
9      to: https://tree.opentreeoflife.org/opentree/argus/opentree13.4@ott195671 (
           label: Caimaninae)
10  - Crocodyloidea:
11    - Resolved maximum phylorefence
```

```
12        to: https://tree.opentreeoflife.org/opentree/argus/opentree13.4@ott335582 (
             label: Crocodylidae)
13  - Crocodylidae:
14    - Resolved minimum phylorefence
15        to: https://tree.opentreeoflife.org/opentree/argus/opentree13.4@ott1092501
             (label: Longirostres)
16  - Diplocynodontinae:
17    - Could not resolve: no_mrca_found:400
```

About this notebook

This document was created as a Jupyter Notebook, and the source file is available in our GitHub repository. We recommend installing Jupyterlab via Homebrew on Mac, but other installation options are available. Once Jupyter Notebook is set up, you should be able to open this notebook for editing by running `jupyter notebook Introduction\ to\ phyx.js.ipynb` from the command line.

We use IJavascript to use Javascript as a kernel in Jupyter Notebook. If you would like to edit this notebook, you will need to install this as well.