The pipeline GitHub version tutorial

This pipeline transforms a character matrix obtained from Phenoscape Knowledge base (KB) into a version that can be efficiently integrated with an Open Tree phylogeny.

User Inputs

There two user inputs required for the pipeline

1. A character matrix downloaded from Phenoscape KB
2. Open tree phylogeny downloaded from Open Tree of Life

The specialty of this pipeline is that it can handle large matrices and phylogenies. For instance, this tutorial is conducted for Teleostei phylogeny, which has 38,830 taxa. This pipeline’s large scale data handling capability was demonstrated in Jackson paper

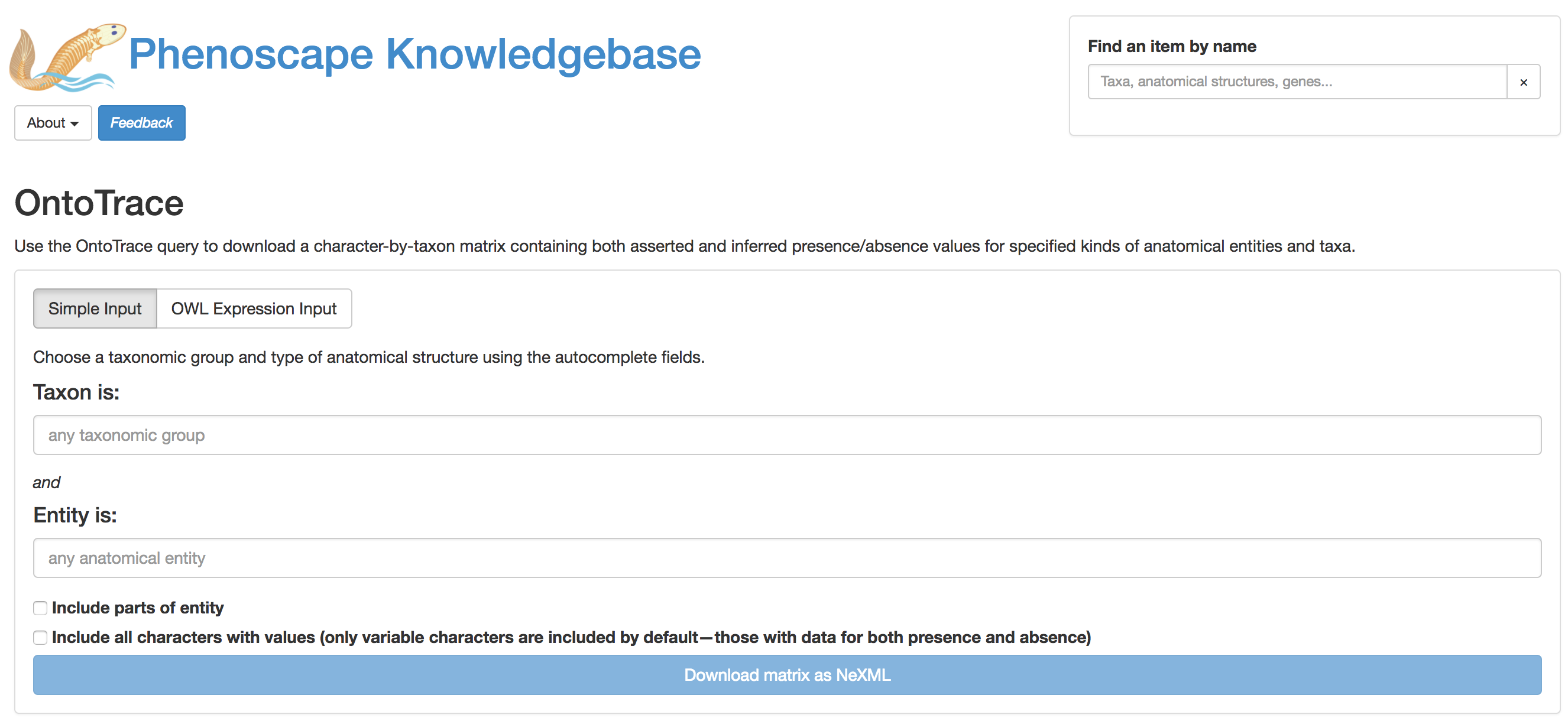
Phenoscape Knowledge base (<http://phenoscape.org/> )

Phenoscape KB is an online repository, which stores ontology annotated phenotypic data collected from published character matrices and some monographic treatments. The database can be queried using Onto Trace tool, which is a downloadable matrix generation tool that generates synthetic morphological supermatrices in NeXML format.

To download the NeXML matrix click on the link below.

<http://kb.phenoscape.org/#/ontotrace>

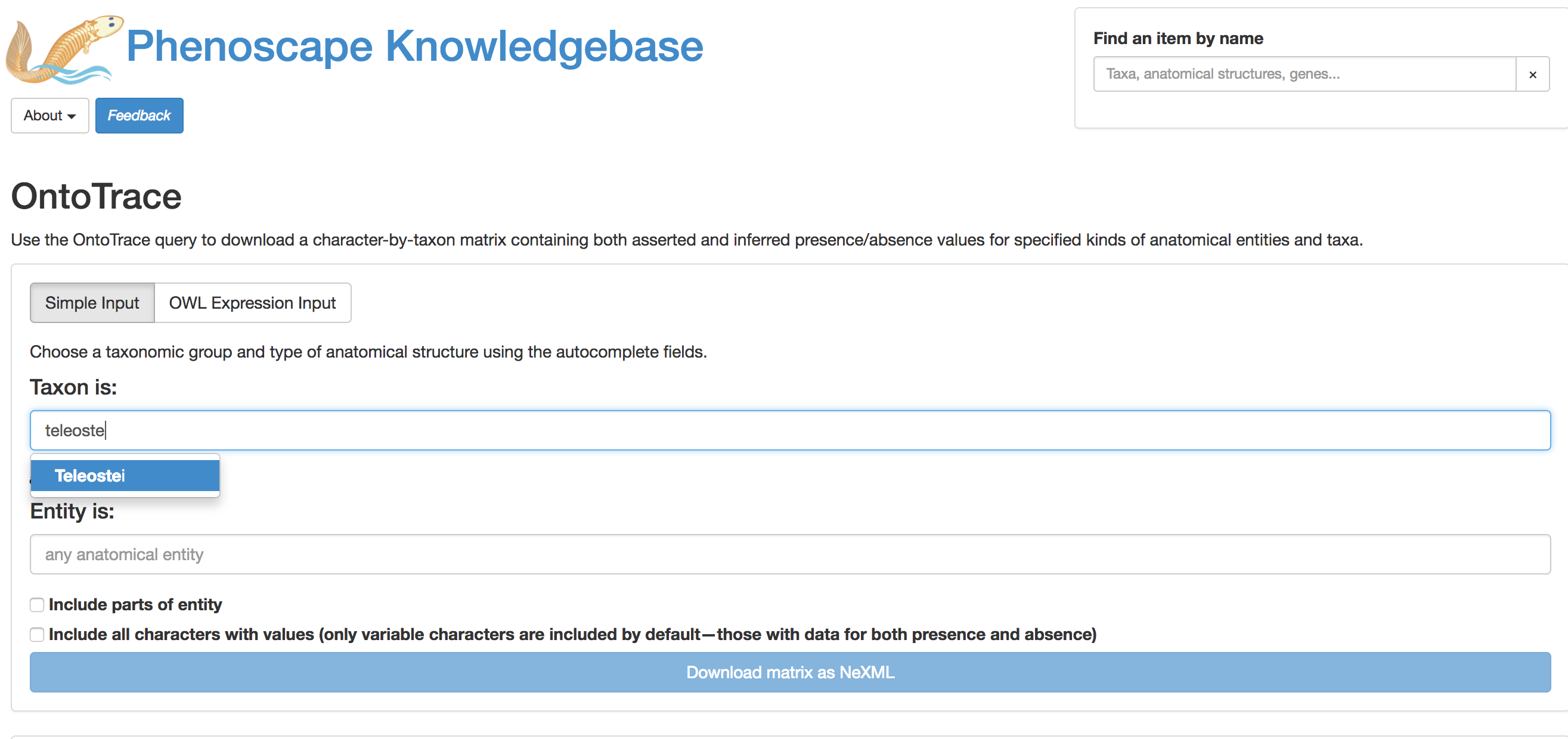
This will take you to the Onto Trace web interface represented in the following image



You can use the interface to download a presence/absence matrix for a given character and a given taxonomic group.

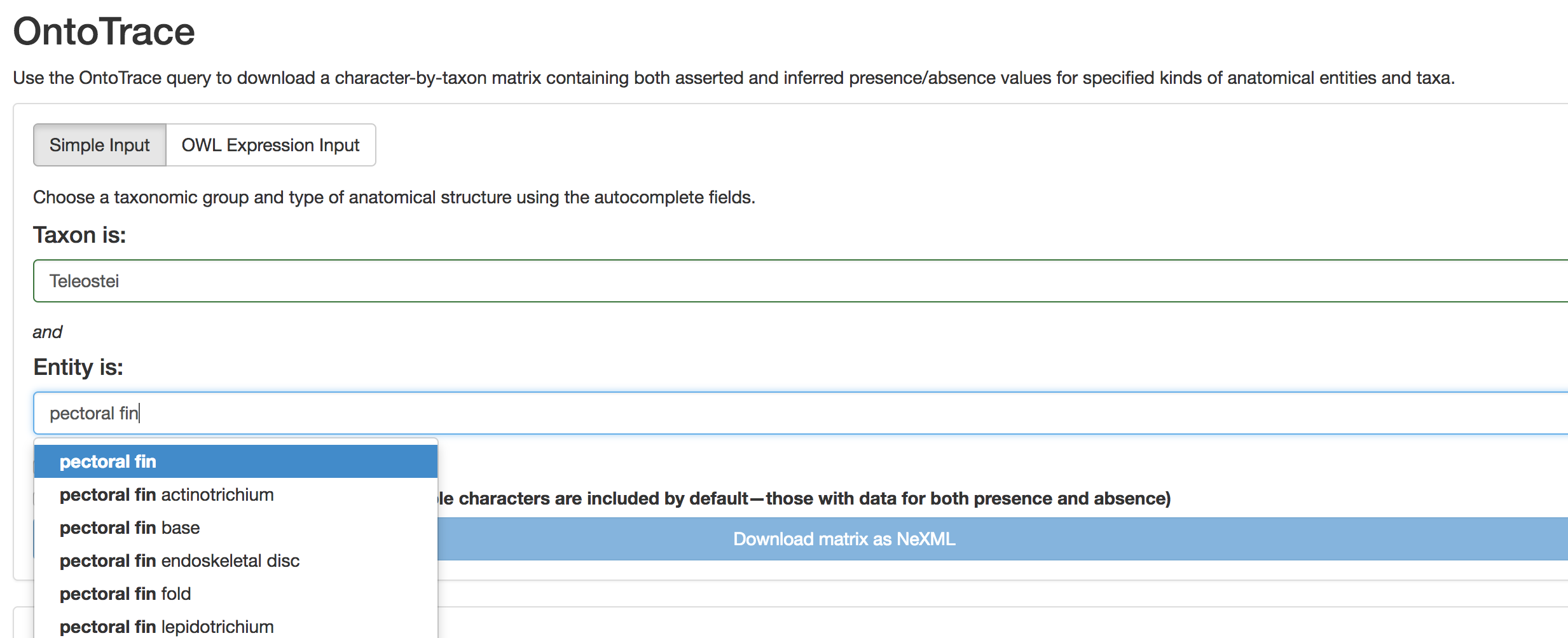
For example, let’s say we want to download pectoral fin character matrix for *Teleostei*, which is the largest ray-finned fish group.

First search for Teleostei in the empty field below the ‘Taxon is’ text.



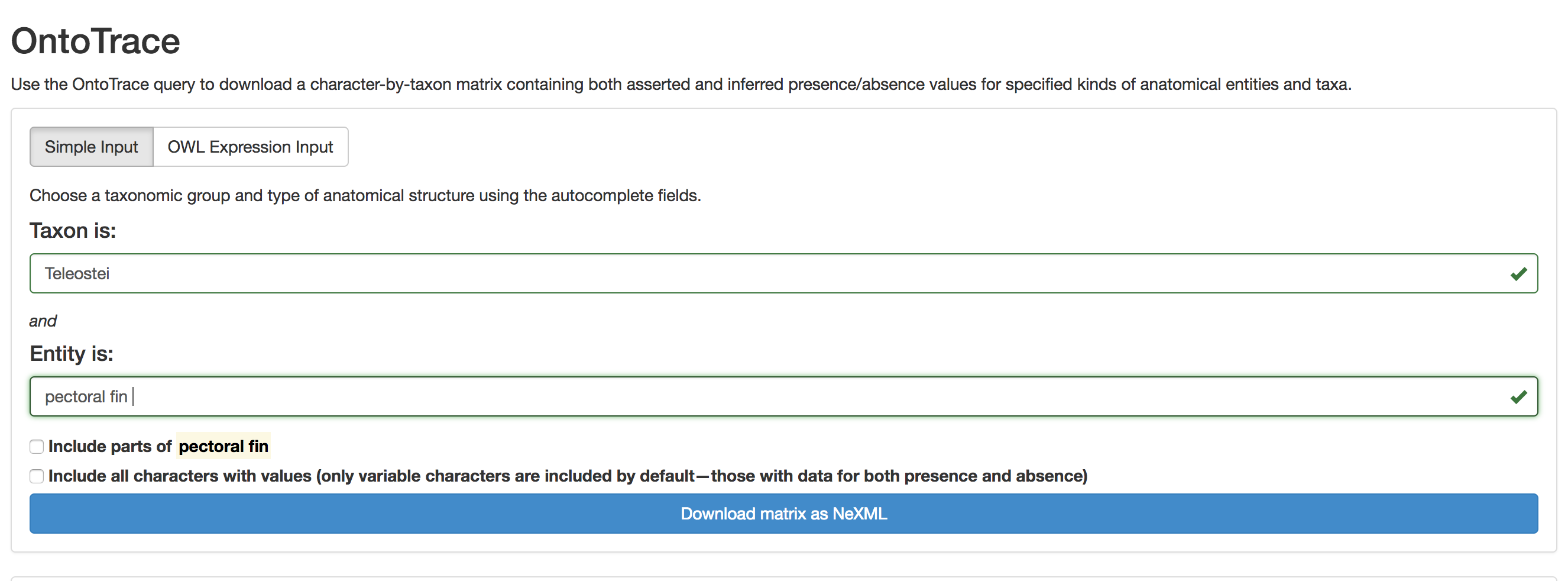
The KB will automatically recommend the taxa name when you start typing.

Next type the phenotypic entity/ character below the ‘Entity is’ field



Again, it will suggest the character entities stored in the KB. For our example, we are interested in pectoral fin, therefore, select pectoral fin.

Finally, click the Download matrix as NeXML button at the bottom of the interface



This will download a character matrix for pectoral fin in NeXML format. However, the downloaded file will be named as ‘ontotrace.xml’. You can rename it to something meaningful like ‘teleostei\_pectoral\_fin.xml’.

If you open the xml file in a text editor it will look like this.



This completes obtaining the first input file (character matrix). Now, we need the second file which is the phylogenetic tree.

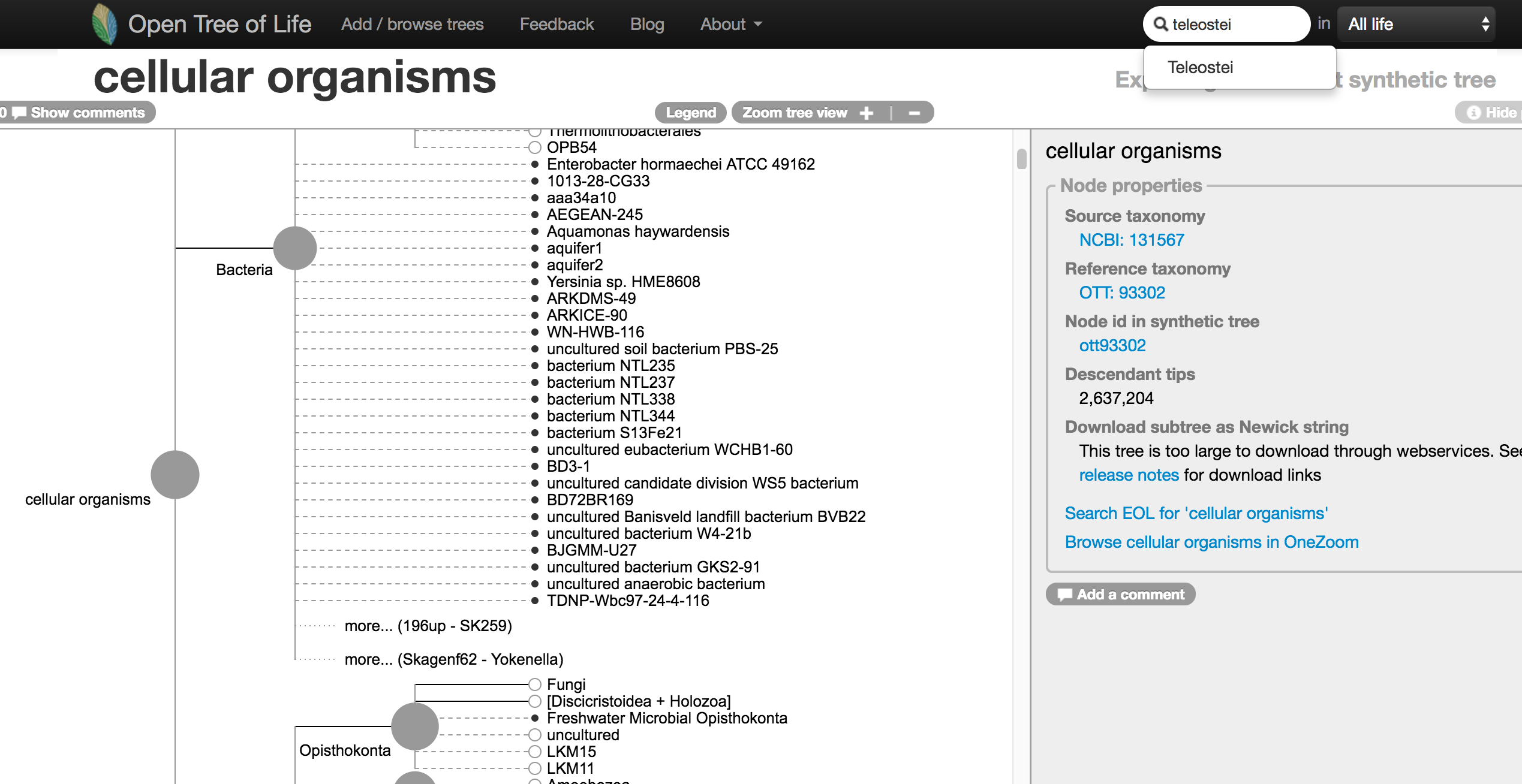
Open Tree of Life ([https://tree.opentreeoflife.org/](https://tree.opentreeoflife.org/opentree/argus/opentree9.1@ott93302) )

The Open Tree of Life dynamically constructs a tree by synthesizing published phylogenies along with taxonomic data using the ‘propinquity’ supertree pipeline. You will be able to download a comprehensive species-level tree for any taxon.

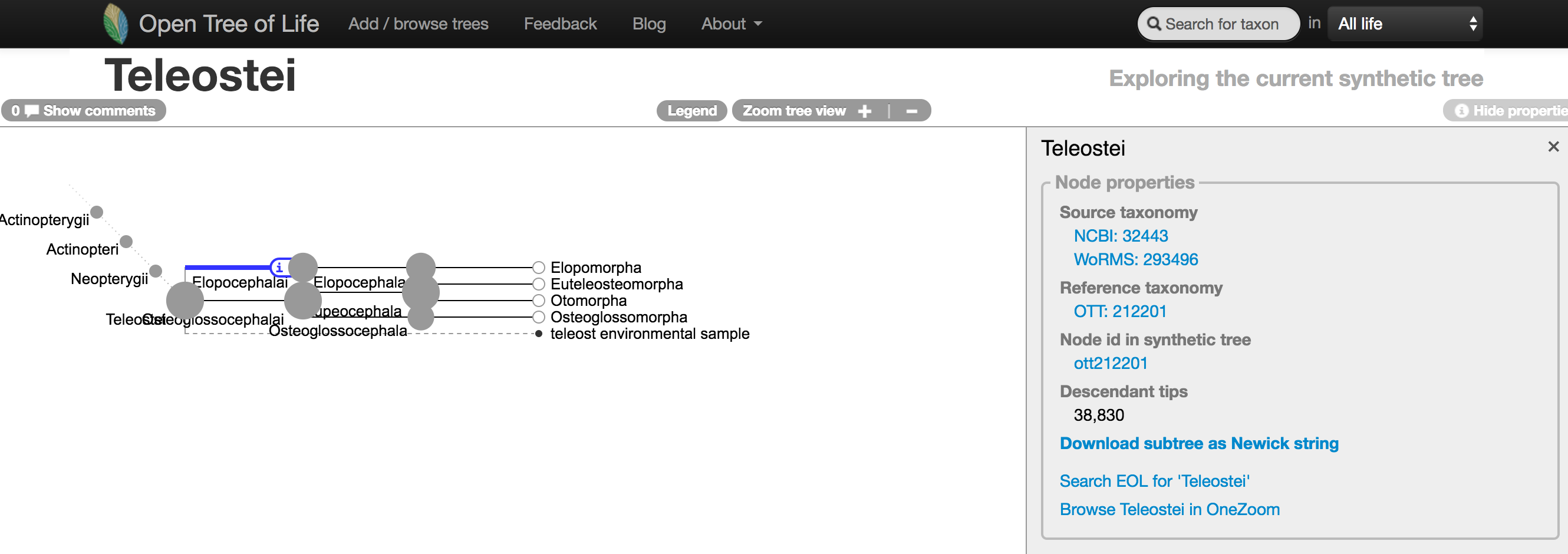
First go to Open Tree of Life using the link below

[https://tree.opentreeoflife.org/](https://tree.opentreeoflife.org/opentree/argus/opentree9.1@ott93302)

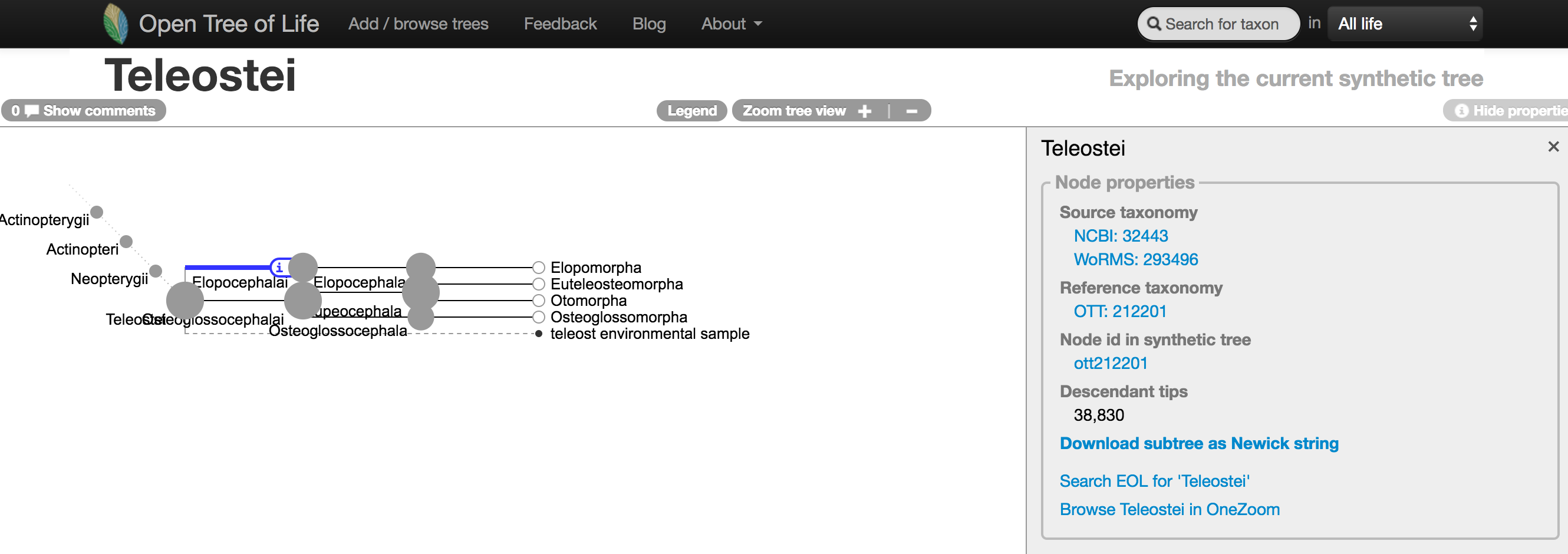
The Open Tree interface is given below. For our example, we need a tree for *Teleostei.* Search for Teleostei in the search bar in the top right corner. It will show the suggested taxa, so you can select the correct one.



After selecting *Teleostei,* the tree browser will zoom in on that particular group as below.



You can click on different taxa names to expand on one particular taxa group. For now, we need to download a tree file for *Teleostei*. To do that click on Download subtree as Newick string link on the right side of the interface as shown below.



This will download the phylogeny for *Teleostei* as a Newick string. However, the downloaded file will have ‘.tre’ extension. In this example, it is named as: ‘subtree-ottol-212201-Teleostei.tre’. Again, rename the file to something meaningful like ‘teleostei\_input\_phylogeny.tre’.

When you open this file in a text editor, it will look like this.



Notice that each taxa name has an Open Tree id attached to the end of the name (ott\_49495502 etc.). These Open Tree ids will be removed by the pipeline to integrate the tree with the character matrix.

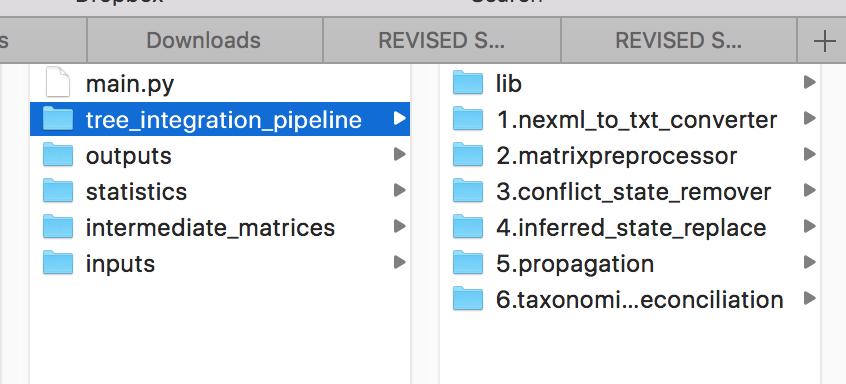
Now, you have completed obtaining the second input file. You are ready to run the pipeline.

**Explain how to obtain meta data here.**

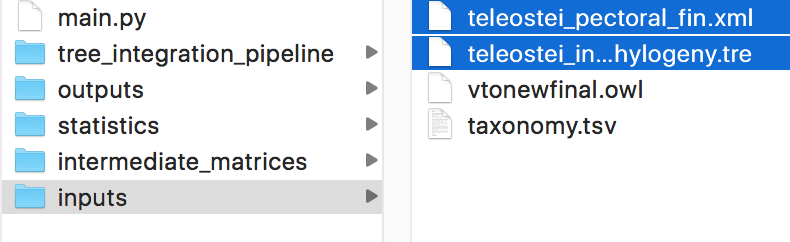
Implementing the pipeline

Explain here how to download the pipeline with github

The general folders and files for the pipeline is given below.



First, move the two input files (matrix and the tree file) downloaded during previous steps to the inputs folder.



Notice that this folder already contains two data files (vtonewfinal.owl and taxonomy.tsv) that are required for the pipeline.

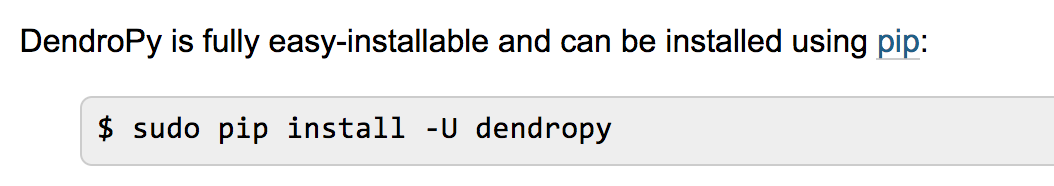
Now you are ready to implement the pipeline, but first, we need to look at the system requirements for the pipeline implementation

System requirements

* Python version 2: recommended version is Python 2.7 or newer. (will not work in Python 3)
  + Install Python if it is not already installed in your system. Mac OS users have a default python installation
* Networkx python library
  + Networkx is an external python library used to analyze network data (<https://networkx.github.io/>)
  + You can install Networkx using ‘pip install’ command in command line as shown below



* + For more details on the installation, please refer to: <https://networkx.github.io/documentation/development/install.html>
* Dendropy python library
  + Dendropy is the most popular library in python for phylogenetic computations
  + You can install it using the pip command as shown below

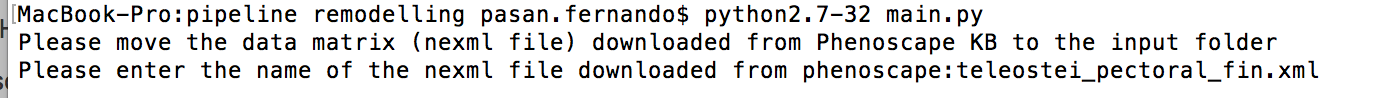


* + For more details, visit the following link: <https://www.dendropy.org/>

Executing the pipeline

If all the system requirements are met, you can run the pipeline by executing the main.py script in command line.

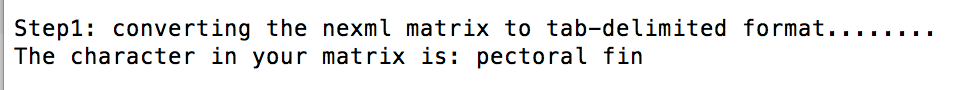
First, open the command line and move to the folder where main.py is located and execute the script with the correct python version (you may have several versions of python installed; be sure to select the correct version in which you installed the required external libraries. In my example, the version is python 2.7)



After the execution of the main program, it will prompt the user to move the character matrix downloaded from Phenoscape KB to the inputs folder (which you already did). Then enter the correct name of the matrix with the extension as shown above.

Now, the pipeline will start; It contains six steps, which we will discuss one step at a time

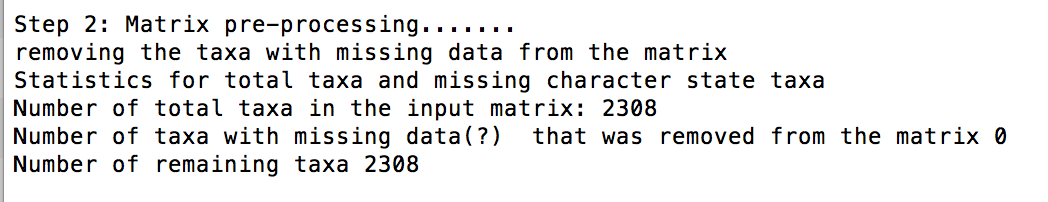
Step 1: Matrix conversion



As shown above, step 1 will convert the input character matrix into a tab-delimited format. Tab-delimited format is easy to process during the remaining steps in the pipeline.

Step 2: Matrix pre-processing

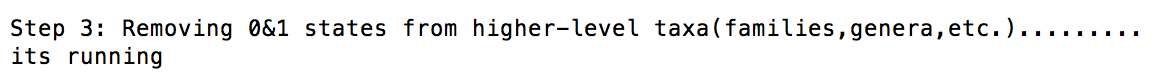
This step will pre-process the tab-delimited matrix. Specifically, it will remove the taxa with missing characters from the matrix. These taxa have a ‘?’ assigned to them and can be included in the matrices downloaded from Onto Trace. They are of no value and must be removed before proceeding to remaining steps.



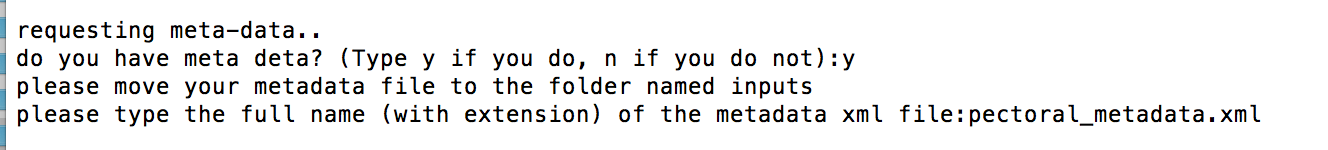
In this instance, there were no taxa with missing data, hence, all 2308 taxa were kept in the matrix.

Step 3: Removal of apparent polymorphisms and conflicts from higher-level taxa

The character matrix can contain cells that has both presence and absence (0&1). When this occurs in taxa above the species level, it can be due to conflicts (Conflicting statements by two authors/papers for the same taxa and the same character) or apparent polymorphisms (Jackson paper). Apparent polymorphisms occur when authors state that a certain family or genus has some species with character presence and some with character absence, but do not provide the species names. In such cases, the family or genus will be represented by ‘0&1’ for the character. These ‘0&1’ states for higher-level taxa must be removed before proceeding to the remaining steps, because we are not entirely sure which characters lack or possess the character.



However, if the user is interested in investigating into such conflicts or apparent polymorphisms, this step will generate a file (conflict\_counts.txt) with a list of all taxa with ‘0&1’ states and their literature sources. This file will only be generated if you have the meta-data file in the inputs folder.



As shown above, user will be prompted for the meta-data file, if the user provides the name of the meta-data file, this step will generate the conflict\_counts.txt file in statistics folder, and proceed into step 4.

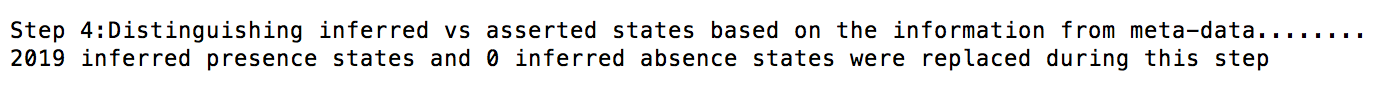
If the user does not have the meta-data file the conflict\_counts.txt file will not be generated and the step 4 is entirely skipped. However, the pipeline will still proceed without the step 4.

Put the image here

Step 4: Distinguishing inferred vs asserted states

Because phenotypic data in KB is ontology annotated, some character states can be inferred based on ontological relationships. For instance, if an author declares that part of the pectoral fin is present, then based on ontological relationships, the presence of the pectoral fin can be inferred. Please refer to the Jackson paper for more details.

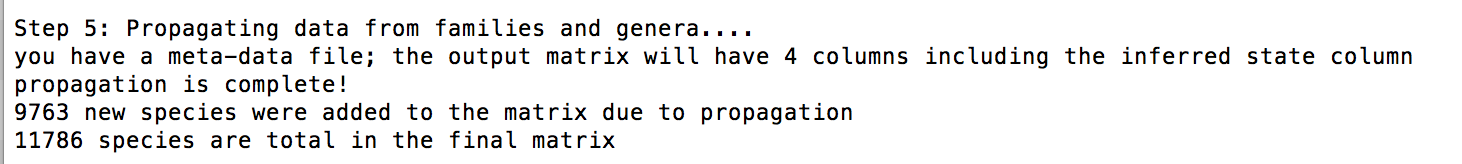
It is important to distinguish between inferred and directly asserted states. This is performed in step 4 using the metadata file. The inferred presence states will be replaced by ‘2’ and inferred absence states will be replaced by ‘3’. This distinguishing is important for visualizing inferred states at a later stage (refer to visualization section)



The lists of taxa with inferred vs asserted presence and absence are saved in the file named ‘inferredstats.txt’ in outputs folder.

Step 5: Propagation

Propagation is an important step in the pipeline. The matrices downloaded from the KB contain some annotations to higher-level taxa (families, genera, etc.). Because the final mapping to the tree is done at species level, propagation of higher-level data to corresponding species is essential to avoid losing higher-level data.



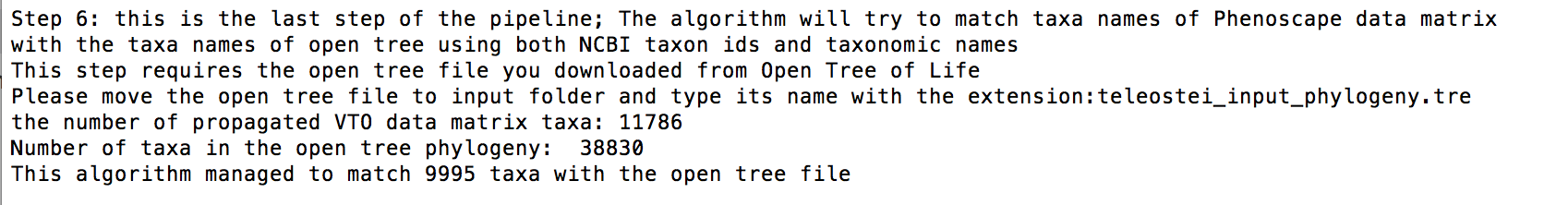
As shown above 9,763 new taxa are added to the matrix by propagation. Now the propagated matrix contains total of 11,786 taxa.

Statistics about the propagation are saved in statistics folder as ‘propagationstatistics.txt’.

Step 6: Taxonomic reconciliation

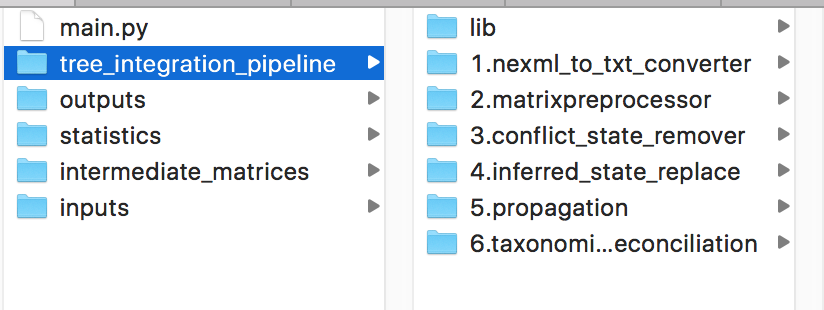
This is the final step of the pipeline. The character matrix from Phenoscape KB is based on Vertebrate Taxonomy Ontology (VTO) taxa names, and we are trying to merge it with an Open Tree phylogeny in which the taxa names are based on NCBI taxon naming system. There can be potential discrepancies between the two naming systems. Taxonomic reconciliation attempts to obtain an efficient taxon matching by first using NCBI taxonomic ids and then using taxa names.

For this step, you need to input the name of the Open Tree phylogeny you moved into the inputs folder



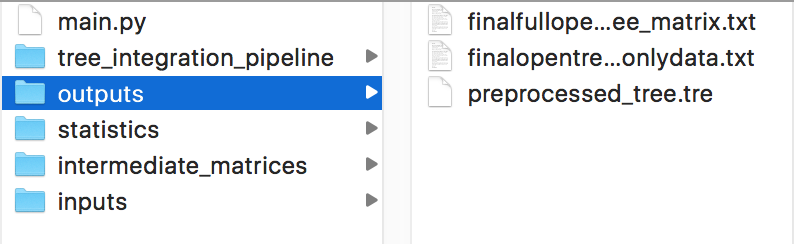
In our example, this step managed to match 9,995 taxa of 11,786 in the propagated matrix. Taxa mismatches are largely due to improper naming conventions, extinct taxa, taxonomic reshuffling in VTO. The details of the mismatched taxa are saved in the statistics folder as ‘finalmismatchedlist\_andstats.txt’

If you are further interested in each individual step of the pipeline, you can find the scripts for each step in tree\_integration\_pipeline folder.



Outputs

The pipeline will generate 3 output files in the outputs folder. These files can be used for ancestor state reconstruction.

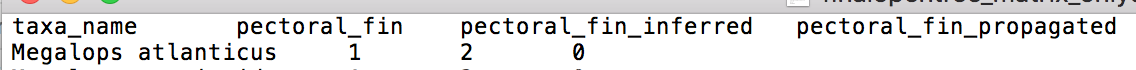


1. ‘preprocessed\_tree.tre’: this is the pre-processed version of your input phylogeny from Open Tree of Life. This version must be used for the ancestral state reconstruction, because it does not have ott ids at the end of each species name. If you use the input version the ancestral state reconstruction will not run efficiently.
2. finalfullopentree\_matrix.txt: This matrix is ready to be merged with the Open Tree phylogeny. This matrix contains all the taxa in Open Tree file, which means there are taxa that are not in the propagated matrix based on VTO taxonomy. For instance, our example had 11,786 taxa after the propagation. However, the Open Tree phylogeny had 38,830 total taxa. This matrix also contains 38,830 taxa but only 11,786 taxa have data for pectoral fin. This matrix can be used for further analysis to investigate phylogenetic clades with missing data.
3. finalopentree\_matrix\_onlydata.txt: This matrix is also ready to be merged with the Open Tree phylogeny. The difference between the previous matrix is: this matrix only contains taxa that are in the propagated matrix. In our example, this means it has 11,786 taxa, not 38,830. This matrix does not have taxa with missing data as before.

The two data files mentioned above (2 and 3) contain four columns

* taxa\_name: contains taxa names
* character\_name: presence or absence of the character
  + 0: absence
  + 1: presence
  + 0&1: both presence and absence due to conflicts or polymorphisms
* character\_name\_inferred: the character states of this column are distinguished for inferred vs asserted as given below
  + 0: asserted presence
  + 1: asserted presence
  + 2: inferred presence
  + 3: inferred absence
* character\_name\_propagated: represents the propagated status of each taxon
  + 0: not propagated
  + 1: propagated

Consider this example below



It can be read as Pectoral fin is present (1 for pectoral fin) for *Megalops atlanticus*. The presence is inferred (2 for pectoral\_fin\_inferred). It is not propagated from a higher- level taxa (0 for pectoral\_fin\_propagated), hence, the state is coming from the original NeXML matrix downloaded from KB.

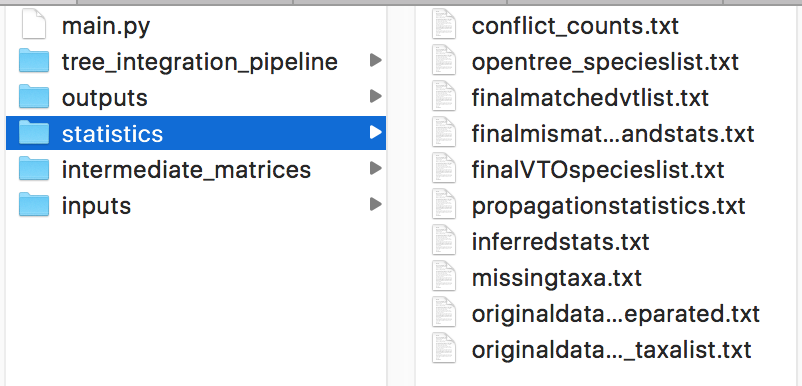


The above example can be read as pectoral fin is present for *Myroconger seychellensis*. The presence is asserted presence and it is propagated from a higher-level taxa.

Ancestral reconstruction piece goes here

Statistics folder

This folder contains files that has valuable statistics regarding the procedure of the pipeline



originaldatamatrix\_taxalist.txt: contains a list of taxa in the original input data matrix

originaldatamatrix\_taxalist\_separated.txt: The same original taxa list separated to different taxonomic levels (families, genera, etc.)

missingtaxa.txt: List of taxa with missing data that were removed during step 2 of the pipeline

conflict\_counts.txt: This file contains a list of all taxa that has ‘0&1’, separated into different taxonomic levels with their literature sources.

inferredstats.txt: lists of taxa separated based on asserted vs inferred presence and absence. This file will only be generated by step 4 if you input the meta-data file

propagationstatistics.txt: statistics about the propagation process conducted during step 5 of the pipeline.

finalVTOspecieslist.txt: list of all the species in the propagated data matrix

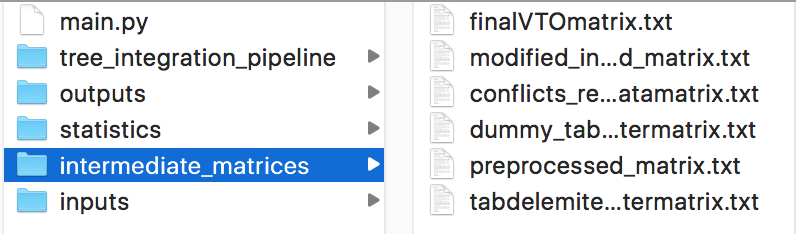
opentree\_specieslist.txt: list of all the species in the open tree phylogeny

finalmatchedvtlist.txt: list of species in the propagated matrix that matched with the open tree species during step 6 of the pipeline.

finalmismatchedlist\_andstats.txt: list of species in the propagated matrix that did not match with the open tree species. This list is separated based on the reason for being mismatched (improper naming syntax, taxa being extinct, etc.)

intermediate\_matrices folder

Each step of the pipeline generates an intermediate version of the data matrix downloaded from KB. Only the final output matrices are included in the outputs folder. All the intermediate matrices are included in this intermediate\_matrices folder.



tabdelemited\_charactermatrix.txt: this matrix is resulted during the step 1 of the pipeline after converting the input data matrix downloaded by KB to tab-delimited format. This matrix should be tab-delimited and contains two columns: taxa\_name and the character you are interested in. The meaning of the character states for the character column is given below.

0: absence

1: presence

0&1: presence and absence: conflicts or polymorphisms

preprocessed\_matrix.txt: This matrix is generated after the step 2 of the pipeline. The taxa with missing data is removed in this matrix.

conflicts\_removed\_datamatrix.txt: this matrix is generated in step 3 after removal of ‘0&1’ states from higher-level taxa.

modified\_inferredadded\_matrix.txt: this matrix is only generated in step 4 if you input the meta-data file. The matrix contains an additional column (character\_name\_inferred) for inferred character states. The meaning of the different states in this column is given below

0: asserted presence

1: asserted presence

2: inferred presence

3: inferred absence

finalVTOmatrix.txt: this matrix is generated after propagation in step 5. The matrix contains a new column (char\_name\_propagated) to represent the propagated status for each taxon. The propagated state can be

0: not propagated

1: propagated

**Important: after running the pipeline for a desired matrix, please remove the contents of the outputs, intermediate\_matrices, and statistics folders to another location or make a backup. If you re-implement the pipeline for a new matrix (another character), these output files will be replaced. Make sure to keep above three folders intact; do not delete them move only their contents.**