International Journal of Medical, Medicine and Health Sciences
ISSN: 2517-9969
Vol:1, No:6, 2007

A System to Integrate and Manipulate Protein Database Using BioPerl and XML

Zurinahni Zainol, Rosalina Abdul Salam, Rosni Abdullah, Nur'Aini, and Wahidah Husain

***Abstract*—**The size, complexity and number of databases used for protein information have caused bioinformatics to lag behind in adapting to the need to handle this distributed information. Integrating all the information from different databases into one database is a challenging problem. Our main research is to develop a tool which can be used to access and manipulate protein information from difference databases. In our approach, we have integrated difference databases such as Swiss-prot, PDB, Interpro, and EMBL and transformed these databases in flat file format into relational form using XML and Bioperl. As a result, we showed this tool can search different sizes of protein information stored in relational database and the result can be retrieved faster compared to flat file database. A web based user interface is provided to allow user to access or search for protein information in the local database.

***Keywords*—**Protein sequence database, relational database, integrated database.

I. Introduction

OUR major activity in Bioinformatics is the management of the database which contains DNA, RNA or protein data. These databases are used by biologists to manipulate and analyze data. One of the basic operations that need to be implemented on the protein database is to find the protein sequence. Currently, the local scientists use BLAST [15] or FASTA to find the protein sequences and other information in the public domain database.

Public sequence databases contain one of the most frequently accessed information on the web [2]. Databases such as Swiss Prot and EMBL provide descriptions of common biological entities (genes, proteins, among others). Numerous computational methods and algorithms in data mining have been implemented to extract hidden knowledge in these databases.

Manuscript received May 19, 2005. This work was supported by University Science Malaysia (USM) under USM short term grants no 304/PKOMP/ 635070.

Zurinahni Zainol is Senior Lecturer at the School Computer Science, USM, Malaysia (e-mail: zuri @cs.usm.my).

Rosalina Abdul Salam is Senior Lecturer at the School Computer Science, USM, Malaysia (e-maill: rosalina @cs.usm.my).

Rosni Abdullah is Associate Professor at the School Computer Science,USM, Malaysia (e-mail: rosni@cs.usm.my).

Nur'Aini is Senior Lecturer at the School Computer Science, USM, Malaysia (e-mail: nur'aini@cs.usm.my).

Wahidah Husin is Senior Lecturer at the School Computer Science, USM, Malaysia (e-mail: wahidah@cs.usm.my).

Therefore, the problem of managing enormous databases is compounded by the fact that if the scientist wants to receive the maximum benefit from this information, they must compare and relate information from difference databases. A new standard is needed to interrogate biological data stored in different types of data presentation [1].

Further more; standardized data formats(s) are also needed to exchange data because all the information is stored in different format databases. Our approach will integrate all different protein databases into once central database for easier and faster access by end users. Besides that our system will also provide a tool enabling data management and search facilities.

II. Related Work

In order to integrate the different protein databases, data exchange standard is needed to integrate these databases. There are many standards currently implemented such as flat files, common Object Request Broker Architecture (COBRA), Abstract Syntax Notation Number 1(ASN.1) and Extensible Markup Language (XML).

Flat files contain machine-readable data that is typically encoded as printable characters. A flat file database usually contains a series of record, where each record is a sequence of fields. Much of it is stored in flat files as tab-delimited text, comma-separated values, or some similar format. The usage of flat file in protein database have a few problems such as accessing the data is limited to the single user, high probability of data damage, data control is very difficult and data are highly redundant. CORBA is a good solution for developing and deploying application in heterogeneous environments [4]. CORBA infrastructure was used for development at EMBL-EBI [8] and proof that the CORBA interface can address some of the limitation of the flat-files format. It provides means for accessing and distributing EMBL Data [5]. Disadvantages of CORBA are the construction of servers in CORBA is difficult and time consuming. Interface Definition language (IDL) places too many restriction on how to use the data and make it user hardly to use [7]. ASN.1 is also being used at NCBI to store data as well as send it and received it. The common known formats that are produced from NCBI database is GENBANK flat file. The disadvantages of ASN.1 are the values are in binary format; need the expertise to guide in order to understand the result [5]. XML is an emerging standard for storing biological data in a structured format [8]. XML is

highly flexible and easy to understand and interpreted. XML is internet oriented and has very rich capabilities for storing data. It also provides an open framework for defining standard specification. Several searching tool for protein database have been implemented such as Uniprot[10], ExPasy[11] international protein [12], pfam[13]and OWL[14]. Most of the database they used is Swiss Prot, TrEmbl, PIR and Genbank. All of these databases were mainly in flat-file format. Our main focus is to provide a platform of central integrated database in a standard format so that the searching and manipulating the protein data will be easier. We also provide a tool to search and manipulate protein information.

III. Methodology

Protein databases can be created using flat files or relational database management system such Oracle, DB2 or MySQL. In our proposed system, we will transform the existing database and data format into XML format. The proposed system can form the platform for protein databases that support large entities of data stored in a flexible format. The user can use native storage facilities in flat files, relational DBMS. The overview of the system proposed is shown in figure 1.

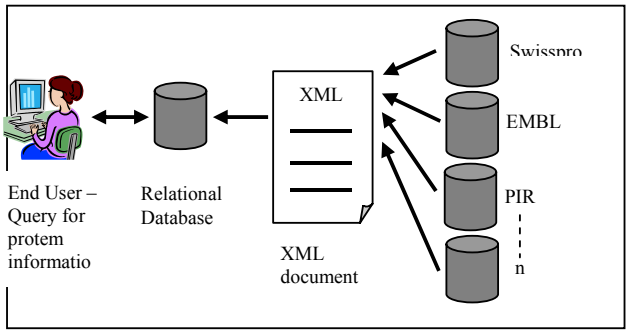

Fig. 1 Proposed System Overview

From the figure 1, we can see when the users request for protein information, the result will be retrieved from the relational database. This database will be updated occasionally with latest information from different protein databases. The flow of the system can be divided into 3 sections. There are as follows

(a) Convert information from different databases into XML format
(b) Construct Relational database & transform XML data into relational database
(c) Display result based on user query

Each of this proposed solution would be discussed in detail starting with convert information from different databases into XML format, construct Relational database and transform XML data into relational database and followed by display result based on user query.

A. Convert information from different databases into XML format

Informations stored in protein database are in flat file formats. The XML provides the Document Type Definition (DTD) interchange. With DTD, the data is verified by specified the required attribute, number of times an attribute appear and type of data item. Using XML parser the XML document will be translated into XML parse tree. Figure 2 shows the process flow to generate XML document and transform into relational database (*protein_info*).

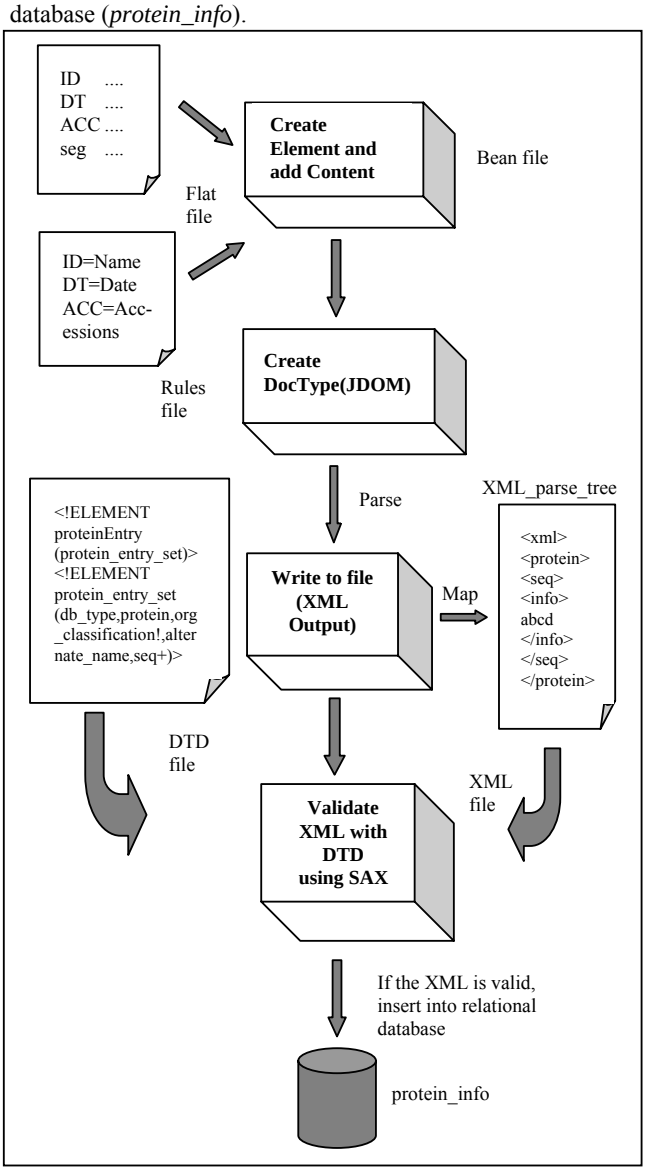

Fig. 2 Process flow to generate the XML and transform into relational database(*protein_info*)

B. Construct Relational Database

We provide a server with Bioperl module i.e. Bioperl DB and Bio SQL. We have constructed database called *Protein_info* using MYSQL for RDMS. *Protein_info* have 27 tables. Each of the table is constructed based on BIOSQL database schema generated from *biosqldb_mysql* script. The 27 tables are classified into 4 main divisions to ease the design. There

are *seqfeature*, *bioentry*, *terms* and *relation*. B*iosqldb_mysql* script creates an instruction to create table, primary key, secondary key and determine the relation between each table. Figure 3 shows the entity relationship diagram for the *protein_info.*

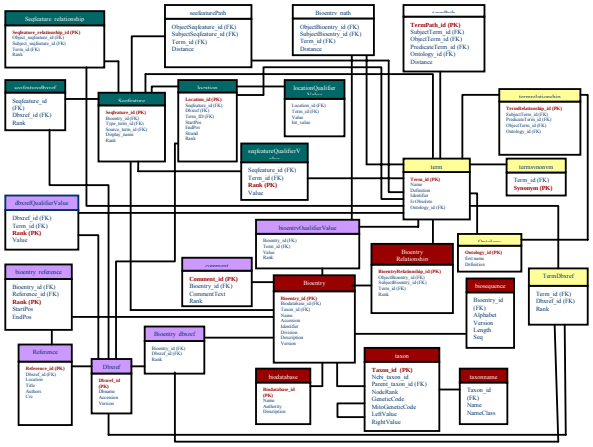

Fig. 3 Entity Relationship diagram for the *Protein_info*

C. Display Result based on User Query

We provide a searching tool using bioperl[9] and SQL. This tool is used to find/search the sequence and other information in the *protein_info* database based on sequence, id or species name.

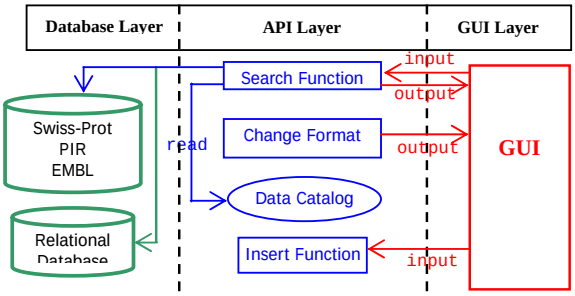

Fig. 4 System's architecture

From the figure 4, the user can access the system via web to manipulate and search the information from the relational database i.e *Protein_info* by using Search Function. We used CGI/Perl to access the Bioperlmodules. The Search Function, Insert function, Change format and Data Catalog are the Bioperl functions.

IV. Implementation

We implemented this system on a PC cluster machine called TIARA installed with Apache HTTP server. The tool is written using Bioperl modules and C/C++ programs. The graphical user interface uses HTML and JAVA script with CGI Perl to interface with the backend program. MySQL is used to store the protein data. The Web based system is available at http://tiara.cs.usm.my/Bioprotein.html. We have used Swiss-Prot, EMBL, PIR as sequence database.

As described at figure 2 in section III, this system will map protein flat files from the flat file databases into XML and changed into a relational database. The conversion of the information from different databases will be transformed into one standard format. We have developed one *rules file* to store all mapped information. The mapped information is written into a file with extension *.swissprot* for Swissprot database, *.Pir* for pir database and etc. The mapping file is created only once and can be used every time user wants to change the protein information into the database.

The *rule file* is created with the protein flat file. The *bean file* is also created to extract out the respective fields of information from different protein databases. Each of the protein databases has their own *bean file*. By using Doc type, XML *Out putter* is used to generate XML. Generated XML file with predefined DTD will be used to validate the XML. Data will be inserted to the relational database if the XML file is valid.

Once the protein flat files is converted into XML and validated, the data can be inserted into relational database i.e *protein_info* which has been constructed as mentioned in section III. In order to evaluate and compare the performance of the system, search module is provided. Each of the queries to *protein_info* using SQL forms in the Perl Language. Searching module is developed to search sequences and other information based on sequence, ID or species requested by the user. Example of protein sequence: MASVKSFILILSQVIECQPOS, example of ID protein: 108_LYCES and example of species is proteinTheileria parva.

V. Experimental Results

The performance of search is evaluated using various measurements such as reliability, accuracy and performance. Each of this measurement will be discussed in details. 100000 data protein from difference protein databases have been downloaded into the Tiara server. In this system, the final results are displayed through search result. The analysis will be focused on search module.

i. Reliability

Reliability of the information was evaluated by comparing the search results in relational database with the original flat file database. There are 23 unique common names in the database. The features used to evaluate are contents of information. The result is same as in flat file format and all the values mapped able to retrieved from relational database is same as in flat files.

ii. Accuracy

Accuracy evaluation was done to determine the ratio of retrieved data protein is relevant. Execution of this evaluation

for search is important because users are interested in getting relevant and correct information. The search result for the flat file is accurate when user input same as data as in a flat file. For the sequence input cannot has small letter and there is no gap in between the words and for the species name, the name must be started with capital letter and input for ID must in capital letter too. Whereas, by using the relational database, the searching is more accurate compare to flat file database. There are no case sensitive for species name, ID and sequence.

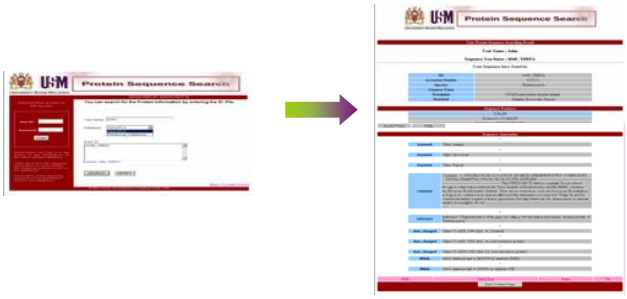

Fig. 7 Interface screen Snapshots for searching protein data Relational database using ID, Sequence and Species name

iii. Performance

The performance was evaluated based on the time taken to return the search result. For the flat file database, the searching is started from the first record until record 100000. From the figure 8 shown, the searching performance for data protein in relational database form record 10000 to record 100000. The maximum time recorded is 0.03 second but the minimum time is 0.01 second. Time taken to search data is not more than 0.01 second. The ratio of the searching time is independent from the data location but rely on server time.

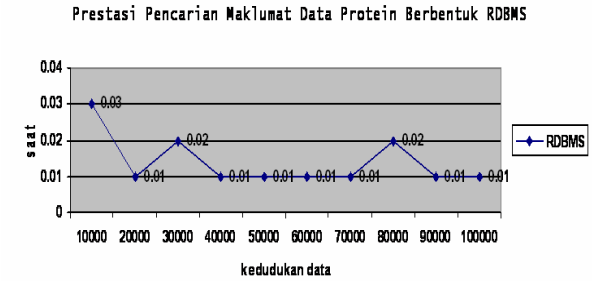

Fig. 8 Execution time for searching protein data Relational database

Figure 9 shown, the searching time for the flat file format compare to relational format started from data location at record 10000 until record 100000. The difference time taken between these two formats databases is 2023.97 second and minimum time difference is 227.99 sec. The different between minimum and maximum access time shows a big range between this database formats. We have shown that a relational database is better than flat file format for searching and manipulating protein databases.

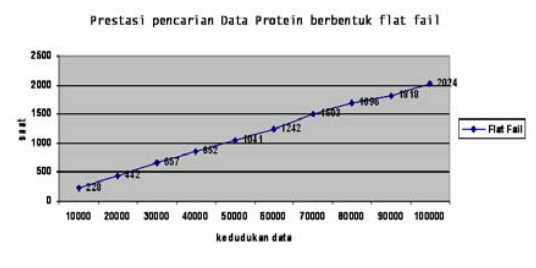

Fig. 9 Execution time for searching protein data flat file database

VI. Conclusions and Future Work

In this paper we present a new approach to search and manipulate protein sequence using Bioperl and XML in a PC cluster system. Our system significantly reduced the processing time and user friendly. We also have implemented a system that can integrate different database protein flat file into XML format and stored the protein data in a relational database. Based on the analysis results, XML is better approach to integrate different databases. The information transferred using XML and Bioperl is reliable and readable and it is proven based on search result. Further work, we are going to implement this system in other platform such as grid computing to reduce the communication overhead among the PCs.

Acknowledgment

Our thanks to those who have developed the system for our group; Mekala Nagasamy, Mohd Hafiz, Mohd Izham and Mohd Najmee.

References

[1] Guochun Xie,Reynold DeMarco,Richard Blevins and Yuhong Wang, *Stroing biological sequence databases in relational form*, http://www.bioinformatic.oupjournals.org, 1999.
[2] Andre Bergholz,Jorg A. schenk, stepehn Heyman,Johann Christoper , *Sequence comparison using a relational database approach*,http://www.citeseer.ist.psu.edi/bergholz97sequence.html, 1997.
[3] P.mork,A.halevy, P.tarczy, *A model for data integration system of Biomedical Data Applied to Online Genetic Databases*, 2000.
[4] Wang L., Riethiven-Tom, P., N,McNail P.,Robinso Redaschi, *A.,Lijnzaad,Exploiting XML with CORBA to improve Distributing EMBL data*, EMBL Outstation , European Bioinformatics Institute,2001
[5] Wang L., Riethiven-Tom, P., N,McNail P.,Robinso, *Accessing and distributing EMBL data using CORBA*, Genome Biology 2000 1(5): research, 2000
[6] E.V. Kriventseva, W.Flieschman, E.M Zdobnov, R. Apweiler, *CluSTr: A database of clusters of Swis-sprot + Trembl Protiens, Nucleic Asids Research*, Vol 29, No1, pg 33 – 36, 2001
[7] Emmanuel, B,Leser,U. Lijnzaad,P,Cussat-Blanc,Jungferm K.Guyon,F., Vaysseix, G, Jhelgesen,C., and Rodriguez-Tome, *P. A Proposal for a standard CORBA interface for genome Maps, Bioinformatics*, vol 15, No 2, , pg 157 – 169, 1999
[8] http://www.w3.org/XML/
[9] http://www.bio.perl.org/
[10] http://www.ebi.uniprot.org/uniprot-srv/uniprotsearch
[11] http://au.expasy.org/
[12] http://pir.georgetown.edu/pirwww/dbinfo/pirpsd.html
[13] http://pfam-wust1.edu/hmmsearch.shtml
[14] http://umber.sbs.man.ac.uk/dbrowser/OWL
[15] S.F. Altschul et al., "*Basic Local Alignment Search Tool*,", Journal of Molecular Biology 215, 403-420, 1990