CDAO-Store: Ontology-driven Data Integration for Phylogenetic Analysis

Brandon Chisham*, Ben Wright *, Trung Le, Tran Son and Enrico Pontelli

Department of Computer Science, New Mexico State University, Las Cruces, New Mexico, USA

Email: Brandon Chisham*- bchisham@cs.nmsu.edu; Ben Wright *- bwright@cs.nmsu.edu; Trung Le - tle@cs.nmsu.edu; Tran Son - tson@cs.nmsu.edu; Enrico Pontelli -epontell@cs.nmsu.edu;

*Corresponding author

Abstract

Background: The Comparative Data Analysis Ontology (CDAO) is an ontology developed, as part of the EvoInfo and EvoIO groups supported by NESCent, to provide semantics to the descriptions of data and transformations commonly found in the domain of phylogenetic inference. The core concepts of the ontology enables the description of phylogenetic trees and associated character data matrices.

Results: Using CDAO as the semantic backend, we developed a triple-store, named *CDAO-Store*. CDAO-Store is a RDF-based store of phylogenetic data, including a complete import of TreeBASE. CDAO-Store provides a web-based front-end to perform both user-defined as well as domain-specific queries; domain-specific queries include search for nearest common ancestors, minimum spanning clades, filter multiple trees in the store by size, author, taxa, tree identifier, algorithm or method. In addition, CDAO-Store provides a visualization front-end, called *CDAO-Explorer*, which can be used to view both character data matrices and trees extracted from the CDAO-Store. CDAO-Store provides import capabilities, enabling the addition of new data to the triple-store; files in PHYLIP, MEGA, and NEXUS formats can be imported and their CDAO representation added to the triple-store.

Conclusions: CDAO-Store is made up of a versatile and integrated set of tools to support phylogenetic analysis. To the best of our knowledge, CDAO-Store is the first semantically-aware repository of phylogenetic data with domain-specific querying capabilities. The portal to CDAO-Store is available at http://.....

Background

The *CDAO-Store* is a novel portal aimed at facilitating the storage and retrieval of phylogenetic data. The novelty of CDAO-Store lies in the use of a *semantic-based* approach to the storage and querying of data, building on established ontologies for the semantic annotation of data. This approach enables us to overcome restrictions imposed by the use of specific data formats (facilitating interoperation among phylogenetic analysis applications) and makes it possible to formulate more meaningful domain-specific queries.

Motivations

Phylogenetic trees have gained a central role in modern biology. Trees provide a systematic structure to organize evolutionary knowledge about diversity of life. Trees have become fundamental tools for building new knowledge, thanks to their explanatory and comparative-based predictive capabilities. Evolutionary relationships provide clues about processes underlying biodiversity and enable predictive inferences about future changes in biodiversity (e.g., in response to climate or anthropogenic changes). Phylogenies are used with increase frequency in several fields, e.g., comparative genomics [**?**], metagenomics [**?**], and community ecology [**?**].

- Phylogenetic Repositories Repositories provide a well-known certeralized location for sharing results with the research community. As mentioned in the TreeBASE's overview statement this promotes the *reuse reassessment, and recombination* [**?**] of existing results.
- Data Interoperation Data Reuse however is not practically possible without data interoperation. Data tied to a particular tool, or worse, a particular version of a particular tool provides limited value to users of a repository. Ideally repositories should supply their clients with results in a maximally compatible format that does not limit the client to the use of particular software. This issue of particular interest to the Evolutionary Biology community. Several compeating formats exist for representing phylogenies and morphological character data. Additionally there are no commonly accepted methods for applying annotations to branches in a phylogeny, or describing evolutionary models. Also other meta-data such as provenance is not handled.
- semantics and ontologies

  Given the challenges posed by relying on particular file formats, the CDAO-store is built on an ontology for Character State and Phylogenetic data, CDAO, so that data may be supplied in any particular format because the repository operates on semantics rather than relying on any particular file syntax because while data formats capture the syntax of data (e.g., for data transmission), explicit semantics is necessary (e.g., [**?**]) for interpretation, repurposing and application of phylogenetic data. In recent years, knowledge representation in the biomedical domains has predominantly built on the use of domain specific ontologies [**?**, **?**].
- domain-specific querying

  Domain specific querying is also an important feature for a phylogenetic repository. [**?**] This level of query support helps investigators easily pose questions to the resource that might be difficult or impossible to be expressed in a general purpose query language. While a certain amount, of query complexity can be hidden behind the resource's user-interface.

CDAO

The *Comparative Data Analysis Ontology (CDAO)*[1] [**?**] provides a formal ontology for describing phylogenies and their associated character state matrices. It was developed as part of the *Evolutionary Informatics (EvoInfo)*[2] working group, sponsored by the National Evolutionary Synthesis Center.[3]

The CDAO ontology provides the semantic component of a data representation and interoperation stack for phyloinformatics, known as the *EvoIO stack* [**?**]—along with a data exchange format, called `Nexml` [**?**], and a phyloinformatics web servces API, known as PhyloWS [**?**]. CDAO forms the base of this stack defining the semantics for the data represented as `Nexml` files, or otherwise supplied by services implementing this set of standards. Figure 1 illustrates the EvolIO stack.

CDAO is defined in terms of an OWL-DL ontology. It provides a general framework for talking about the relationships between taxa, characters, states, their matrices, and associated phylogenies. The ontology is organized around four central concepts (see also Figure 2): OTUs, characters, character states, phylogenetic trees, and transitions. The key concepts and their mutual relationships within CDAO are illustrated in Figure 3. A phylogenetic analysis starts with the identification of a collection of *operational taxonomic units (OTUs)*, representing the entities being described (e.g., species, genes).

[1] http://www.evolutionaryontology.org
[2] https://www.nescent.org/wg_evoinfo/Main_Page
[3] http://www.nescent.org/index.php

Each OTU is described, in the analysis, by a collection of properties, typically referred to as *characters*. In phylogenetic analysis, it is common to collect the characters and associated character states in a matrix, the character state matrix, where the rows correspond to the different OTUs and the columns correspond to the different characters.

In evolutionary biology, phylogenetic "trees" and "networks" are used to represent paths of descent-with-modification, capturing the evolutionary process underlying the considered OTUs. Since evolution moves forward in time, the branches (edges) on a tree are directed. The terminal nodes typically are anchored in the present time because they represent observations or measurements made on currently existing organisms. The internal nodes represent common ancestors, with the deepest node as the "root" node of the tree. The restriction that each node has at most one immediate ancestor reflects the assumption that evolutionary lineages, once separate, do not fuse; this assumption follows from the *biological species concept* based on reproductive isolation. Branching is seen as a binary process of splitting by speciation or, in the case of molecular sequences, by gene duplication. Even with terminal nodes anchored in the present, it may be impossible to infer the direction of each internal branch, in which case the tree may be referred to as an "unrooted tree," or as a "network." Even the restriction of single parentage may be abandoned, for strictly biological reasons, in the case of lateral transfer or reticulate evolution.

As a general framework, CDAO supplies general classes and relations between those classes, that can be further specialized to meet the needs of a specific application—*Beak length* might be defined as a specialization of CDAO's *Standard* character type.

`nexml`

`nexml` [**?**] is a file format for exchanging data containing character state data matrices and phylogenies. It is syntax is defined in terms of an XML schema, and the semantics of its elements are defined in terms of CDAO classes. Being defined in this way allows direct translation to CDAO class instances. This guarantee is also important to using it as a medium of exchange since its semantics can be agreed upon by both the provider and recipient of a dataset.

[ENRICO: The description of NeXML is too vague. Characterize the main elements in the format]

PhlyoWS

PhyloWS (Phyloinformatics Web Services API) is a standard for exposing phylogenetic data as a Web service, in such a way that particular data items, can be referenced by persistent HTTP URI's. PhlyoWS URI's begin with `/phylows/` as the standard delimeter, then based on the phylogenetic information being queried, e.g., accession number or what type of data, such as taxon, tree, or study. [ENRICO: the previous sentence is suspended] A specification for PhyloWS can be found at www.nescent.org/wg_evoinfo/PhyloWS/REST.

[ENRICO: people may not be familiar with web services and similar things; please define what a web service API is, what is it for. Provide one example in PhyloWS; include citations]

Implementation

CDAO-store builds on the EvoIO technology stack to provide a framework for supplying semantic services for phylogenetic data services. The platform is open-source and is available on source-forge, at http://sourceforge.net/projects/cdaotools/. It's divided into three main parts. A data importer/file translator, a database and web interface, and a gui visualization tool.

The file importer/translator is implemented in C++ and Python. In addition to its own set of parsers, the translator uses the NCL[4] library to read certain file formats. After reading, it maps data from these files on to an object model that mirrors CDAO classes, and then either converts to some specified format or to an RDF/XML serialization of the data. The import portion of this part of the system is written in Python and uses the RDFlib[5] module to store the RDF serializations produced by the translator into a database making it available to query on the web or by using the visual tools.

The web and database portion of the application stores, and provides access to the data for the visual tools. This portion of the application is primarily implemented as a set of scripts in a variety

[4] http://sourceforge.net/projects/ncl/
[5] http://www.rdflib.net/

of languages. The web interface is divided into two principal parts an HTML user interface, and a PhyloWS data provider. The HTML interface allows for online querying/exploration of datasets, while the PhyloWS interface supplies access to datasets for our visual tools or other third party programs. The database portion of the application is implemented as an RDFlib store running on a MySQL database.

The visual tools are implemented as a Java JNLP application called CDAO-Explorer. It uses a variety of frameworks to support its operation including Pellet[6] and Prefuse[7].

CDAO-Explorer provides a tree and matrix search windows which allow one to search for and load particular datasets, and visualizers for those data sets. It also allows one to make annotations about a dataset, or a general project space, a set of data sets of interest. These annotations can be from CDAO, Dublin-Core, or from a user-supplied source of annotation types.

Results

Web-Tools

The web tools provide a variety of querying and data access features for both human and programmatic access to data. It allows one to retrieve data sets by author name, tree identifier, taxon, algorithm, or method. It also supports computing the minimum spanning clade or the nearest common ancestor of a set of taxa. It also allows one to list trees conforming to certain measures. For example, finding all tress larger or smaller than a given size.

Our PhyloWS implementation is the basis for all the data access features of CDAO-Store. The other web components, and the CDAO-Explorer tool use it to access data. URI's are divided into three conceptual parts. The address of the store site, and path prefix http://www.cs.nmsu.edu/~cdaostore/cgi-bin/phylows, a query type (i.e. tree, matrix, msc, nca, listing, or size), and parameter list. The specific parameters depend on the query type. For example the msc and nca query types expect a list of taxon id's separated by '/' The listing query takes optional limit and offset parameters to paginate results. The size query takes a direction (greater, less, or equal), a criteria (node, internal, or leaf) and a size (some numeral).

[6]http://pellet.owldl.com/

[7]http://prefuse.org/

[8]http://graphml.graphdrawing.org/

CDAO-Explorer

CDAO-Explorer has achieved a basic level of functionality. It provides search and visualization for both trees and matrices and a set of additional features not currently available in a single tool.

Annotation is an important part of CDAO-Explorer. It allows users to attach arbitrary annotations to data items, as well as collections of resources. CDAO-Explorer also allows users to load or save custom data not in the CDAO-triplestore. It also allows users to export pictures of particular visualizations.

Tree Viewer

Tree Viewer is the graphical application used to display trees. It is built on top of the Prefuse visualization framework. Data from the CDAO-triplestore is converted into the Graphml format [8] and then run through the visualization application.

Tree Viewer has a few key features. The first is that there is two different layouts for the tree to be displayed. By default, it uses a Force Layout which allows the nodes of the tree to 'bounce' around as if pulled by strings till it reaches an equilibrium. The second is called a Node Layout which resembles a more standard parent/child structured tree going from left to right.

Another ability that TreeViewer has is to search across the tree using the node and edge label names, highlighting all that currently apply. For instance, a tree may have many nodes that have as part of its name '#Ilex_'. When this search is performed, all nodes with the label containing that will be highlighted. Labels for nodes are generally the taxa name for that TU or if it is an unknown internal node will have the convention of being named '#nodeX' where X is some number. Edge labels are similar in that they are the labels of the two nodes combined as 'source_destination'.

It is also possible to view more specific details on a specific node or edge. Currently, the only detailed information available is the label.

Lastly, there is the option to save the tree visualization as a jpg or png file.

Matrix Viewer

We have developed a custom framework for visualizing matrices. It assigns color codes to character states allowing one to graphically appraise large matrices to quickly see patterns in the source data. It allows users to scale matrices, select regions of a matrix to see in greater detail, and attach annotations to particular cells of a matrix.

Related Work

Nexplorer

Nexplorer is an application that also allows for the browsing of phylogenetic trees and character matrices. However, it only allows for NEXUS formatted files and only displays the trees and matrices in one layout. Nexplorer does have the ability to look at internal nodes in the trees, however it does not have the ability to look at edges.

PhyloWidget

PhyloWidget is another phylogenetic tree viewer application. This one is much more interactive and customizable than Nexplorer, however it only displays trees defined in the Newick or Nexus file format. Like Nexplorer, PhyloWidget also does not do anything with tree edges.

Discussion

With this basic level of functionality in place, we envision extending CDAO tools to include support for describing workflows in cooperation with the MIAPA[9] effort. We also plan to add additional query features to the web-interface including the ability to process user supplied SPARQL Queries.

Conclusions

Current State

The CDAO-store tool set provides a robust foundation for a semantically aware, phylogeny resource. The query and translation services are well developed and based on an easily extensible framework to easily address additional development of features. The CDAO-Explorer portion of the store has achieved a good base-line functionality and provides a set of useful features to advance the current state of visualization of large data sets in this field. Also it provides a good proof-of-concept for integrating semantic information and other meta-data in a seamless and natural way.

Future Directions

Several exciting features are envisioned to extend the existing tool set. For the web we plan to allow users to submit and execute their own *SPARQL* queries to our data-store so they can accomplish queries not supported by the interface. Also we hope to add additional file-types to the translation tool. CDAO-Explorer will include tighter integration between the tree and matrix visualizations, and also phase in support for describing processes and workflows, as part of it's existing support for annotating sets of tree and matrix files.

Availability and Requirements

Project name: CDAO Tools
Project home page: http://www.cs.nmsu.edu/~cdaostore/
Operating system(s): Linux, Mac, Unix
Programming language: Bash, C++, Java, Perl, PHP, Python
Other requirements:
License: GPL
Any restriction to use by non-academics:

Authors contributions

Brandon focused on development of the web and database tools, and the integration of the tree and matrix and tree visualizers into the CDAO-Explorer application.
Trung developed the mega format reader for the translator tool,as well as the matrix visualization tool.
Enrico guided the development of the project.
Son guided the development of the project.
Ben developed the tree viewer portion of the CDAO-Explorer tool, as well as updating the translator tool to accommodate the latest changes to the CDAO standard.

[9]Minimum Information About a Phylogenetic Analysis

Acknowledgements

This project is currently funded by NSF CREST grant HRD-0420407 and NSF IGERT grant DGE-0504304

References

Figures

Figure 1 - The EvoIO Stack

This is the structure of the EvoIO stack developed by he EvoInfo working group of the National Evolutionary Synthesis Center.

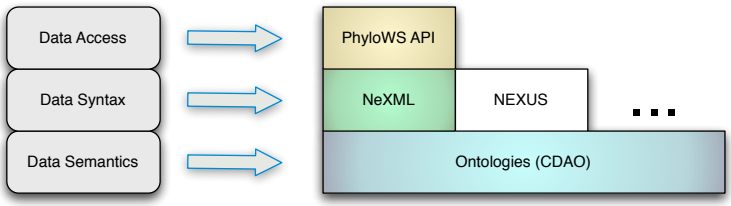

Figure 1: The EvoIO Stack

Figure 2 - The Principle View of OTUs and Characters

This figure summarizes the core concepts from phylogenetic analysis that are captured by the CDAO ontology.

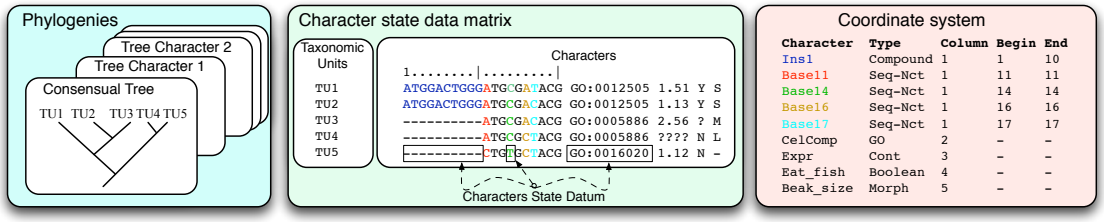

Figure 2: OTUs and Characters

Figure 3 - Snapshot of the Key concepts of CDAO

This figure provides a very small summary of the core concepts and relations described in CDAO.

Figure 2 - TreeViewer with search

This is the TreeViewer Application displaying the tree Tree3099 from Treebase and searching for all nodes and edges with #Ilex_

Additional Files

Additional file 1 — TreeViewerSearch.jpg

This is Figure 1

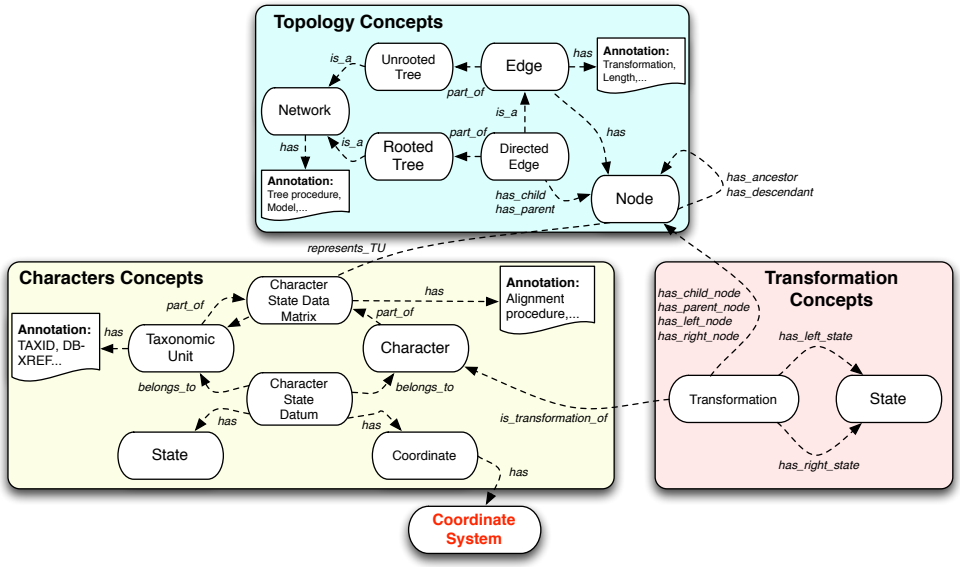

Figure 3: Core Concepts in CDAO