CDAO Store: A New Vision for Data Integration

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

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

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

*Corresponding author

Abstract

Background: The Comparative Data Analysis Ontology(CDAO) [1] is an ontology developed, as part of the EvoInfo[2] and EvoIO[3] groups supported by NESCent[4], 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 a backend triple-store, we developed CDAO-Store. CDAO-Store is a web-based application that can be used to query phylogenetic data for nearest common ancestor or minimum spanning clade as well as filter multiple trees in the store by size, author, taxan, tree identifier, algorithm or method. In addition, CDAO-Store also has a visualization application called CDAO-Explorer which can be used to view both the character matrix or tree of anything in CDAO-Store. In order to keep CDAO-Store up to date, PHYLIP, MEGA, and NEXUS files can be translated to CDAO format and then uploaded to CDAO-Store.

Conclusions: CDAO store is made up of a versatile and compelling set of tools. It provides an good example of what can be done with an extensible, semantically aware data resource.

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

Background

CDAO

CDAO, Comparative Data Analysis Ontology, provides a framework for describing phylogenies and their associated character state matrices. It was developed as part of the Evolutionary Informatics working group along with the NeXML file format, and the PhyloWS Webservice standard, forming what the group called the EvoIO stack. 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.

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. As a general framework it supplies general classes and relations between those classes, it is intended that for practical work these will be extended to for example talk about more specific types of characters or states. (e.g. Beak length might be defined as a specialization of CDAO's *Standard* character type).

NeXML

NeXML[5] 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 it's semantics can be agreed upon by both the provider and recipient of a dataset.

PhlyoWS

PhyloWS is a standard for exposing phylogenetic data as a webservice, in such a way that particular data items, can be referenced by persistent HTTP URI's.

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

[5]http://www.nexml.org

translator uses the NCL[6] 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[7] 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 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[8] and Prefuse[9].

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

[6] http://sourceforge.net/projects/ncl/
[7] http://www.rdflib.net/
[8] http://pellet.owldl.com/
[9] http://prefuse.org/

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).

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 [10] 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.

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

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.

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[11] 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.

[11] Minimum Information About a Phylogenetic Analysis

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.

Acknowledgements

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

References

Figures

Figure 1 - TreeViewer with search

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

Figure 2 - Sample figure title

Figure legend text.

Tables

Table 1 - Sample table title

Here is an example of a *small* table in LaTeX using `\tabular{...}`. This is where the description of the table should go.

My Table		
A1	B2	C3
A2	...	..
A3	..	.

Table 2 - Sample table title

Large tables are attached as separate files but should still be described here.

Additional Files

Additional file 1 — Sample additional file title

Additional file descriptions text (including details of how to view the file, if it is in a non-standard format or the file extension). This might refer to a multi-page table or a figure.

Additional file 2 — Sample additional file title

Additional file descriptions text.

TreeViewerSearch.jpg

This is Figure 1