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 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://www.cs.nmsu.edu/~cdaostore.

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.

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 [1], metagenomics [2], and community ecology [3].

- Phylogenetic Repositories Repositories provide a well-known centralized location for sharing results with the research community. As mentioned in the TreeBASE's overview statement this promotes the *reuse reassessment, and recombination* [4] of existing results. Also when data are collected into a repository, one can query across large sets of data to determine general features about phylogenies or merge phylogenies into a super-tree like the Tree of Life. [5, 6] For example, being able to gather statistics about the structure of published trees, software developers, can write better test-suites for their packages. A variety of specialities from population genetics to historical linguistics will benefit from such a comprehensive resource [5]. A resource's data-model is critical to determining its ability to serve these functions, because the model restricts the kinds of queries one can perform on the resource. Many prior resources use Newick strings to represent trees limiting the ability for users to query based on structures contained in their trees. [5]

- 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 competing 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., [7]) for interpretation, re-purposing and application of phylogenetic data. In recent years, knowledge representation in the biomedical domains has predominantly built on the use of domain specific ontologies [8, 9].

- domain-specific querying

  Domain specific querying is also an important feature for a phylogenetic repository. [5] 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] [7] 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* [10]—along with a data exchange format, called `Nexml` [11], and a phyloinformatics web servces API, known as PhyloWS [12]. CDAO forms the base of this stack defining the semantics for the data represented as `Nexml` files, or otherwise supplied by ser-

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

vices 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). 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` [11] 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.

A basic overview of the nexml structure:

- `<nexml>` This is the root element for `nexml` [13]. Meaning everything is nested inside here.
- `<otus>` This is the similar to the TAXA block in `NEXUS` files. Nested inside are ids and (optionally) labels to all the relevant TUs [13].
- `<characters>` This is very similar to the CHARACTERS block in `NEXUS` files [13]. As such, it sets up the character state matrix. It can do this in a few different formats such as molecular sequences, categorical data, or continuous data. A difference from `NEXUS` though is that more information per character can be specified here. Depending on the format, the matrices can be formed by either `<matrix>` `<row>` or `<states>` `<state>` elements [13].
- `<tree>` This element sets up the tree in a manner very similar to `GraphML` [14]. Where the tree is described as a list of `<node>` and `<edge>` elements [13]. Where all the elements in the tree are listed as individual nodes with the connections of the trees being explicitiy stated as edges between nodes. Edges work in a source to target manner [13].
- `<dict>` This allows one to set up arbritrary key/value pairs that may be useful to the data file, but not necessarily defined elsewhere in the `nexml` format [13].

PhlyoWS

PhyloWS (Phyloinformatics Web Services API) is a standard for exposing phylogenetic data as a web service. Web services are tools that can perform certain tasks via HTTP [15]. PhlyoWS specifically uses a RESTful style web service which uses a few well-known operations to relay data [16] [17]. This

works in a similar way as GET or POST for HTTP [17]. All PhlyoWS URI's begin with `/phylows/` as the standard delimeter. Then based on the phylogenetic information being queried a datastructure will be given, such as taxon, tree, or study. This is followed by any specific identifiers needed for the query. For example, http://purl.org/phylo/treebase/phylows/tree/TB2:Tr3099?format=rdf is a way to access information from TreeBase2 using PhyloWS. When this url is accessed, it returns the tree with the Treebase ID of 'Tr3099' in an rdf format [18]. A specification for PhyloWS can be found at [16].

Implementation

CDAO-store builds on the EvoIO technology stack to provide a semantic-based repository of phylogenetic data, accessible through semantic web services and a domain-specific query language. The CDAO-strore platform is open-source and is available as a SourceForge project, at sourceforge.net/projects/cdaotools.

The implementation of CDAO-store is organized in three interconnected modules, as illustrated in Figure 4: a *data importer module*, a *repository module*, and an *exporter module*.

Data Importer Module

The purpose of the *data importer* module is to import phylogenies and their associated data into the repository, automatically extracting their representations in terms of instances of the CDAO ontology. The *data importer* module can process phylogenetic data encoded in several commonly used data formats. The importing process is used to extract a semantic-based encoding of the input data, as instances of the concepts and properties of CDAO. The current implementation provides sub-modules that can extract CDAO instances from files encoded in NEXUS [19], `nexml` [11], PHYLIP [20], and MEGA [21]. The various parsing sub-modules have been developed either from scratch, using combinations of C++ and XSLT style sheets, or using pre-built libraries, such as the NEXUS Class Library (NCL).[4] The data importer module is also designed to enable the processing of the content of the TreeBase[5] repository—a popular repository of user-submitted phylogenies and associated generating data—importing the corresponding CDAO instances into CDAO-store. After reading each input file, the data importer module maps data from these files to an object model that mirrors CDAO classes, producing RDF/XML triples that can be deposited in the CDAO-store repository (i.e., passed to the repository module). The data importer module is also capable of mapping the object model back into any of the acceptable input data formats; this enables the use of the CDAO-store system for conversion among data formats.

Repository Module

The repository module provides two core functionalities: *storage* and *querying*. The repository module maintains a triple store, used to maintain all the CDAO instances created, either through submitted user files or through processing of TreeBASE content. The triple store is implemented in Python and uses the RDFlib [**?**] module to store the RDF serializations of CDAO instances in a relational database (implemented using a MySQL database). The repository modules supports the execution of queries against the triple store.

This set of queries is primarily drawn from the descriptions given by Nakhleh et al. [5], who provide a characterization of a relevant set of domain specific queries that are desirable for any repository of phylogenetic structures. The repository module supports all the, fully-specified, types of queries identified in [5]. This is a diverse set of queries ranging from those that can be processed using simple syntactic matching requiring little additional reasoning to complex queries of tree structures. To address this variation, the query system is divided into two primary modules.

- A scripting library has been built on top of the RDFlib library allowing the system to process most queries using standard SPARQL queries. This portion of the query engine is primarily used for queries that require simple matching or basic reasoning about types.
- More advanced queries are accomplished using Prolog. It is a popular programming language for knowledge representation and rea-

[4] sourceforge.net/projects/ncl
[5] www.treebase.org

soning, which provides a more natural milieu for expressing some complex relations, especially those that are transitive, or require matching or reasoning about tree structure.

Query	Engine
Minimum Spanning Clade of a Set of Taxa [5]	Prolog
Given a set of Taxa find all the relationships between them in all phylogenies [5]	Not Supported
Given a set of Taxa find all phylogenies containing that set [5]	SPARQL
Given a phylogenetic method find all phylogenies constructed using it [5]	SPARQL
Given an integer find all phylogenies that contain that many taxa [5]	Prolog
Find all phylogenies constructed by an author or tool [5]	SPARQL
Given a characteristic, return phylogenies with that characteristic [5]	Prolog
Given a phylogeny and a measure return all phylogenies some distance from it [5]	Prolog
Given a model of evolution return phylogenies reconstructed using that model [5]	Not Supported
Given some measures return statistics about the database [5] for them	SPARQL

For the queries that are not supported, the primary cause is a lack of full specification of the query, or the data is not currently available. For example, the set of relationships one might be interested in having returned was not fully specified in the original article. Though once some specific characteristics of phylogenies are identified it will be possible to incorporate them into the system with minimal effort. Also the source data did not contain information about which models where used to construct which phylogenies, nor does CDAO currently contain a model description framework, so it is not currently possible to perform this query on the repository.

Exporter Module

The goal of the exporter module is to provide interactions with the user. The module provides three main interaction mechanisms: a *web portal*, a *web service interface*, and a set of *visualization tools*.

The web portal offers a HTML interface to interact with the repository. The interface allows the online submission of queries, the ability to browse the content of the triple store, and forms to submit new data sets to the triple store. The web portal allows also 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 (i.e., another ontology).

The web service interface is an implementation of the PhyloWS protocol; this is realized by a collection of scripts, capable of generating the necessary SPARQL queries to be submitted to the repository module.

The visual interface, called CDAO-Explorer, provides two graphical visualization tools; one tool is used to provide a graphical representation of phylogenetic trees and networks, while the second one provides graphical representations of character data matrices. The tools have been implemented using a combination of Java and the Prefuse visualization toolkit.[6]

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

[6] prefuse.org

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).

[ENRICO: in the previous two paragraphs we need more details; in particular I would like a precise list of the different types of queries we can handle.]

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 using the Prefuse visualization framework. Data from the CDAO triple store (provided by the repository module) is converted into the GraphML format [14] and then supplied to the visualization application. Figure 5 shows a snapshot of the tree visualization.

The Tree Viewer has several key features. The first is that there is two different layouts for the tree to be displayed. By default, it uses the Tree Viewer uses a *force layout*, which allows the nodes of the tree to "bounce" around as if pulled by strings until an equilibrium is reached. The second layout is called *node layout*, which resembles a more standard parent/child structured tree going from left to right.

Another feature provided by the Tree Viewer is the ability 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 the corresponding 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.

Finally, the Tree Viewer provides the option to save the tree visualization as a jpeg 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.

Figure 6 shows a snapshot of the Matrix Viewer.

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[7] 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, Prolog
Other requirements:
License: GPL

[7]Minimum Information About a Phylogenetic Analysis

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

1. Ellegren H: **Comparative Genomics and the Study of Evolution by Natural Selection**. *Molecular Ecology* 2008, **17**(21):4586–4596.
2. Wu M, Eisen J: **A Simple, Fast, and Accurate Method of Phylogenomic Inference**. *Genome Biology* 2008, **9**(10):R151.
3. Webb C, Ackerly D, McPeek M, Donoghue M: **Phylogenies and Communtiy Ecology**. *Annu. Rev. Ecol. Syst.* 2002, **33**.
4. **TreeBASE**. *http://www.treebase.org* 2010.
5. Nakhleh L, Miranker D, Barbancon F: **Requirements of Phylogenetic Databases**. In *Third IEEE Symposium on Bioinformatics and Bioengineering*, IEEE 2003:141–148.
6. **Tree of Life**. *http://www.tolweb.org* 2010.
7. Prosdocimi F, Chisham B, Pontelli E, Thompson J, Stoltzfus A: **Initial Implementation of a Comparative Data Analysis Ontology**. *Evolutionary Bioinformatics* 2009, **5**:47–66.
8. Schulze-Kremer S: **Ontologies for Molecular Biology and Bioinformatics**. *In Silico Biology* 2002, **2**(17).
9. Sklyar N: **Survey of Existing Bio-Ontologies**. Tech. rep., University of Leipzig 2001.

10. Stoltzfus A, Cellinese N, Cranston K, Lapp H, McKay S, Pontelli E, Vos R: **The EvoIO INTEROP Project**. http://www.evoio.org/wiki/Main_Page, National Evolutionary Synthesis Center 2009.
11. Vos R: **`nexml`: Phylogenetic Data in XML**. *http://www.nexml.org* 2008.
12. Lapp H, Vos R: **Phyloinformatics Web Services API: Overview**. https://www.nescent.org/wg/evoinfo/index.php?title=PhyloWS, National Evolutionary Synthesis Center 2009.
13. **Future Data Exchange Standard**. *https://www.nescent.org/wg/evoinfo/index.php?title=Future_Data_Exchange_Standard&diff=5937&oldid=prev#Element_description*.
14. Brandes U, Eiglsperger M, Herman I, Himsolt M, Marshall MS: **GraphML Progress Report - Structural Layer Proposal** 2002.
15. **Web Services Glossary**. *http://www.w3.org/TR/ws-gloss/*.
16. **PhyloWS/REST**. *www.nescent.org/wg_evoinfo/PhyloWS/REST*.
17. Fielding RT, Software D, Taylor RN: **Principled Design of the Modern Web Architecture**. *ACM Transactions on Internet Technology* 2002, **2**:115–150.
18. **SourceForge.net: API - treebase**. *http://sourceforge.net/apps/mediawiki/treebase/index.php?title=API*.
19. Maddison D, Swofford D, Maddison W: **NEXUS: an Extensible File Format for Systematic Information**. *Syst. Biol.* 1997, **46**(4):590–621.
20. Felsenstein J: **PHYLIP: Phylogeny Inference Package**. *Cladistics* 1989, **5**:164–166.
21. Kumar S, Dudley J, Nei M, Tamura K: **MEGA: A Biologist-centric Software for Evolutionary Analysis of DNA and Protein Sequences**. *Briefings in Bioinformatics* 2008, **9**:299–306.

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.

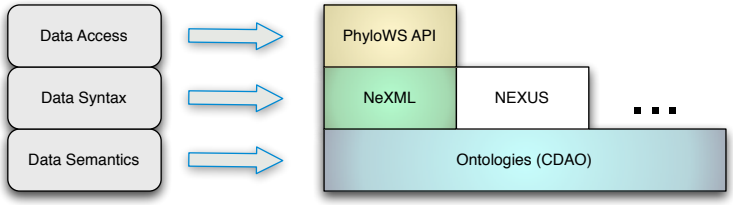

Figure 1: The EvoIO Stack

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.

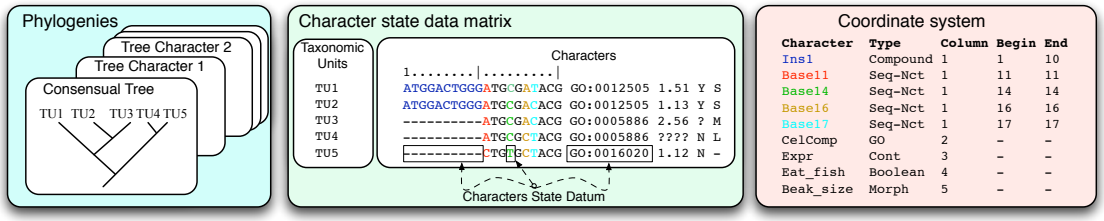

Figure 2: OTUs and Characters

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.

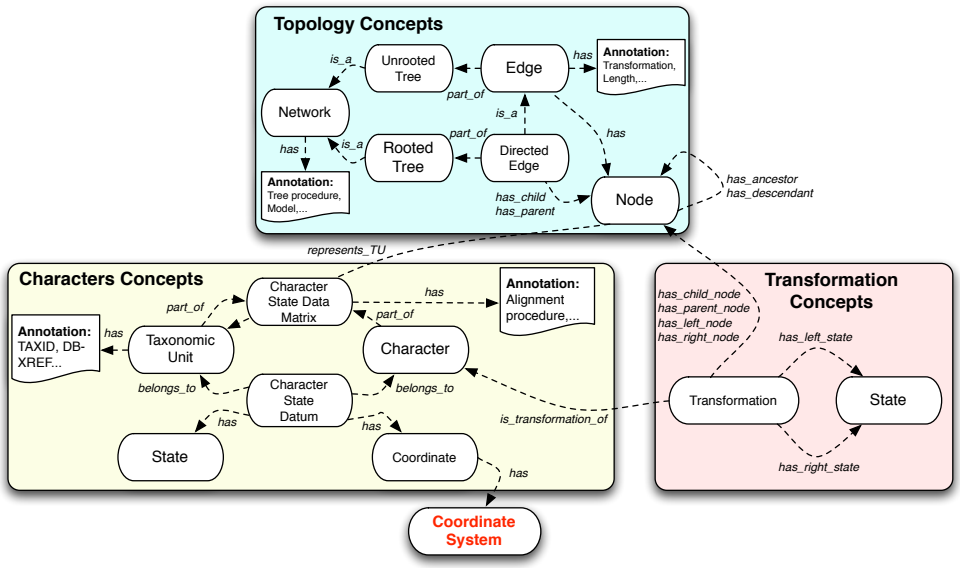

Figure 3: Core Concepts in CDAO

Figure 4 - Structure of CDAO-store
This figure shows the overall structure of the implementation of the CDAO-store.

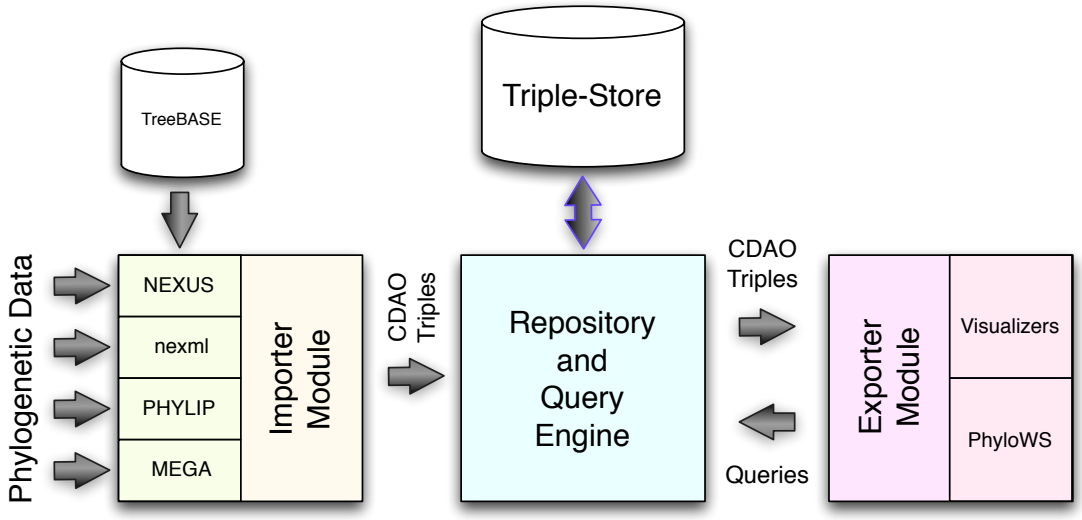

Figure 4: Overall Organization of CDAO-store

Figure 5 - Tree Viewer with search
This is the TreeViewer Application displaying the tree Tree3099 from TreeBase and searching for all nodes and edges with `#Ilex_`.

Figure 6 - Matrix Viewer

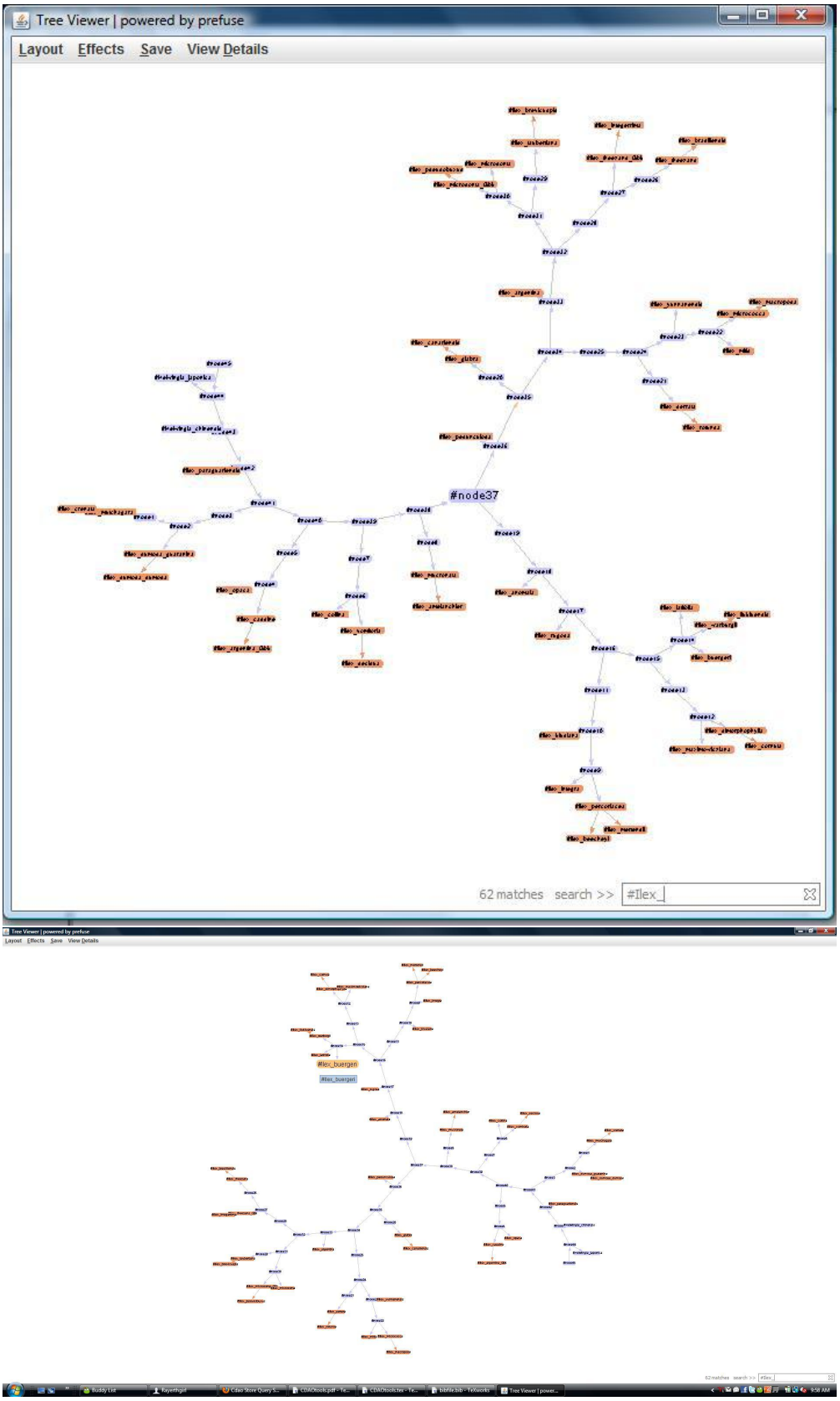

Figure 5: TreeViewer Application with the force layout and the search feature.

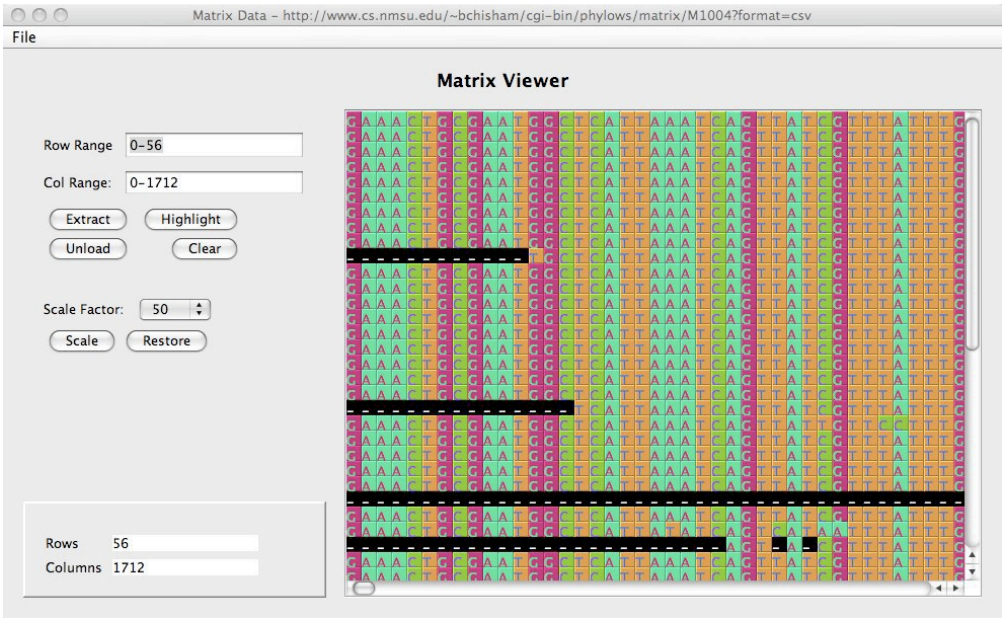

Figure 6: MatrixViewer Application