BIO-AJAX: An Extensible Framework for Biological Data Cleaning

Katherine G. Herbert[*] Narain H. Gehani[†] William H. Piel[‡]

Jason T. L. Wang[§] Cathy H. Wu[¶]

Abstract

As databases become more pervasive through the biological sciences, various data quality issues regarding data legacy, data uniformity and data duplication arise. Due to the nature of this data, each of these problems is non-trivial. For biological data to be corrected and standardized, new methods and frameworks must be developed. This paper proposes one such framework, called BIO-AJAX, which uses principles from data cleaning to improve data quality in biological information systems, specifically in TreeBASE.

1 Introduction

Due to advances in biological research, information in biological data repositories has grown substantially. As with any database, these data repositories now must cope with a number of data quality issues that are inherent to very large databases. These data quality issues include synonymy, polysemy and data redundancy, to name a few. However, due to the complex and diverse nature of the data, the problem of improving the data quality is non-trivial. If the data quality is not maintained or improved, then for data extracted from the repositories by a third party or for data mining purposes, the applications and knowledge based on the inconsistent data will either fail or be skewed.

Data repositories are also expanding their functionalities, requiring more interactions among the data, thus creating more data quality problems. Currently, most data repositories are interrelating with each other, creating a number of integration problems such as reconciling both data and schemas. With biological research changing rapidly, creating more detailed information about biological processes, repositories must address the issue of how to integrate complex submission data with legacy data that follow different data models [10]. Many data repositories endeavor to reduce redundancy within their databases in favor of giving the user highly annotated consensus data. For example, the Protein Information Resource [17] strives to provide users with highly annotated information about the proteins stored within its databases. To provide this, its curators must manage the data through both manual and automated techniques. Data cleaning can offer methods for making the process more automated. Finally, the discoveries in biology affect not only the scientific community, but also other communities such as the business and finance communities. Therefore, there is a need for simple interfaces that give results in a concise, easy to understand style. All of the challenges can be interpreted as data quality problems and addressed through data cleaning and exploratory data mining [3].

The aim of this paper is to propose an extensible toolkit to address data quality issues through data cleaning within biological data repositories. This toolkit, BIO-AJAX, is an extensible framework with various operations that allows a repository to extend concepts within the framework to cater to a repository's needs. The operations can be interpreted as needed by the repository to perform various data quality and data cleaning activities.

As an example, we will show in the paper how BIO-AJAX is applied to solving the nomenclature problem in TreeBASE. TreeBASE is a phylogenetic and evolutionary information system, available at `www.treebase.org` [11] and accessible from GenBank. The system contains citation and experimental data for evolutionary studies. It can display the experimental data through dendrograms on the website. The user can then navigate these dendrograms, finding other dendrograms with similar taxa.

The nomenclature problem in TreeBASE is mainly concerned with the fact that nomenclature for evolutionary units is not entirely standardized. While Linnaean

[*] Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA.

[†] Department of Computer Science, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA.

[‡] Department of Biological Sciences, 608 Cooke Hall, State University of New York at Buffalo, Buffalo, NY 14260, USA.

[§] To whom correspondence should be addressed: College of Computing Sciences, New Jersey Institute of Technology, University Heights, Newark, NJ 07102, USA. Tel: (973) 596-3396, Fax: (973) 596-5777, Email: wangj@njit.edu.

[¶] Department of Biochemistry and Molecular Biology, Georgetown University Medical Center, 3900 Reservoir Road, NW, Box 571455, Washington, DC 20057-1455, USA.

nomenclature (e.g. "Homo sapiens" or "Canis familiaris") is used pervasively within scientific communities, non-scientific communities use general vernacular terms such as "human" or "dog". This creates a number of limitations for evolutionary databases. First, it limits the community of users only to those familiar with the nomenclature. Second, it creates challenges concerning integration with other evolutionary biology resources. Finally, it creates inconsistencies in analyzing the data if there are non-standard interpretations of the nomenclature within the database.

In general, due to the type of data stored in TreeBASE, TreeBASE is unable to control the inconsistency problems within the database concerning the nomenclature. Ideally, all of the nomenclature should conform to a specific set of nomenclature, such as the Linnaean nomenclature, which TreeBASE strongly recommends within its submission guidelines. However, there are cases where the nomenclature needs to be slightly modified so that an experiment is properly modeled. For example, in an evolutionary study among organisms of the same species, each organism needs to be differentiated. Therefore, one common practice is to amend the taxon name, putting a suffix at the end of the taxon name unique to the organism. While this modification has great meaning to the study, it does create inconsistent data concerning the nomenclature and makes it difficult to have complete studies about all occurrences of a species within the database. We propose in the paper a technique, implemented and incorporated in BIO-AJAX, for solving these inconsistency and incompleteness problems concerning the nomenclature in TreeBASE.

The benefits of this research can greatly improve many components of biological databases in general, and TreeBASE in particular. First, it can enhance information retrieval and knowledge discovery results within these databases. Any data mining performed on the database will reflect the content of the database rather than the inconsistencies within the database. Integration with other repositories can be conducted more efficiently. Finally, the groups of users viewing the data can be expanded since the scientific data can be mapped onto an uncomplicated and easy-to-understand representation of the data.

2 BIO-AJAX

BIO-AJAX is a data cleaning toolkit for biological information systems that is designed to improve data quality at both data level and schema level. It adopts and modifies the conceptual operations originally developed in the declarative framework AJAX [7] so that they specially meet biological data needs. Previously, most data cleaning methodologies were specific to the domain or greatly tied into the physical layers of the database. Their heuristics were highly incorporated into the implementation of the data cleaning tool. AJAX, however, offered a way to encapsulate the primary operations of data cleaning to speak of the data cleaning techniques in an abstract manner. This gives flexibility to the system since the conceptual operations can be preserved while the instantiations of the operations can change as needed.

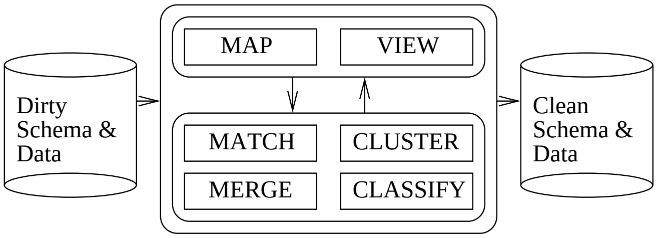


Figure 1: The BIO-AJAX framework.

Figure 1 shows the BIO-AJAX framework consisting of six interrelated operators. These operators each have specific individual purposes but can, and in some cases need, to work with the results from other operations. These operators are:

- MAP: translates the data from one schema to another schema.
- VIEW: extracts portions of data for cleaning purposes.
- MATCH: detects duplicate or similar records within the database.
- MERGE: combines duplicate records or similar records into one record within the database.
- CLUSTER: identifies sets of relations within the data and organizes the data into those relations.
- CLASSIFY: analyzes a given data point, categorizing it according to various domain rules.

In general, there can be multiple extensions and implementations of these operators. For example, there are a number of algorithms that can perform matching in any data set, including phylogenetic trees. The operator CLASSIFY uses various classification algorithms to perform many tasks such as classifying a protein or helping to standardize metadata. CLUSTER also performs operations associated with outlier detection in the data.

Implementing BIO-AJAX upon a data set first requires studying and detecting errors within the data set. This can involve using data mining algorithms catered to the data set to detect dirty data as well as discussions with curators and experts in the fields to understand observed problems. The next step is to identify algorithms that can effectively detect and clean the dirty data. These algorithms are tested on a small set of the database to gauge their effectiveness. If the algorithms are effective,

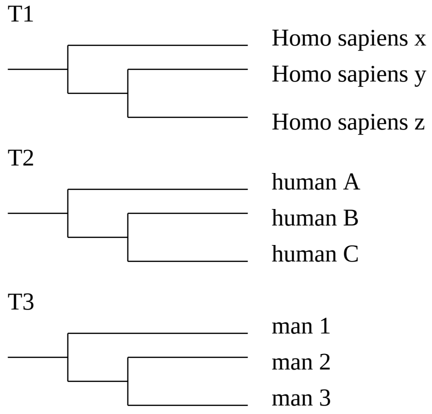


Figure 2: Example illustrating the nomenclature problem.

they are then implemented onto the entire database and integrated into the cleaning tool. The tool then runs automatically, without any more curator interaction.

3 BIO-AJAX and TreeBASE

One of the data quality problems concerning phylogenetic data in general, and TreeBASE specifically, is the nomenclature problem. In evolutionary biology research, various nomenclature issues can arise. See the NCBI TaxBrowser Website where there are resources entirely dedicated to updating users on nomenclature changes [5]. While the curators at TreeBASE specify directions for nomenclature, many submissions do not follow these directions.

Figure 2 illustrates this problem. In this example we have three trees, all of which model evolutionary relationships among the species "Homo sapiens". In the case of these trees, each tree studies evolutionary relationships between specific organisms within the species. T_1 shows the evolutionary relationships among three taxa "Homo sapiens x", "Homo sapiens y", and "Homo sapiens z". T_2 illustrates the relationships among three taxa "human A", "human B", and "human C". T_3 shows the evolutionary relationships among three taxa "man 1", "man 2", and "man 3". Now, consider tree T_1. The letters after the taxon name "Homo sapiens" indicate that a specific organism within the species or some other specificity is involved concerning the taxa within the tree. Therefore, this tree may not be generally about "Homo sapiens" but about a specific set of Homo sapiens organisms. In similar instances in TreeBASE, if we query for "Homo sapiens", most likely we will not get these three trees since the taxon "Homo sapiens" is not explicitly and exactly specified in the trees. (Only querying for "Homo sapiens x", for example, will allow us to get T_1.) Similarly, if we query for "human" or "man", we will not get the three trees even though all the three trees are related to "human", "man", or "Homo sapiens".

Thus, because of the inconsistency among the taxon names, we are unable to get a complete set of trees about the species "Homo sapiens". To solve the inconsistency and incompleteness problems, BIO-AJAX cleans the taxon names by implementing a layer between the user and the original database while not modifying the experimental data. Keeping the data intact is necessary since TreeBASE is an archival data repository where the original experimental data needs to be preserved.

Specifically, the operators of BIO-AJAX are instantiated to clean the nomenclature in the following manner (MAP and CLASSIFY are not relevant here and are omitted):

- VIEW: generates a list of prefixes for all taxa contained within TreeBASE.
- MATCH: compares the list of prefixes with the taxonomy entries in NCBI Taxonomy database. If an entry is found, obtain all nomenclature associated with the entry and index the names.
- MERGE: indexes the nomenclature so that any one of the prefixes, original nomenclature or nomenclature found in NCBI Taxonomy database can be used to query and obtain related trees.
- CLUSTER: uses hashing methods to group nomenclature and prefixes together that would refer to the same data within TreeBASE.

These cleaning tasks can be specified by high-level data cleaning scripts such as those used in [7]. For example, the VIEW operator can be specified as follows:

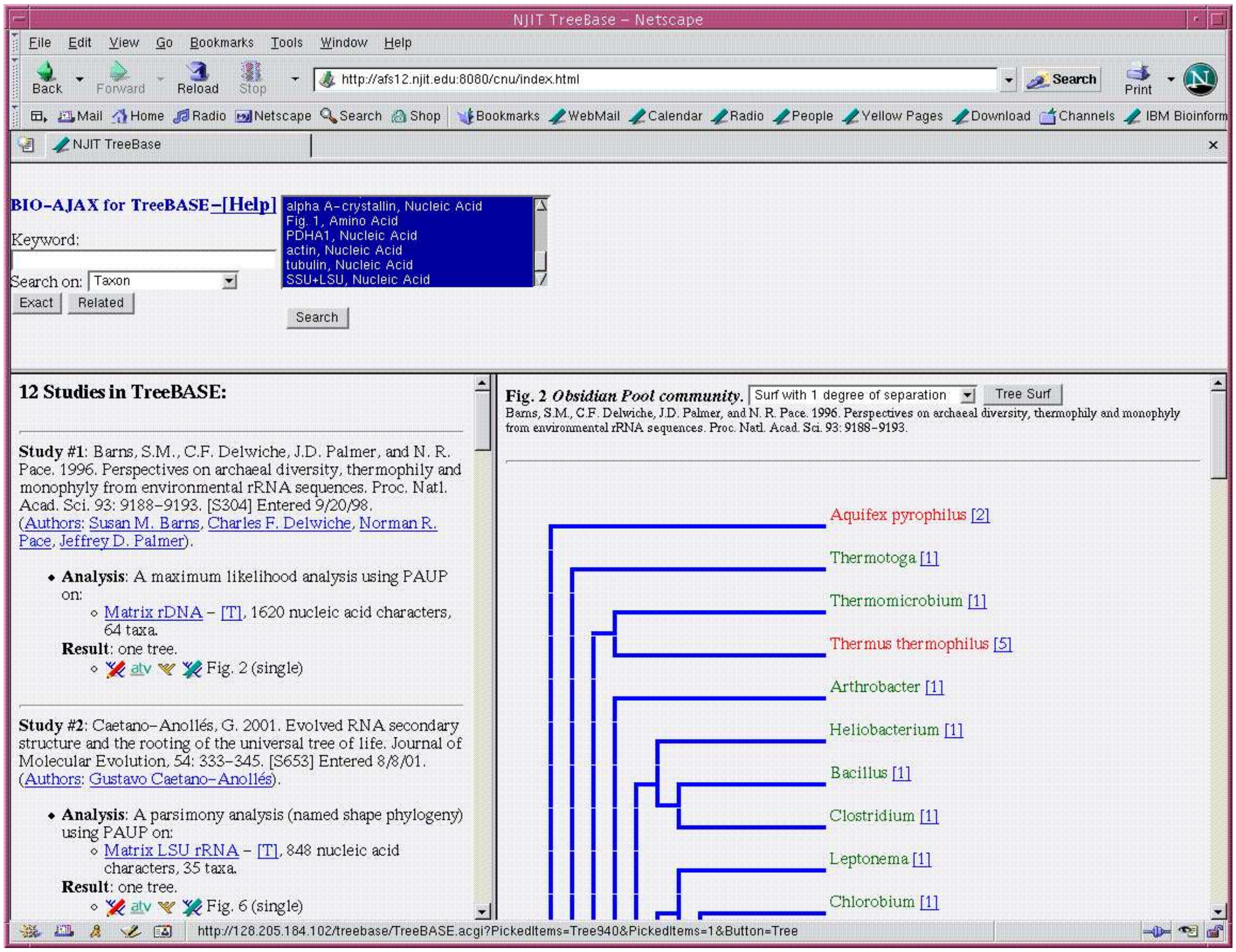
```
CREATE VIEW PhyloPrefixes
FROM DirtyTaxonData d
WHERE d <> null
{SELECT prefixGen(d.TaxonName)
AS prefix INTO PhyloPrefixes}
```

Here, the prefixes are created from the table DirtyTaxonData that contains the extracted data from TreeBASE. The function `prefixGen` produces the prefixes from DirtyTaxonData. Other operator specifications follow similarly.

This cleaning architecture offers some advantages over traditional methods such as record-based methods. Foremost, it allows for comparison of the nomenclature against a reliable resource (e.g. NCBI Taxonomy database) that is dynamically adjusted to the recognized state of the art within phylogenetics. Other methods for cleaning this data, such as record based comparison cleaning, would not be able to exploit such updates.

3.1 Implementation

BIO-AJAX is implemented using Perl, JAVA, JSP, HTML, and placed over TreeBASE as middleware to preserve


NJIT TreeBase – Netscape
File Edit View Go Bookmarks Tools Window Help
Back Forward Reload Stop
http://afs12.njit.edu:8080/cnu/index.html
Search Print
Mail Home Radio Netscape Search Shop Bookmarks WebMail Calendar Radio People Yellow Pages Download Channels IBM Bioinform
NJIT TreeBase
BIO-AJAX for TreeBASE–[Help]
Keyword:
Search on: Taxon
Exact Related
alpha A-crystallin, Nucleic Acid
Fig. 1, Amino Acid
PDHA1, Nucleic Acid
actin, Nucleic Acid
tubulin, Nucleic Acid
SSU+LSU, Nucleic Acid
Search
Surf with 1 degree of separation
Tree Surf
Aquifex pyrophilus [2]
Thermotoga [1]
Thermomicrobium [1]
Thermus thermophilus [5]
Arthrobacter [1]
Heliobacterium [1]
Bacillus [1]
Clostridium [1]
Leptonema [1]
Chlorobium [1]
http://128.205.184.102/treebase/TreeBASE.acgi?PickedItems=Tree940&PickedItems=1&Button=Tree

12 Studies in TreeBASE:

Study #1: Barns, S.M., C.F. Delwiche, J.D. Palmer, and N. R. Pace. 1996. Perspectives on archaeal diversity, thermophily and monophyly from environmental rRNA sequences. Proc. Natl. Acad. Sci. 93: 9188–9193. [S304] Entered 9/20/98. (Authors: Susan M. Barns, Charles F. Delwiche, Norman R. Pace, Jeffrey D. Palmer).

- **Analysis**: A maximum likelihood analysis using PAUP on:
  - Matrix rDNA – [T], 1620 nucleic acid characters, 64 taxa.

  Result: one tree.
  - atv Fig. 2 (single)

Study #2: Caetano-Anollés, G. 2001. Evolved RNA secondary structure and the rooting of the universal tree of life. Journal of Molecular Evolution, 54: 333–345. [S653] Entered 8/8/01. (Authors: Gustavo Caetano-Anollés).

- **Analysis**: A parsimony analysis (named shape phylogeny) using PAUP on:
  - Matrix LSU rRNA – [T], 848 nucleic acid characters, 35 taxa.

  Result: one tree.
  - atv Fig. 6 (single)

Fig. 2 ***Obsidian Pool community.***
Barns, S.M., C.F. Delwiche, J.D. Palmer, and N. R. Pace. 1996. Perspectives on archaeal diversity, thermophily and monophyly from environmental rRNA sequences. Proc. Natl. Acad. Sci. 93: 9188–9193.

Figure 3: The BIO-AJAX interface implemented for TreeBASE.

data integrity. Figure 3 shows the interface of the system. Due to the sensitive nature of biological data, specifically in this case phylogenetic trees, BIO-AJAX has been designed to never alter original submission data, but rather to interact with the original data and provide facilities to clean the interactions with the data. This is necessary, especially in an archival biological database cleaning since the data is associated with publications by the researchers.

To perform the nomenclature cleaning, first, all of the taxa are extracted from TreeBASE. The taxa are extracted with the name of the experimental study file that contains them. Once the taxa are extracted, they are organized lexicographically according to taxon names. The extraction file then goes through a rudimentary cleaning phase. This cleaning phase removes characters that could not possibly be a part of the taxon names as well as formats the extraction file for interaction. These characters, such as a forward slash before a taxon name, were determined to be extraneous characters during the analysis phase in the BIO-AJAX implementation for TreeBASE. Next, the file is formatted to become input for a prefix generation tool.

The prefixes are generated by producing all possible prefixes containing the first word of the nomenclature. (Most nomenclature consists of more than one word and the first word is an identifying term of a species.) For example, given the taxon “Homo sapiens x”, the following prefixes would be generated: “Homo”, “Homo s”, “Homo sa”, “Homo sap”, “Homo sapi”, “Homo sapie”, “Homo sapien”, “Homo sapiens”, “Homo sapiens x”. If the taxon name is one word long, then the prefix containing only that one word is created for that taxon ensuring every taxon in TreeBASE will be tested. Once the prefix list has been created, this list is then automatically used as input for the NCBI TaxBrowser query tool [1, 5, 15].

The NCBI Taxonomy database [1, 5, 15] is a repository of phylogenetic and taxonomic data about various species. The NCBI Taxonomy database provides tools to search and browse phylogenetic data about most species. Moreover, it conglomerates data from multiple resources about taxa and species. This data is dynamically updated as the Taxonomy database is updated, representing a concise representation of peer reviewed phylogenetic data. Therefore, it provides an ideal resource for solving the nomenclature problem in TreeBASE.

The list of prefixes is queried against the NCBI Taxonomy database's search tool TaxBrowser. This tool offers the user a number of options for searching for taxa. For BIO-AJAX's purposes, the following four types of searches are used: Complete Name search, Wild Card search, Token Set search and Phonetic Name search. The search for each prefix yields one of three possible results. First, the prefix is exactly found within the NCBI taxonomy database. If this is the case, the tool has found a "data page" about the prefix. The prefix and all other nomenclature NCBI associates with that entry are indexed. The results from each of these searches are then combined together. For example, consider again T_1 in Figure 2 where "Homo sapiens" is a prefix of "Homo sapiens x". When sending this prefix to the NCBI taxonomy database, we will get "human", "man" and "Homo sapiens" returned, and therefore all the three taxon names are linked with T_1. Similarly, when sending the prefix "human" ("man", respectively) of "human A" ("man 1", respectively) in T_2 (T_3, respectively) to the NCBI taxonomy database, we will also get "human", "man", and "Homo sapiens" returned. Thus, the three taxon names will be linked to all the three trees T_1, T_2 and T_3.

In the second case where the prefix does not return any match, then the prefix is discarded. Finally, if the prefix returns a hierarchical listing of possible matches, then an exact phrase search is performed within the list. If the exact prefix is found in the list, then that link is explored and treated as a data page. Only the exact match is used since, if each result in the list were used, then many taxa that are not related significantly enough to the original TreeBASE taxon would be included. For example, if the original TreeBASE taxon is "Homo sapiens", the prefix "Homo" would be generated from that taxon. The query "Homo" on TaxBrowser results in a list containing the taxa "Homo", "Homo sapiens", and "Homo sapiens neanderthalensis". Only the taxon "Homo" will be explored. "Homo" represents a genus that Homo sapiens belong to. Therefore, data about this group may be of interest to a user. If all of the list were explored, "Homo sapiens neanderthalensis" would have also been included as a possible association to "Homo sapiens". Since these are two distinct species, this is a relationship that should be eliminated when creating the associations.

Once the list is obtained from NCBI, it is indexed using hash tables. These tables allow for the exploitation of both the original data from TreeBASE and the data obtained from the prefix generation and querying. One index is comprised of the original taxa obtained from TreeBASE. The other index is comprised of the prefixes and other nomenclature obtained during the NCBI Taxonomy verification stage.

Now consider again the BIO-AJAX interface shown in Figure 3. This interface has been modeled similarly to the TreeBASE interface. The user can enter a nomenclature query in the top frame. This query is then formatted for interaction with the index. The query is checked against both indices described above. If a match is found in the index reflecting the original TreeBASE taxa, this match is considered an "Exact" match and any data linked to this match is highlighted as an exact match. If a match is found in the index that contains the nomenclature obtained from NCBI and the prefixes, then these matches are treated as "Related" matches. With each index, the matrix accession for its related studies is also housed with it. Once the matches are found, the data is then extracted from the TreeBASE database through using the matrix accession numbers of the studies. The results are then formatted, with the exact matches listed first, and displayed to the user similarly as to how the results are displayed on the TreeBASE website.

When searching, if the user wants to search for the taxon with exact matches in TreeBASE, he or she clicks on the "Exact" button. If the user wishes to search for any related matches to the taxon, he or she would click on the button "Related". The results of the search appear in the text box in the top frame. From there, the user can select one to all studies to be displayed. Once the studies are selected, they can be displayed in the bottom left frame. Data from these studies, including the phylogenetic trees, can be displayed in the bottom right frame.

For example, consider again the three trees in Figure 2. When the query is "Homo sapiens x" and the "Exact" button is clicked, T_1 will be returned. On the other hand, when the query is "Homo sapiens" and the "Exact" button is clicked, none of the three trees will be returned. However, if we click on the "Related" button when submitting the query "Homo sapiens", all the three trees will be returned.

3.2 Experiments and Results

To see how the proposed technique helps to gather a more complete set of trees related to a species, we conducted the following experiments. We extracted all of the taxon names from TreeBASE and used these taxon names as queries against NCBI TaxBrowser. This would reflect how well the TreeBASE nomenclature was compared with standard nomenclature. Moreover, the NCBI Taxonomy database also links to the TreeBASE database to provide users with more data about a given taxon. This experiment would reflect how well the taxa in the TreeBASE archive were recognized by the NCBI Taxonomy database.

For each taxon that is in a study or tree housed in the TreeBASE archive, it was used as a query for the NCBI TaxBrowser. If any data page was returned to the user, it

Search option	TreeBASE taxa	With prefixes
Complete Name	13,076 (49%)	23,493 (88%)
Wild Card	13,079 (49%)	23,493 (88%)
Token Set	13,172 (49%)	23,801 (89%)
Phonetic Name	14,025 (52%)	24,633 (92%)

Table 1: Results of taxa searches in NCBI TaxBrowser with TreeBASE nomenclature.

was considered a match. This data page could be one of two types of page. First, it could be a terminal page for a specific taxon's phylogenetic information. Second, it could be a hierarchical listing of possible matches. Of the 26,737 taxa obtained from TreeBASE, four experiments were performed with the different types of search (i.e. Complete Name, Wild Card, Token Set and Phonetic Name) available on TaxBrowser.

In addition, the prefix lists were tested. 290,118 prefixes were generated for the 26,737 taxa. If any one of the prefixes for a given taxon resulted in a match within the NCBI Taxonomy database, the taxon was considered as matched. The results are shown in Table 1. Thus, for example, in the table, for the Complete Name search option, 13,076 taxa out of the 26,737 taxa, which were about 49% of the taxa obtained from TreeBASE, resulted in a match within the NCBI Taxonomy database. On the other hand, for the same search option and with the prefix generation technique, 23,493 taxa out of the 26,737 taxa, which were about 88% of the taxa obtained from TreeBASE, were considered as matched.

The results in Table 1 reflected how BIO-AJAX found a number of taxa related to the TreeBASE taxa that were not previously detected. This has a number of implications for the cleaning method as well as TreeBASE. First, within the studies archived in TreeBASE, while efforts have been made to minimize the nomenclature problems, only approximately 50% of the taxa in TreeBASE are recognizable by the standards within the phylogenetic community stored in the NCBI Taxonomy database. Moreover, since many data repositories link to TreeBASE, such as the NCBI Taxonomy database, this has implications for them as well. Second, with the prefix generation technique, the recognition of the taxa can be improved to approximately 88%. This helps to solve the inconsistency and incompleteness problems concerning the nomenclature in TreeBASE as well as any other tool that links to TreeBASE.

4 Conclusion and Future Work

This paper presents BIO-AJAX, a cleaning framework for biological data. The toolkit has been implemented and placed, as middleware, over the phylogenetic information system TreeBASE. A prefix generation technique for instantiating the operators in BIO-AJAX has been developed, which helps to clean the nomenclature in TreeBASE. In contrast to existing work for cleaning relational records or genome sequences (see, for example, [6, 9]), the techniques presented here are focused on evolutionary trees and hence are different from those existing data cleaning methods.

Future work concerning this framework includes creating more data analysis tools such as viewing the statistical distributions of taxa or co-occurrence of taxa [12]. Another direction is to extend BIO-AJAX to other data repositories. This involves the implementation of an API (application programming interface) for wrapping the taxonomy source such that other taxonomy (say, enzyme classification, gene ontology, etc.) can also be used so long as they adhere to the required interface. There may exist inconsistencies among the different taxonomy sources and how to resolve them remains to be a challenge research problem, both in phylogenetics and in data engineering.

We also plan to study other data types including those in the Protein Information Resource (PIR). PIR is an integrated public resource of protein informatics to support genomic and proteomic research and scientific discovery [17]. Through the development of PIR databases, a number of problems have been observed that could benefit from the BIO-AJAX data cleaning architecture. For example, in [16], two types of cleaning issues are identified: protein classification and functional annotation.

Protein classification is a complex area with a number of problems. First, there is no standard way to compare proteins. Therefore, different methods turn out different results. Another problem is that many of the protein classification issues have arisen from the data integration phase of eliminating redundancy from the repository. For example, it has been noted that there are numerous errors found in genome annotation [2, 4]. Moreover, with current genome annotation standards, many proteins are defined only to have one function. This can limit the ability to classify the proteins properly since most proteins are multifunctional. This leads to the concerns facing the functional annotation. A possible method for addressing this problem could be adopting feature extraction for proteins [13, 14] and using extracted features to categorize the proteins.

Functional annotation itself is used in many aspects of protein data repository management. For example, it can help to classify unknown protein sequences. However, current functional annotation has many problems. Besides the aforementioned problem, a major issue of functional annotation is concerned with protein name ontology [8]. Such an ontology is important, as the protein name is the form in which a protein object is referred to and communicated in the scientific literature and biological databases. As with the nomenclature problem in TreeBASE, there is also a long-standing problem of nomenclature for proteins. Scientists may name a newly discovered or characterized protein based on its function, sequence features, cellular location, molecular weight, or other properties, as well as their combinations or abbreviations. A protein name ontology provides a basis for consistent database annotation where functional conservation is curated with a common language.

BIO-AJAX offers possible improvements and solutions to these basic problems. BIO-AJAX allows for the extension of multiple algorithms for classifying proteins as well as creates environments for these algorithms to interact with each other. It also affords the opportunity to implement and extend new algorithms and methods. This allows for BIO-AJAX to examine how well the algorithms perform classification and interact with the older methods. We plan to extend BIO-AJAX to clean data in protein repositories, specifically in PIR.

Acknowledgment

We thank the anonymous reviewers for their constructive suggestions, which helped to improve the paper.

References

[1] Benson, D.A., Karsch-Mizrachi, I., Lipman, D.J., Ostell, J., Rapp, B.A., and Wheeler, D.L. “GenBank.” *Nuc. Acids Res.*, 28(1):15-18, 2000.

[2] Brenner, S.E. “Errors in Genome Annotation.” *Trends in Gen.*, 15:132-133, 1999.

[3] Dasu, T. and Johnson, T. *Exploratory Data Mining and Data Cleaning*. John Wiley & Sons, 2003.

[4] Devos, D. and Valencia, A. “Intrinsic Errors in Genome Annotation.” *Trends in Gen.*, 17:429-431, 2001.

[5] Federhen, S., Harrison, I., Hotton, C., Leipe, D., Soussov, V., Sternberg, R., and Turner, S. NCBI Taxonomy Homepage. `http://www.ncbi.nlm.nih.gov/Taxonomy/taxonomyhome.html/`.

[6] Gajer, P., Schatz, M., and Salzberg, S.L. “Automated Correction of Genome Sequence Errors.” *Nuc. Acids Res.*, 32:562-569, 2004.

[7] Galahardas, H., Florescu, D., Shasha, D., Simon, E., and Saita, C.A. “Declarative Data Cleaning: Language, Model and Algorithms.” In *Proc. of the 27th International Conference on Very Large Data Bases*, 2001, pp. 371-380.

[8] Hirschman, L., Park, J.C., Tsujii, J., Wong, L., and Wu, C.H. “Accomplishments and Challenges in Literature Data Mining for Biology.” *Bioinformatics*, 18(12):1553-1561, 2002.

[9] Low, W.L., Lee, M.L., and Ling, T.W. “A Knowledge-Based Approach for Duplicate Elimination in Data Cleaning.” *Information Systems*, 26(8):585-606, 2001.

[10] Ludäscher, B., Gupta, A., and Martone, M.E. “A Model Based Mediator System for Scientific Data Management.” Eds. Z. Lacroix and T. Critchlow, *Bioinformatics: Managing Scientific Data*, Morgan Kaufmann Publishers, 2003, pp. 335-370.

[11] Piel, W.H., Sanderson, M.J., and Donoghue, M. “The Small-world Dynamics of Tree Networks and Data Mining in Phyloinformatics.” *Bioinformatics*, 19(9):1162-1168, 2003.

[12] Shasha, D., Wang, J.T.L., and Zhang, S. “Unordered Tree Mining with Applications to Phylogeny.” In *Proc. of the 20th International Conference on Data Engineering*, 2004.

[13] Wang, J.T.L., Ma, Q., Shasha, D., and Wu, C.H. “New Techniques for Extracting Features from Protein Sequences.” *IBM Systems Journal, Special Issue on Deep Computing for the Life Sciences*, 40(2):426-441, 2001.

[14] Wang, J.T.L., Marr, T.G., Shasha, D., Shapiro, B.A., Chirn, G.W., and Lee, T.Y. “Complementary Classification Approaches for Protein Sequences.” *Protein Engineering*, 9(5):381-386, 1996.

[15] Wheeler, D.L., Chappey, C., Lash, A.E., Leipe, D.D., Madden, T.L., Schuler, G.D., Tatusova, T.A., and Rapp, B.A. “Database Resources of the National Center for Biotechnology Information.” *Nuc. Acids Res.*, 28(1):10-14, 2000.

[16] Wu, C.H., Huang, H., Yeh, L.S.L., and Barker, W.C. “Protein Family Classification and Functional Annotation.” *Computational Biology and Chemistry*, 27:37-47, 2003.

[17] Wu, C.H., Yeh, L.-S., Huang, H., Arminski, L., Castro-Alvear, J., Chen, Y., Hu, Z., Kourtesis, P., Ledley, R. S., Suzek, B.E., Vinayaka, C.R., Zhang, J., and Barker, W.C. “The Protein Information Resource.” *Nuc. Acids Res.*, 31(1):345-347, 2003.