Biological Data Handling Methods

Pradeep Achan[1], **Ajit G Warrier**[1], **and Bhadrachalam Chitturi**[2]
[1] School of Biotechnology, Amrita Vishwa Vidyapeetham University, Amritapuri, Kollam, India
[2] Department of Computer Science, Amrita Vishwa Vidyapeetham University, Amritapuri, Kollam, India

Abstract *-Biological data has more variation in type and format compared to other types of data. Thus, it poses new challenges. However, it encapsulates critical information; thus, handling it is of primary interest. Data handling includes storage and retrieval of data with associated formats and methods of data transfer, data format conversion, algorithms that run on the data and the output methods including visualization of the results. High throughput methods have been yielding biological data at a fast pace. This data includes protein-protein interactions, gene sequences, gene co-expressions, and protein sequences. This data is supplemented with huge amounts of clinical data conveniently captured in electronic medical records and the wet lab data. We describe the current approaches, each with a model system and identify its key contributions. We propose some ideas for biological data handling in the future.*

Keywords: biological data handling, cloud computing, data integration, data modeling, semantic web, systems biology

1 Introduction

The term biological data is used in a broad sense. It includes genomics/proteomics data, the data generated from experimental biology, diseases data and patient clinical data. High throughput screening has been yielding large quantity of new data in biology. Micro array analysis provides gene co-expression data, the next generation sequencing, *i.e. NGS*, yields DNA sequences and so on. Even though various types and formats of the data pose challenges the information in the data is vital. Biological data is distributed in various sources; it has redundancy, different formats and naming conventions. A researcher potentially needs the information from various sources. The features that contribute to the difficulties in handling of such data are: 1) the quantity of the data, 2) various sources, formats and naming conventions, 3) the dynamic nature of the data and 4) the complex relationships between several data objects which can be of various types.

The enormity of biological data renders *warehousing* (*i.e.* data warehousing), computing, transmission of the data over the network difficult owing to higher requirements in storage space, computation and bandwidth. Also, integration of large quantities of data is resource intensive. Examples of the data formats are a flat file such as "tab delimited format" or a database dump such as MySQL database dump or an excel spreadsheet. A protein can be addressed with various names in various databases owing to diverse naming conventions. A researcher typically collects genome data, literature abstracts, protein information, pathways, and 3D structure from Genome database, PubMed, Uniprot, KEGG and PDB respectively [42]. New entries of a given object type and new relationships are continually discovered. For example, a new protein (of object type "protein") can be discovered. Likewise, a previously unknown interaction can be detected between a pair of proteins present in the database. Thus, the data is dynamic in nature. This causes problems in systems with a warehouse or without it *i.e.* federated system. A central repository will be outdated if new data is added to external sources after the last update. A federated system might become dysfunctional due to schema modification at one or more data sources.

Keeping these databases up to date and in phase with each other is quite challenging, more so in the wake of NGS technologies. Consider a system with a warehouse C which uses data sources $S = \{s_1, s_2, \ldots, s_q\}$. As stated earlier, C can have older data compared to S. Also, the data in S can be inconsistent. Consider a scenario where a new gene g and a protein p coded by it are discovered. Let p interact with a known protein q. Say s_i, s_j and s_k have protein (with foreign key to gene), gene and protein-protein interaction (PPI) data respectively. Some of scenarios where the data in S is incomplete are: (a) s_j is updated with g, s_i is updated with p whereas s_k is not updated accordingly (it does not have PPI for p and q), (b) s_j is updated with g whereas s_i and s_k are not correspondingly updated, (c) s_j and s_k are correctly updated whereas s_i is not correspondingly updated. In (a) just the interaction information is missing but it does not have any serious inconsistency. In (b) both the protein and the PPI are missing which is a minor inconsistency because we do not find the protein for a given gene. In (c) the critical link between g and the interacting pair p and q is missing. Thus, C can have two types of problems; *i.e.* it can have outdated data compared to S or it can be in phase with S and yet inherit inconsistency that is inherently present in S. These problems point to the need for frequent access to the information across different databases which are spread across different Internet data sources, consistency check of the data and the practical limitation of having large databases (multi-terra bytes) warehoused centrally due to the limitation of storage space.

Biological data has unique complexity and levels of abstraction as detailed in Section 2. The processing of biological data involves various tasks that depend on the application and the input data. One can broadly subdivide the process into the following chronological sequence of four tasks: a) data acquisition and preprocessing, b) analysis of relationships between data objects c) creating a data model for a given application, and d) creating output.

Data acquisition refers to acquiring the data from the data source(s). Data is often stored in various formats, e.g. flat files, spreadsheets etc. which are not directly conducive to computation. Such data is often converted into a database table; this step is *preprocessing*. *Analysis of relationships* between data objects primarily refers to the domain knowledge; e.g. the relationship between: a protein and a domain, a gene and protein etc. Analyses of the relationships between data objects are represented as structured information in database systems. These are read into application-specific in-memory organization of data. This application specific data organization in database system as well as in-memory data structure can be called as the *data model* of the application. Application can process the data model to create secondary information by selection (retrieving specific pieces of information), aggregation (aggregating information from different sources), or mining (for patterns within the data). The application presents the results of a query as *output*.

Methods for data acquisition and preprocessing are well established and the analysis of relationships is achieved with the expertise provided by biologists. We discuss Output in some detail. Data modeling and the associated task of data integration are more thoroughly covered. Data model which comes from the analyses of relationships can be viewed as a template; when it is executed, it results in data integration.

In Section 2, different approaches for building systems are described with a special focus on the emerging semantic web methodologies. Section 3 details handling of the output. Section 4 gives the features provided by cloud computing. Section 5 details a few recent innovative projects. Section 6 states key findings from different approaches and lists open problems and the work that mitigates some of these problems. It also states some desirable features for the future biological data handling systems.

2 Approaches for biological systems

A system has certain functionality and it is built with a specific approach. In this section, we discuss approaches for building such systems. Subsections 2.1 and 2.2 explore the approaches of the vital aspects of such systems, *i.e*. data integration and data modeling respectively.

2.1 Approaches for data integration

Data integration needs for applications vary considerably with the user who can be a biologist, bioinformatician or a systems biologist. A review of various integration approaches is given in [7], where they are labeled as light to heavy in terms of integration efforts.

Integration techniques which include the use of scripts written in Perl and Python [42] exist. Service based methods like WSDL an XML format provides a model for describing Web services [42]. [20,46] classify data integration approaches into warehousing, mediator or view integration and also as link or navigational. [46] describes the use of Web Services, Distributed Annotation System (DAS) and Globally Unique Identifiers in data integration and also proposes an approach, termed as "knuckles-and-nodes approach", where in the source databases remain independent but a few important relationships are stored in special-purpose linking databases. In addition the use of scripting, peer-to-peer systems, semantic web technologies and workflow-based were introduced in [42]. The approaches mentioned in [42, 20, 46] overlap with each other in various aspects; *i.e*. technological choice, methodology etc. Also they are not mutually exclusive but use or depend on some others for effective data integration. Link integrations are used in building systems based on either relational model or semantic web technology.

Archival databases like NCBI, EMBL, DNA Data Bank of Japan, maintained by International Nucleotide Sequence Database Collaboration accept data directly from sequencing labs and are referred as primary sequence database [47]; they aggregate data centrally. Similarly, primary protein sequence databases include PIR and UniprotKB (Swiss-Prot/TrEMBL) which handle the protein sequences. Other systems act as value added integrators of this data such as Ensembl, UCSC Genome Browser, Uniprot and Model organism databases [47]. These provide data in convenient formats for further aggregation and analysis. Secondary data sources like PROSITE, PRINT, Pfam aggregate data centrally and also link to primary data sources by unique identifiers.

Most of the primary and secondary databases link to other information sources through link integration. Some systems are built by power (advanced) users from these primary and secondary sources for custom application systems [47]; they may be general purpose or special purpose systems [37]. We refer to them as *tertiary* systems, *e.g*. BioWarehouse [35], ATLAS [44] and ONDEX [31]. All of these aggregate data. In contrast, TAMBIS [48], BIO-BROKER [1], and SEMEDA [32] use a mediator approach *i.e*. they use a wrapper to access original data sources.

Some other systems [46] store a part of data in a warehouse in addition to the use of mediation for effective integration. In [30] another approach was introduced to integrate gene expression data and proteins stored in data warehouse with annotation data retrieved from public sources using sequence retrieval system. The above mentioned integration methods [30,46] are also termed as hybrid systems. SADI does not store data locally and links with other systems using REST-based [15] web services [54].

The advantages of warehousing approach are: it relies less on network [20], allows faster query performance, allows the system to filter, validate, modify, and annotate the data obtained from the sources [20], *e.g*. BioWarehouse [35]. It also facilitates the integration of locally derived experimental data into the repository. However, it needs large storage (the biological data is semi-structured and is not easily stored in relational databases (or simply *RDB*s) [42] and it must be synchronized with underlying sources for updates [49]. Biological data needs significant computation to be stored in the typical format *i.e*. RDBs.

Semantic web is an emerging technology by WWW consortium describing it as "web of data" [22]. An informal definition for the Semantic web technologies could be "comprising of four essential component technologies namely

RDF, RDFS, OWL and SPARQL" [2]. Semantic web uses uniform resource identifier, URI, to represent a data object, mostly in a triple containing *subject*, *predicate* and *object*. This triple, which uses three URIs, is called Resource Description Framework (RDF) [23]. A *triple store* stores this triple (RDF data). RDF represents the information or data as a graph. RDFS and OWL [24] are ontology languages. Querying the RDF graph is done with a querying language similar to SQL called SPARQL [26]. A SPARQL query is denoted by a graph pattern containing the patterns of triples that are similar to RDF triples but are replaced with variables.

Current usage of Semantic web technologies for biological knowledge management has been described in [2]. Knowledge management refers to the process of systematically capturing, structuring, retaining and reusing information to develop an understanding of how a particular system works, and subsequently to convey this information meaningfully to other information systems. [2] lists selected resources and projects which use Semantic web technologies and suggests more prevalent use of it in future systems.

Majority of the data is stored in RDBs and it is difficult for Semantic web technologies to access them. Thus, an application tries to create its own relational to semantic mapping and thereby accessing the relational data using SQL. Semantic web layer can play a great role in integrating relational data into Semantic web technologies, it defines the standard vocabularies, formal models and semantic relations between RDBs [9]. Datagrid [9] framework along with a set of practical semantic tools was used to facilitate the integration of heterogeneous RDBs using Semantic web technologies. OWL [41] is a technique to extract the semantics of a RDB and transform it into RDF/OWL. It extracts the schema information of the data source and converts it automatically into ontology. With this technique every RDB can automatically be an integral part of Semantic web. Thus, web applications can access and query data stored in RDBs using their own built-in functionality [41]. Jiang et al. describe an architecture to expose RDB to Semantic web application using Hibernate [18]. OWL ontology is translated to java classes and then a runtime SPARQL to hibernate query language (HQL) translation algorithm was introduced for efficient run time translations [18]. This method suits queries without cycles and a subset of SPARQL language [18].

2.2 Approaches for data modeling

Data modeling is considered to be the critical task of Biological Data Handling. Some of the open problems in in it are covered in [10,14]. Elmasri et al. [10,14] state that ordering (e.g. DNA sequences), 3D structures of proteins and functional processes (e.g. metabolic pathways) as the main characteristics of biological data. Conventional data representation does not explicitly include these characteristics. However, they are biologically relevant and ideally data representation should include a mechanism to represent these characteristics. [10,14] propose a new enhanced ER (EER) schema, notation to represent the same and give methodology to implement the same in a RDB. Ordered relationships are modeled by extending the relationship concept in two directions 1) allowing related entities to be ordered and 2) allowing the repetitions of a relationship instances. Molecular spatial relationship deals with the representation of 3D structures in conceptual EER modeling. Atoms and amino acids are modeled with molecular spatial relationships and these spatial structures generate the measurement data like bond angles and bond distance. Atom is treated as points and its position is represented with coordinates in space. Process relationships have three basic entities *i.e*. input, output and catalyst. Inputs are used by the process, the outputs are produced by the process and catalysts are needed for the process to work. Biological pathways are examples of process relationship where an output of one reaction becomes the input of another. For example, the output of transcription process, mRNA, serves as an input for the subsequent translation process.

In [10,13] a multilevel EER model for biological processes which incorporates the multilevel concepts and relationships is proposed. [13] highlights biological examples along with their conceptual EER modeling notations to show that multilevel modeling can be effectively used in biomedical domains and introduces the important concept that at different levels of abstraction, data needs to be modeled differently. The method in [13] also introduces various approaches for data source integration namely horizontal and vertical approaches. The advantage of vertical approach over the horizontal approach is that it integrates data sources from different abstraction levels while the horizontal approach facilitates the integration of data source from same level of abstraction.

In RDB systems, data elements are stored in RDB tables and each table contains an entity with primary key and attributes. Two different entities are related through foreign-key relationships between their keys. Such relationships are not formally defined with specific names. So, such relationships cannot be queried upon. In contrast, Semantic web technology uses RDF and the relationship is treated as a first class entity (predicate), referenced by a URI and stored along with subject and object. In RDF, relationships can also be queried (*e.g.* SPARQL query). This means, the graph of persistent RDF nodes contains the full semantic information about the entities and the relationship between them. In RDBs custom programs are needed for each database schema and the programmer must know the relationship between the tables. Likewise, these relationships are specified in the queries. However when data is stored as RDF graphs, general purpose programs can be written without the knowledge of the underlying RDF graphs, and this could provide a general purpose querying interfaces to the underlying RDF graphs.

2.2.1 Systems biology and data modeling

Systems biology studies introduce another dimension, by requiring different search and modeling needs depending on the user. [8] Introduces different Systems Biology standards that are either accepted or in development. E.g. minimum requirements like MIRIAM and MIASE, the description formats like SBML, SBRML used to represent

data and the associated ontologies like SBO, KiSAO and TEDDY are used to integrate different models to have a better understanding of the complete system. [19] highlights the complexity of biological data as one of the major problems along with the scale of data generated NGS and the scope of the experimental investigations with systems biology. It introduces new data integration architecture Addama. An approach to integrate information management supporting the bottom-up systems biology was introduced in [50]. It proposes to build an automated integration system that can automatically capture the experimental data and integrate it with models.

3 Output methods in data handling

Depending on the nature of the application, output methods can widely differ. Many systems provide knowledge extraction for a human or a computer. Such systems provide search/-results interfaces typically based on a query where the results are displayed as output [17]. Many systems provide structured search capabilities. This is achieved by allowing the input keywords to be associated with specific data elements; providing matching conditions like >, <, contains etc. and search the underlying data for specific matching criteria [34].

Search results have various presentation styles that include computer readable formats. For knowledge extraction systems, faceted browsing [39] is a suitable style. It is effective in showing biologically relevant data where the result set can be easily filtered and categorized. BioFacets [36] allows a faceted classification *i.e*. dynamic categorization of biological result set. Faceted interfaces go naturally with semantic query search and retrieval systems and can help modeling the biological data. Often output has inter-related information; i.e. gene-gene interactions and pathways; which demands visualization to effectively display the search results.

Visualization gives insight into the biological process and hidden relationships between data elements. A survey of visualization tools for biological network analysis highlights the pros and cons of each tool [40]. For visualizing the output data Cytoscape, Ondex, PATIKA [40] etc. provide excellent support. Cytoscape can be enhanced by plugin interfaces [45], it supports Semantic web by importing data from triple store through simple text table or XML-RDF, loading and visualizing RDF data as networks and querying the RDF data with SPARQL. It also helps in developing custom Semantic web applications with Jena and Sesame. It can also be used with other tools like statistical programming language R with sna/ igraph package. For GenomeGraphs [12] an add-on package for R was developed for visualization of genomic datasets. Addama [19] also uses R for its dynamic visualization capabilities.

Often visualization systems provide interactive visualization capabilities. Querying the Semantic web with SPARQL may not be easy for a novice who does not know the structure of the ontology. [29] describes a rewriting of SPARQL to allow users to write queries from their perspective (without knowing the structure of the ontology) but it has limitations. A similar approach was described in [6], which introduces a semantic approach to process knowledge in two phases *i.e*. constructing a semantic query from the user input and displaying the semantic result using scalable vector graphics. Here, the results are output as an RDF graph, often with interactivity to navigate the RDF graph. For systems that output data to be fed into other computer systems, communication standards, ontology, data integration and minimal specification languages play an important role.

4 Approaches enabled by cloud

Various computational solutions to large scale biological data handling are explained in [43]; specifically cloud computing and heterogeneous computing. Currently, the quantity and the storage of genomic data is a vital issue. Cloud computing plays a vital role in the management of genome informatics [47]. Large datasets that act as a virtual disk are stored in a cloud. It inspired projects like Galaxy [51] to build tools to easily setup clusters on cloud platforms. Problems of large datasets requiring huge storage space, processing power and network bandwidth are largely mitigated by commercial scale cloud enabled approaches [47]. Data source providers can expose the data for many consumers, who can access only the requested data through service oriented approaches from the cloud. Extension systems can co-exist in local systems with the cloud. It may be noted that analytical toolbox for biological data like Bioconductor [16] and Galaxy [51] provides prebuilt images for the popular commercial cloud platform Amazon Elastic Computing Cloud (EC2), thus, eliminating large scale datasets and complex software setups on a local network.

5 Examples of data handling systems

The study of biological data handling systems yields the following aspects.

-Data is either aggregated or linked to.
-For non-warehoused systems mediator is needed.
-Ad hoc data retrieval methods extract data and information in unintended ways.
-Extendibility in functionality (ability to add new functions to the system by scripts/ programs).
-Expandable data models (Open world system).
-Use of semantic relationships between data elements.
-Technology choices (Web services, REST)
-Use of infrastructure (Cloud)
- Systems Biology requirements
-Use of output methods

Here, we explore a few innovative systems to identify the underlying concepts. Sample systems are meant to demonstrate such concepts; they are not comprehensive.

5.1 BIO2RDF

Bio2rdf project [4] gives the standards for a system to use Semantic web technology to cross-link information sources and expose services to each other. Since many of the existing systems are not enabled with these technologies, current implementation of Bio2rdf also transforms the data

into semantically linked formats, and exposes a semantic query front end. That is, it has a warehouse for demonstration purposes. It asserts that if the Bio2rdf standards are implemented by the systems then warehousing of the data and Bio2rdf project itself are not needed. Bio2rdf tries to create a network of coherent linked data across the life science databases and provides various SPARQL endpoints to query the RDF graphs without locally storing the graph [4]. A user can define a SPARQL query in a query form and it can be sent to the triple store, and the results can be sent back to the user. With this approach it is possible to link different databases containing the RDF data using the federated and distributed SPARQL queries. Bio2rdf successfully integrated 163 million documents from a large number of data sources [4].

5.2 SADI

Semantic Automated Discovery and Integration, SADI is a Semantic Web Service (SWS) framework which integrates the data from various sources [54]. It is seen that the web services create an implicit biological relation between the supplied input and the retrieved output, but SADI links the input with the output with a common base identifier and the services are annotated thereby explicitly describing the semantic relation between them [54]. SADI framework attempts to build a virtual database by extracting RDF triples through web services, the data can be queried by SPARQL. SADI has improves upon BioMoby and SSWAP by having a SWS framework that integrates itself more naturally into the Semantic web [54]. SHARE is a mediator system which enables federated querying where resources are exposed as services using the SADI SWS framework [54]. SADI services are also REST-like; there is only a standard basic set of HTTP methods, *i.e*. GET and POST [54]. A GET operation on a given service returns its semantic description, while a POST initiates service execution and returns the same RDF graph with the annotations created by the service [11].

CardioSHARE [52] is a unique framework for querying distributed data and performing data analysis using Semantic web standards. The SPARQL query engine of CardioSHARE retrieves the required data dynamically from web services [52]. CardioSHARE project is built on the strengths of BioMoby [55] and addresses its weakness by replacing its syntax with Semantic web ontologies [52]. It is a prototype application that accesses SADI services in response to SPARQL queries. It was initially designed for the analysis of clinical data on heart disease but can be extended to integrate any type of biological data [52].

SADI addresses the problem that most of biological data is in "deep web" and enables discovery of new information from it [54]. SADI proposes a scaled-down version of web service usage, especially suited to bioinformatics; and thus improves upon the earlier Web Services implementations like BioMoby and SSWAP [54]. SADI tries to expose analytical services as REST-enabled URLs [15] that can be combined to form analytical workflow pipelines. Thus, SADI supports and enables ad hoc extension of its data models and functionality.

5.3 ADDAMA

A recent article [19] highlights the complexity of biological data as a major problem along with the scale of data generated and scope of the experimental investigations with systems biology. It introduces new data integration architecture Addama which has been developed for systems biology investigations. Addama tries to integrate and extend existing enterprise technologies to enable the rapid development of ad-hoc tools, and to provide a robust and scalable software infrastructure [19]. The ongoing research requires an adaptable system which provides an integration framework for the existing software technologies while addressing the user requirements which include universal access, support of discovery process and adaptation to new technologies and usage [19]. Addama meets all the user requirements and it does it by allowing a combination of both enterprise technologies and organic software development models. It supports scientists in the use of heterogeneous data types and through the development of related visualization and analysis tools. It defines service interfaces to integrate selected technologies with the underlying infrastructure [19].

6 Key findings and recommendations

The objectives of all systems are similar; so, the best aspects of all systems can be combined to yield a better approach. Data warehousing still has better performance and reliability, and acceptance from academia and industry. Relationships between entities are lost when E-R diagrams are converted into database schema [33]. These can be restored by adding tables to store relationships and multiplicity to model RDB tuples as RDF. Each RDB entity can have a reference id, as defined in some specific domain-standard ontology. Thus, RDB can be "semantically enriched".

Warehouse data can achieve data provenance (authentication) by storing information about source and version; this along with conflict resolution methodologies can be used to build automated/semi-automated update cycles. Warehousing systems build custom parsers to convert source data, e.g. for Uniprot data, BioWarehouse [35] has a parser with XMLBeans [3] technology and object to relational mapping (ORM) conforming to the DRY principle [27]. A class/object model generator that can take OWL based data models as input and an ORM toolset that generates semantically-enriched RDB schema is desired.

Database systems tend to be a closed-world system but they present a consistent snapshot of the knowledge. Open world data from heterogeneous sources can be inconsistent. Warehousing can be enhanced to have knowledge discovery (KD) capabilities by providing connectors to open-world systems; *e.g.* SADI allows other SPARQL end points from the open web [53]. Results from such queries can checked for consistency with the standard snapshot version of information. SADI effectively addresses the problem of a researcher having to go to multiple websites [10]. SPARQL gives a system the capability to extend its knowledge store [38], this is highly desirable.

The major disadvantage of distributed querying is performance [38,46,49] which can be mitigated by caching. Extensive research has been performed in the area of caching the SPARQL queries [38, 49]. Here the query result is cached with an idea of reusing the computed results of previously generated queries avoiding the network usage and increasing the robustness of the system by providing a local copy of cached data when the original source data is unavailable [49].

The adoption of Semantic web technologies for data integration needs productivity enhancing tools for programmers. Possibilities for ORM tools to be architecturally enhanced to work with RDF and OWL is referred to in [18].

SPARQL has the potential to be the choice of end user for knowledge management system that uses Semantic web technologies and maintain semantic relationships. More so, if it procures visual query construction methods [6].

Bio2rdf converts the data from other formats into RDF format using RDFizer [21] whereas SADI leaves the data at its original location.

ADDAMA stresses the need for an ad hoc extension of data stores and functionality [19]. Ad-hoc extensions are especially sought if they are easily mastered and are programming language independent.

We argue that in addition to general purpose query capabilities exposed by SPARQL one may build ADDAMA style REST-based data access services into underlying semantic data stores. Addama also provides the process management services layer with REST-like access mechanism and also provides for a coordinating central registry service. Not all ad hoc data inputs from research communities are curated. They are neither sufficiently structured nor formatted to organize them into RDB models. They contain very less details to organize them into RDF-graph, and much less to be mapped to standardized nomenclature systems and ontologies. Such data also can be input into analytical algorithms in addition to well-structured data from well curated public data, Addama supports this use case. ADDAMA uses content repositories in addition to SQL databases for storing ad hoc data inputs. We note that, for any large scale data handling systems to be effective to serve the research community, ADDAMA approach is very important.

Visualization of experimental results and its analysis capability can be provided with programming extensions to large scale systems as illustrated by Addama project. Use of statistical programming language like R [28] is best suited for this purpose.

In SADI where the output is mapped as annotation to the input data structure, it is possible to build pipelines of processes. Also, input and output data structures can be in a common model (RDF graph). Analytical process pipelines are important for biological research to reduce the time taken for knowledge discovery and processing. Further, the addition of Cloud enabled approaches, wherein data source providers can host the datasets in the cloud and the consumers can access only the needed subset of the data through service oriented methods, can solve many problems related to the scale of biological data and also make the systems reusable thereby reducing the duplication of work. This is our key learning from Galaxy [51].

Our general recommendations are stated here. For future systems, facetted UI is the best choice for visualizing the output. Use of semantic web technologies (controlled vocabularies, ontologies and RDF) is highly desirable. Cloud computing overcomes the issues of huge local repository and outdated data. With proper design, federated approaches can be adapted with minimal deterioration in the data availability and system performance. Service oriented approach, with use of REST is important for large-scale data integration.

7 Acknowledgment

We acknowledge Schools of Biotechnology and Engineering (CS dept.) of Amrita University for the support to conduct the research. We also thank Dr. T.C. Gilliam, Dr. N. Maltsev's team (D.S, S.B., E.B., R.K., B.Q.) and U. Dave of University of Chicago.

8 References

[1] JF Aldana, M Roldán-Castro, I Navas-Delgado, MM Roldán-García, M Hidalgo-Conde, O Trelles. Bio-Broker: a tool for integration of biological data sources and data analysis tools. *Software: Practice and Expereince*, 36(14):1585-1604, 2006.

[2] E Antezana, M Kuiper, V Mironov. Bio. knowledge management: the emerging role of the Semantic Web tech. *Brief Bioinform*, **10**(4)**:**392-407, 2009.

[3] Apache XML Project. Java XMLBeans, 2003. Available from http://xmlbeans.apache.org.

[4] F Belleau, MA Nolin, N Tourigny, P Rigault, J Morissette. Bio2RDF: towards a mashup to build bioinformatics knowledge systems, *J Biomed Inform*. 41(5):706-16, 2008.

[5] VY Bichutskiy, R Colman, RK Brachmann, RH Lathrop. Heterogeneous Biomedical Database Integration Using a Hybrid Strategy: A p53 Cancer Research Database, *Cancer Inform*, 2: 277-87, *2007.*

[6] TD Cao. Integrating a Graphical Semantic Query Interface and a SVG-based knowledge presentation method in an Enterprise Knowledge Management System. *Proc. of AUN/SEED-Net Regional Workshop in Information and Communication Technology*, 2009.

[7] G Carole and S Robert. State of the nation in data integration for bioinformatics, *Journal of Biomedical Informatics*, 41(5), Pages 687-693, October 2008.

[8] VL Chelliah, N Endler, J C Laibe, C Li, N Rodriguez, N Le Novere, Data Integration and Semantic Enrichment of Systems Biology Models and Simulations, *LNCS*, V.5647:5-15, Jul 2009.

[9] H Chen,Y Wang, H Wang, Y Mao, J Tang, C Zhou, A Yin, Z Wu. Towards a semantic web of relational databases: a practical semantic toolkit and an in-use case from traditional chinese medicine. *LNCS*, V.4273:750-763, Nov 2006.

[10] J Chen, S Amandeep: Biological database modeling, *Artech House*, ISBN 13: 978-1-59693-258-6, 2008.

[11] LL Chepelev and M Dumontier. Semantic Web integration of Cheminformatics resources with the SADI framework, *J Cheminform.* 3:16, 2011.

[12] S Durinck, J Bullard, PT Spellman, S Dudoit**.** GenomeGraphs: integrated genomic data visualization with R. *BMC Bioinformatics*, 10**:**Article 2, 2009.

[13] R Elmasri, F Ji, J Fu, Y Zhang, & Z Raja: Modeling concepts and database implementation techniques for complex biological data, *Int. J. Bioinformatics Research and Application*, 3(2):366-388, 2007.

[14] R Elmasri, J Fu, and J Feng. Multi-level conceptual modeling for biomedical data and ontologies integration, *CBMS*, pp.589–594, 2007.

[15] RT Fielding. Representational state transfer (REST), *Ph.D. Thesis*, University of California, Irvine, CA, 2000.

[16] RC Gentleman, VJ Carey, DM Bates, B Bolstad, M Dettling, S Dudoit, B Ellis, L Gautier, Y Ge, J Gentry, K Hornik, T Hothorn, W Huber, S Iacus, R Irizarry, F Leisch, C Li, M Maechler, AJ Rossini, G Sawitzki, C Smith, G Smyth, L Tierney, JY Yang, and J Zhang. Bioconductor: open software development for comp. biology and bioinformatics, *Genome Biol.* 5(10): R80, 2004.

[17] L Guo, J Shanmugasundaram, G Yona. Topology Search over Biological Databases, *ICDE*, 2007.

[18] J Hao, J Liwei, X Zhuoming. Upgrading the Relational Database to the Semantic Web with Hibernate, *Intl. Conf. on Web Information Systems and Mining*, 227-230, 2009.

[19] R Hector, K Sarah, S Ilya and B John. An Integration Architecture Designed to Deal with the Issues of Biological Scope, Scale and Complexity, *LNCS*, V.6254:179-191, 2010.

[20] T Hernandez, S Kambhampati. Integration of biological sources: current systems and challenges ahead, SIGMODRec., 33:51–60, 2004.

[21] http://simile.mit.edu/wiki/RDFizers

[22] http://www.w3.org/2001/sw/

[23] http://www.w3.org/RDF/

[24] http://www.w3.org/TR/owl-features/

[25] http://www.w3.org/TR/rdf-schema/

[26] http://www.w3.org/TR/rdf-sparql-query/

[27] A Hunt, D Thomas. "Don't Repeat Yourself," The Pragmatic Programmer, Addison Wesley, Boston, 2000.

[28] CDR Ihaka, R Gentleman. R: a language for data analysis and graphics, *J. Comput. Graph. Stat*., 5(3), 299–314, 1996.

[29] P Jain, P Yeh, K Verma, C Henson, A Sheth. SPARQL query re-writing for spatial datasets using Partonomy based transformation rules, *Proc. of the Third Intl. Conf. on GeoSpatial Semantics*, 140–158, Springer, 2009.

[30] T Kirsten, HH Do, C Körner, E Rahm. Hybrid integration of molecular biological annotation data, *Proc. Intl. Workshop on Data Integration in the Life Sciences*, 2005.

[31] J Kohler, J Baumbach, J Taubert, M Specht, A Skusa, A Rueegg, C Rawlings, P Verier , S Philippi. Graph-based analysis and visualization of experimental results with Ondex, *Bioinformatics* 22:1383–1390, 2006.

[32] J Kohler, S Philippi, M Lange. SEMEDA: ontology based semantic integration of biological databases, *Bioinf.* 19(18):2420–2427, Dec 2003.

[33] M Krishna. Retaining Semantics in Relational Databases by Mapping them to RDF, IAT Workshops 2006.

[34] M Latendresse, PD Karp. An advanced web query interface for biological databases, *Database*, 2010 doi:10.1093/database/baq006.

[35] TJ Lee , Y Pouliot, V Wagner, P Gupta, DW Stringer-Calvert, JD Tenenbaum, PD Karp. BioWarehouse: a bioinformatics database warehouse toolkit, *BMC Bioinformatics*, 7:170, 2006.

[36] M Mahoui, ZB Miled, A Godse, H Kulkarni, N Li. BioFacets: Faceted Classification for Biological Information, *18th Intl. Conf. SSDBM*, 225–234, 2006.

[37] K Marrakchi, A Briache, A Kerzazi et al.. A Data Warehouse Approach to Semantic Integration of Pseudomonas Data, *LNCS*, V.6254:90–105, 2010.

[38] M Martin, J Unbehauen, S Auer. Improving the Performance of Semantic Web Applications with SPARQL Query Caching, *LNCS*, V.6089:304-318, 2010.

[39] M Norman, H David, H Lynette et al.. Data shopping in an open marketplace: Introducing the Ontogrator web application for marking up data using ontologies and browsing using facets, *Stand Genomic Sci*. ; 4(2): 286–292, Apr 2011.

[40] G Pavlopoulos, AL Wegener, R Schneider. A survey of visualization tools for biological network analysis, *BioData Min*, 1:12, 2008.

[41] C Perez de Laborda, S Conrad. Bringing relational data into the semantic web using SPARQL and relational OWL, *Proc. ICDE* 55–60, *2006.*

[42] L Raschid. Data Modeling and Data Management for the Biological Enterprise, *OMICS:* 7(1):51-55, Jan 2003.

[43] E E Schadt, MD Linderman, J Sorenson, L Lee, GP Nolan. Computational solutions to large-scale data management and analysis**,** *Nat. Rev. Genet*., **11:**647-657, 2010.

[44] SP Shah, Y Huang, T Xu, MMS Yuen, J Lin, BFF Ouellette. Atlas–a data warehouse for integrative bioinformatics, *BMC Bioinformatic*s, 6:34, 2005

[45] P Shannon , A Markiel, O Ozier, NS Baliga, JT Wang, D Ramage, N Amin, B Schwikowski, T Ideker. Cytoscape: a software environment for integrated models of biomolecular interaction networks, *Genome Research* 13(11):2498-504, Nov 2003.

[46] L D Stein. Integrating biological databases, *Nature Rev. Genet.* 4:337–345, 2003.

[47] L D Stein. The case for cloud computing in genome informatics, *Genome Biology,* 11:207, 2010.

[48] R Stevens, P Baker, S Bechhofer, G Ng, A Jacoby, NWPaton, CA Goble, A Brass. TAMBIS: transparent access to multiple bioinformatics information sources, *Bioinformatics* 16(2):184-186, 2000.

[49] H Stuckenschmidt. Similarity-Based Query Caching, *LNCS*, V.3055:295-306, 2004.

[50] N Swainston, DJameson, P Li, I Spasic, P Mendes, N Paton. Integrative Information Management for Systems Biology, *LNCS*,V.6254:164-178, 2010.

[51] J Taylor , I Schenck , D Blankenberg, A Nekrutenko. Using Galaxy to perform large-scale interactive data analyses, *Curr. Protoc. Bioinformatics*, 10.5, 2007.

[52] BP Vandervalk, L McCarthy, M Wilkinson. CardioSHARE: Web Services for the Semantic Web, *Semantic Web Challenge*, 2008.

[53] BP Vandervalk, L McCarthy, M Wilkinson. SHARE: A Semantic Web Query Engine for Bioinformatics, *ISWC* pp.367-369, 2009.

[54] MD Wilkinson, BP Vandervalk, L McCarthy. SADI Semantic Web Services — 'cause you can't always GET what you want!, *IEEE Asia-Pacific Services Computing Conference*, pp.13-18, 2009.

[55] MD Wilkinson, M Links. BioMOBY: an open source biological web services proposal, *Brief. In Bioinform.*, 3(4):331–341, 2002.