

UBIC² — TOWARDS UBIQUITOUS BIO-INFORMATION COMPUTING: DATA PROTOCOLS, MIDDLEWARE, AND WEB SERVICES FOR HETEROGENEOUS BIOLOGICAL INFORMATION INTEGRATION AND RETRIEVAL

PENGYU HONG*, SHENG ZHONG[†] and WING H. WONG[‡]

*BioX Program, Department of Statistics,
Stanford University, Stanford, CA 94305-4065, USA*

**pengyuhong@stanford.edu*

†zhong@stanford.edu

‡whwong@stanford.edu

The Ubiquitous Bio-Information Computing (UBIC²) project aims to disseminate protocols and software packages to facilitate the development of heterogeneous bio-information computing units that are interoperable and may run distributedly. UBIC² specifies biological data in XML formats and queries data using XQuery. The UBIC² programming library provides interfaces for integrating, retrieving, and manipulating heterogeneous biological data. Interoperability is achieved via Simple Object Access Protocol (SOAP) based web services. The documents and software packages of UBIC² are available at <http://www.ubic2.org>.

Keywords: Biological data integration; middleware; interoperability; ubiquitous computing.

1. Introduction

The emergence of various high throughput biological experimental techniques has enabled researchers to monitor the activities of cellular molecules at system levels. Diverse biological data (e.g., sequence data, gene expression data, phenotype data, protein interaction data, etc.) is being generated distributedly at explosive rates. At the same time, numerous bio-information computing methods and applications are being developed daily. It has now been widely realized that deeper and more comprehensive biological knowledge can be discovered faster only by systematically utilizing heterogeneous bio-information and computational utilities. It is expected that future bio-information computing could be pervasive in our daily life (e.g., Personal Health Care System) and complex procedures of biological experiments, which generate inter-related heterogeneous data and require timely analyses to

[‡]To whom correspondence should be addressed.

speed up the whole procedures. To this end, we propose to develop ubiquitous bio-information computing units that are capable of selectively integrating and computing heterogeneous bio-information and are interoperable.

Currently, there are substantial barriers for ubiquitous bio-information computing. First, data is widely distributed. There are more than 700 databases related to molecular biology [10]. Second, most databases are mainly designed for human users and require manual interaction with their websites. The information retrieving procedure is non-programmable or requires non-trivial programming. Third, there has been a great diversity of data specification conventions, which are often not semantically sound and may be changed from time to time. Fourth, although there have been existing efforts for developing and sharing bio-information computation programming libraries, such undertakings are often passive without coordinating with the development of data specification protocols.

To speed up the evolution of ubiquitous bio-information computing, an integrated endeavor should be devoted to developing protocols for semantic data specification, software middleware, and interoperable computing units. In the rest of the paper, we first review related works. Then we introduce the concept of UBIC² and report the current status of UBIC².

2. Backgrounds

2.1. Data specifications

XML (the eXtensible Markup Language) is playing an increasingly important role in promoting the evolution and interoperability of the Internet. XML provides a mechanism to describe and exchange structure-rich data in computer understandable ways. Recently, the biological database research community began formatting biological data using XML DTDs (document-type definitions) or XML Schemas. Outstanding achievements include Distributed Annotation System (DAS) [8] for decentralized sequence annotation, XEMBL [39] for distributing EMBL data, RNAML [40] for exchanging RNA information, Systems Biology Markup Language [26] for representing biochemical reaction networks, and Microarray and Gene Expression Markup Language for microarray data [35]. Such efforts often excel in formatting a certain type of data and achieve little for integrating heterogeneous data and developing software middleware.

2.2. Data integration

Works on database middleware for large-scale integration of heterogeneous biological data mainly take the following two approaches: database federation and data warehousing. A federation approach builds a middle layer on top of a collection of distributed databases and makes distributed databases as an integrated one to user. It allows users to access up-to-date distributed data at run-time and requires only very small local space for storing information. On the other hand, the performance of distributed data retrieval at run-time depends mainly on the reliability

and speed of the underlying network connectivity, which often turns out to be the bottleneck. Examples of the federation approach include BioKleisli [6], TINet [9], DiscoveryLink [11], SEMEDA [30], etc.

A data warehousing approach periodically downloads data in batches from remote databases, then extracts, rearranges, and manages data locally. This approach is more efficient for repeatedly retrieving a large amount of heterogeneous data. Usually, different data fetching programs should be written to serve different data sources. However, most of the programs share a similar operation procedure. An inheritable and expandable template can be designed and implemented to reduce the cost of programming. Examples of warehousing systems include AnnBuilder [42], BioMolQuest [4], EnsMart [28], InterPro [31], JXP4BIGI [27], SLAD [34], SRS [41], SOURCE [7], TAMBIS [32], etc.

2.3. Software middleware

Most biological data integration research projects provide limited software middleware support for developing bioinformatics applications. A desired software middleware should at least provide basic classes (e.g., list, vector, hash table, etc.) for storing and manipulating bio-information elements (e.g., gene annotation, biological sequence, etc.). Existing open-source programming toolkits including BioPerl [36], BioPython [5], BioJava [15], BioRuby [17], and BioConductor [14]. These projects aim to facilitate the development of stand-alone applications.

2.4. Interoperability

Stein advocated web services as a solution to achieve interoperability among online biological databases [37]. Web services provide interfaces which are described in a machine-processable format as standard means of interoperating between different databases and computation utilities regardless of their platforms and programming languages. Applications can be developed to weave together web services from a variety of sources to create a distributed application. The BioMOBY project [16], the caCORE project [18], and the myGrid project [19] are three successful projects sharing a similar approach to achieve this goal. They define a set of schemas and each provides a centralized registry of data and services, which are decentralized and can be accessed programmatically. The myGrid project emphasizes on developing high-level middleware to support bioinformatics research on a Grid. Objects in both caCORE and BioMOBY can be serialized into XML streams, which can be exchanged via SOAP web services or HTTP-XML interfaces. The BioMOBY objects are data-only elements and the caCORE objects have rich functions in addition to data.

3. UBIC²

The UBIC² project advocates developing UBIC² units (see Fig. 1), which can be a data consumer, a service provider, or the combination of two. As a data consumer,

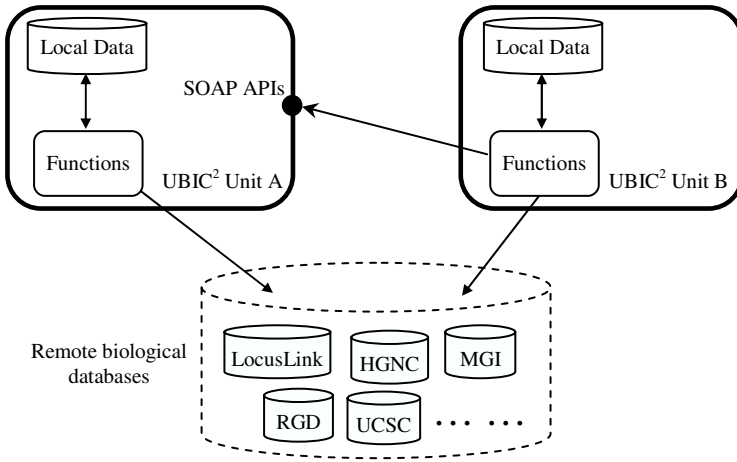


Fig. 1. The UBIC² Concept. Unit A is a data consumer and a service provider. Unit B is a data consumer.

a UBIC² unit is capable of integrating heterogeneous data from selected sources via a mixture of federation and warehousing at run time. Some frequently used remote databases are warehoused. Other databases, which are used occasionally or too large to be managed locally, are integrated via federation. UBIC² units manage data in XML formats. UBIC² adopts several existing XML data specifications and initiates new ones when needed. As a service provider, a UBIC² unit uses the Web Services Description Language to define its data-oriented or procedure-oriented services, which are accessible on the Internet as SOAP web services and can be queried. Different from BioMOBY, caCORE, and myGrid, UBIC² does not have a centralized registration but do require standardizing data specifications. The UBIC² programming library is designed and implemented using object-oriented technology. The library contains a set of inheritable generic classes, which are designed for biological objects sharing similar or common structures and functions.

3.1. Protocols for data specifications and query

Ideally, data specifications can be unified by major biological databases. Many data sources have specified data in XML formats, such as DAS XML specification [8], XEMBL [39], NCBI Document Type Definitions [20], and Swiss-Prot XML Schema [24]. However, there are still many databases formatting data in semantically unsound ways. The UBIC² XML data specification has four categories: Molecule annotation information, Sequence, Ontology, and Literature. Molecule information schemas cover annotation information of genes and proteins. The sequence schemas define the formats for representing nucleotide sequences and amino acid sequences. We designed XML schemas for gene annotation information (including name, symbol, synonyms, products, chromosome locations, Gene Ontology annotation,

```

<UBIC2_GeneInfoQuery>
{
  for $gene in doc('GeneInfo.xml')/Gene
    where $gene/LocusLinkID = "1" or $gene/LocusLinkID = "16999"
  return
    <Gene>
      { $gene/LocusLinkID }
      { $gene/Name }
      { $gene/Symbol }
      { $gene/UniGeneID }
      { $gene/RelatedPapers }
    </Gene>
}
</UBIC2_GeneInfoQuery>

```

```

<GeneInfo>
  <Gene>
    <LocusLinkID>1</LocusLinkID>
    <Name>alpha-1-B glycoprotein</Name>
    <Symbol>A1BG</Symbol>
    <UniGeneID>Hs.390608</UniGeneID>
    <RelatedPapers>
      <PubMedID>2591067</PubMedID>
      <PubMedID>3458201</PubMedID>
      <PubMedID>8889549</PubMedID>
      <PubMedID>12477932</PubMedID>
    </RelatedPapers>
  </Gene>
  <Gene>
    <LocusLinkID>16999</LocusLinkID>
    <Name>latent transforming growth factor beta binding protein 4</Name>
    <Symbol>Ltbp4</Symbol>
    <UniGeneID>Mm.272251</UniGeneID>
    <RelatedPapers>
      <PubMedID>10349636</PubMedID>
      <PubMedID>11042159</PubMedID>
      <PubMedID>11076861</PubMedID>
      <PubMedID>11217851</PubMedID>
      <PubMedID>12208849</PubMedID>
      <PubMedID>12477932</PubMedID>
    </RelatedPapers>
  </Gene>
</GeneInfo>

```

Fig. 2. A query example. Top: query. Bottom: results.

literature links, etc.), gene homology information, and gene upstream sequences. The Ontology category only includes specification of Gene Ontology [1]. We adopted Swiss-Prot XML Schema [24] for protein information and PubMed DTD [21] for literature data.

As for information query, UBIC² uses a query language XQuery [25], which is designed to be broadly applicable to XML data sources.^a This is one of the most significant differences between UBIC² and other projects (e.g., BioMOBY,

^aAlthough the World Wide Web Consortium has not finalized the recommendation of XQuery, UBIC² can easily accommodate future updates of XQuery.

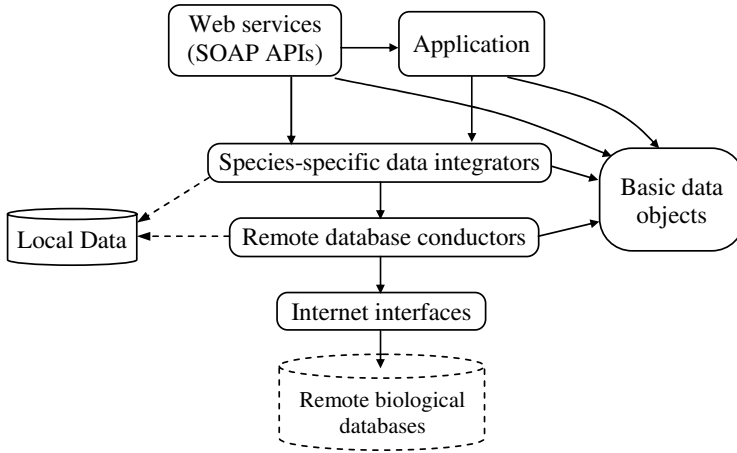


Fig. 3. The UBIc² software architecture.

caCORE, myGrid, etc.). Query results are returned in XML formats defined by UBIc² XML data specifications. Figure 2 shows a query example, which use LocusLink IDs to retrieve the following information of two genes: official names, official symbols, UniGene IDs, and PubMed IDs of papers related to the genes.

3.2. Software architecture

The UBIc² software architecture contains six components: (1) Internet interfaces, (2) remote database conductors, (3) species-specific data integrators, (4) basic data structures, (5) applications, and (6) web services. Their relations are illustrated in Fig. 3. The components and their relations are explained as follows.

3.2.1. Internet interfaces

The Internet interfaces provide programmable means for retrieving data from remote databases. Currently, two network protocols, HTTP and FTP, are supported. Other network protocols can be easily accommodated. If the remote database provides support, a program can choose to download a chunk of data instead of the whole data file using the HTTP protocol. This download method requires a network protocol, an Internet address (e.g., IP address, URI, etc.), and/or the name of a remote data file. The internet interfaces are capable of recovering broken download sessions.

3.2.2. Remote database conductors

Each remote database conductor deals with one remote biological database. Remote database conductors are derived from a generic class, which contains common member variables and functions (e.g., download data, reformat data in XML, index XML data, retrieve XML data, etc.). To make information retrieval efficient, the XML

data files are indexed by major IDs (e.g., LocusLink ID, RefSeq ID, UniGene ID, etc.) at the level of genes/proteins. To retrieve information about genes/proteins, a conductor first uses the indexing information to locate the positions of the corresponding information segments in files, and then load and hash the information segments. To date, the UBIC² software package supports a number of major biological databases including LocusLink [23], HUGO Gene Nomenclature Committee (HGNC) [33], Mouse Genome Database (MGD) [2], Rat Genome Database (RGD) [38], Swiss-Prot [3], HomoloGene [22], and sequence databases from UCSC Genome Bioinformatics [29], and so on.

3.2.3. Species-specific data integrators

Different from BioMOBY, caCORE, and myGrid, UBIC² has species-specific data integrators. Since remote biological databases are developed and maintained cooperatively as well as independently, the information in those databases could be complementary, overlapping, and inconsistent. For example, both LocusLink and HGNC contain annotations for human genes, however with different range of coverage. To integrate those data, we designed an integrator for each species. A species-specific data integrator invokes appropriate remote database conductors to obtain various information of one species from different databases. It then combines the collected information. This layer makes miscellaneous distributed data of a species as coming from one unified source to users. We have implemented integrators for three organisms: *Homo sapiens*, *Mus musculus* and *Rattus norvegicus*. The integrators are inherited from a generic class, which can be derived to support other species.

Currently, the information is categorized into four types: gene annotation, protein annotation, gene homology annotation, and upstream sequence. Gene annotation include the following information of a gene: LocusLink ID, RefSeq ID, UniGene ID, GenBank ID, Swiss-Prot ID, Gene Ontology annotation, official name, official symbol, synonyms, and PubMed IDs of related literature. Protein annotation includes: Swiss-Prot ID, protein name, EMBL IDs, HSSP IDs, InterPro IDs, Pfam IDs, PIR IDs, PRINTS IDs, ProDom IDs, PROSITE IDs, SMART IDs, and TRANSFAC IDs. Each type of information is stored in a species-specific XML data file, which is indexed by major IDs (e.g., LocusLink ID, Swiss-Prot ID, UniGene ID, etc.).

In addition to information integration, the integrators also detect data inconsistency across databases, reconcile data discrepancies, and track obsolete information. The detected inconsistency is stored and reported to users. Currently, the integrators invoke a simple method to automatically reconcile such discrepancies. This method uses a designated database as the correct information source. Developers can replace this method by simply overriding the discrepancy reconciliation function of an integrator based on their own experience with a remote database.

Integrators also handle species-specific query of heterogeneous data. When a query is received, an integrator first tries to retrieve information from local species-

specific XML data files. If the required information cannot be found locally, it will invoke appropriate remote database conductors to retrieve information from remote databases. Online query of remote databases is useful when a remote database (e.g., PubMed) is too big for a local machine. The following case exemplifies how integrators retrieve information. Given a set of LocusLink IDs, the user wants to retrieve related gene annotations, upstream sequences, protein annotations, and PubMed abstracts. The integrators first retrieve gene annotation information, which contains RefSeq IDs, Swiss-Prot IDs of genes' protein products, and PubMed IDs of the related papers. Then, the integrators retrieve upstream sequences from upstream sequence files using RefSeq IDs, protein annotations from protein annotation files using Swiss-Prot IDs, PubMed abstracts from PubMed database via the PubMed conductor using PubMed IDs.

3.2.4. *Basic data objects and applications*

The basic data types of UBIC² include gene annotation, protein annotation, biological sequence, gene expression, etc. Basic data objects, e.g., lists, vectors, hash-tables, etc., are designed and implemented for each data type. Remote database conductors and species-specific data integrators use basic data objects to manage information in computer memory. Built on top of basic data objects and species-specific data integrators, UBIC² applications can retrieve local information and selectively integrate remote data at run time.

3.2.5. *Web services*

The web services of UBIC² units are described in the Web Services Description Language and hence are searchable on the Internet. A client invokes UBIC² Web services by using SOAP-messages, which are typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards. Upon receiving a SOAP message, a UBIC² unit can use species-specific data integrators to access local and remote data or use TCP/IP to communicate with UBIC² applications, which run on the same server and provide complicated computational functions. Data retrieval services can be unified because UBIC² standardizes data specifications and uses XQuery to retrieve data. If a data service provider uses different data specification protocols, a client can still query data from that provider as long as it can transform its data into the data formats of UBIC² using Extensible Stylesheet Language Transformations.

4. UBIC² Version 1.0

The current release of UBIC² (version 1.0) was implemented using Microsoft's C# programming language and .Net technology, which supports FreeBSD OS and Mac OS X 10.2. The UBIC² concept can be implemented using other programming languages and platforms, e.g., Java and Apache/SOAP. UBIC² 1.0 is distributed

with a demo application that can integrate several remote biological databases and perform batch retrievals of heterogeneous biological data. UBIC² 1.0 releases a web service for querying gene annotations [13]. The web service is deployed on a Window Server 2003 installed with Microsoft Internet Information Service 6.0. Documents and manuals of the UBIC² programming library and the demo are available at <http://www.ubic2.org>.

5. Conclusions

The current version of UBIC² is designed to favor individual researchers and small research groups. We have chosen an economical way for maintaining and managing data as local XML files, which do not require a database management system to manage. The UBIC² programming library provides a set of objects and APIs to allow quick development of bioinformatics applications. We have used the library to develop GeneNotes [12], which already has a substantial number of users. A UBIC² unit can provides its data-oriented and procedure-oriented services on the Internet via SOAP-based web services. We hope UBIC² will contribute to solve data comparability and accessibility and computational interoperability in bioinformatics research.

Acknowledgments

The work of Pengyu Hong is supported by NIHGM67250. The work of Wing H. Wong is supported by NIH-HG02341.

References

1. M. C. Ashburner *et al.*, Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium, *Nature Genetics* **25**(1) (2000) 25–29.
2. J. A. Blake *et al.*, MGD: The Mouse Genome Database, *Nucleic Acids Res.* **31**(1) (2003) 193–195.
3. B. Boeckmann *et al.*, The SWISS-PROT protein knowledge base and its supplement TrEMBL in 2003, *Nucleic Acids Res.* **31**(1) (2003) 365–370.
4. Y. V. Bukhman and J. Skolnick, BioMolQuest: Integrated database-based retrieval of protein structural and functional information, *Bioinformatics* **17**(5) (2001) 468–478.
5. B. Chapman and J. Chang, Biopython: Python tools for computation biology, *ACM SIG-BIO Newsletter*, 2000.
6. S. B. Davidson *et al.*, BioKleisli: A digital library for bio-medical researchers, *Int. J. Digital Libraries* **1**(1) (1997) 36–53.
7. M. Diehn *et al.*, SOURCE: A unified genomic resource of functional annotations, ontologies, and gene expression data, *Nucleic Acids Res.* **31**(1) (2003) 219–223.
8. R. D. Dowell *et al.*, The distributed annotation system, *BMC Bioinformatics* **2**(1) (2001) 7.
9. B. A. Eckman *et al.*, Extending traditional query-based integration approaches for functional characterization of post-genomic data, *Bioinformatics* **17**(7) (2001) 587–601.
10. M. Y. Galperin, The molecular biology database collection: 2005 update, *Nucleic Acids Res.* **33** (2005), Database issue D5–D24.

11. L. M. Haas *et al.*, DiscoveryLink: A system for integrated access to life sciences data sources, *IBM Systems Journal* **40**(2) (2001) 489–511.
12. P. Hong and W. H. Wong, GeneNotes: A novel information management software for biologists, *BMC Bioinformatics* (to appear).
13. <http://bayer.fas.harvard.edu/Webservice/BioWebservice/BioWebservice.asmx>
14. <http://www.bioconductor.org/>
15. <http://biojava.org>
16. <http://biomoby.org>
17. <http://bioruby.org/>
18. P. A. Covitz *et al.*, caCORE: A common infrastructure for cancer informatics, *Bioinformatics* **19**(18) (2003) 2404–4412.
19. <http://www.mygrid.org.uk>
20. <http://www.ncbi.nih.gov/dtd/>
21. <http://www.ncbi.nlm.nih.gov/entrez/query/static/PubMed.dtd>
22. <http://www.ncbi.nlm.nih.gov/HomoloGene/>
23. <http://www.ncbi.nlm.nih.gov/LocusLink/>
24. <http://www.ebi.ac.uk/swissprot/SP-ML/>
25. <http://www.w3.org/TR/xquery/>
26. M. Hucka *et al.*, The systems biology markup language (SBML): A medium for representation and exchange of biochemical network models, *Bioinformatics* **19**(4) (2003) 524–531.
27. Y. Huang *et al.*, JXP4BIGI: A generalized, Java XML-based approach for biological information gathering and integration, *Bioinformatics* **19**(18) (2003) 2351–2358.
28. A. Kasprzyk *et al.*, EnsMart: A generic system for fast and flexible access to biological data, *Genome Research* **14**(1) (2004) 160–169.
29. W. J. Kent *et al.*, The human genome browser at UCSC, *Genome Research* **12**(6) (2002) 996–1006.
30. J. Kohler and S. Schulze-Kremer, The semantic metadatabase (SEMEDA): Ontology based integration of federated molecular biological data sources, in *Silico. Biol.* **2**(3) (2002) 219–231.
31. N. J. Mulder *et al.*, InterPro: An integrated documentation resource for protein families, domains and functional sites, *Briefings in Bioinformatics* **3**(3) (2002) 225–235.
32. N. W. Paton *et al.*, Query processing in the TAMBIS bio-informatics source integration system, *Proc. 11th Int. Conf. on Scientific and Statistical Database Management*, IEEE, New York, 1999.
33. S. Povey *et al.*, The HUGO Gene Nomenclature Committee (HGNC), *Hum. Genet.* **109**(6) (2001) 678–680.
34. C. Schonbach, P. Kowalski-Saunders, and V. Brusica, Data warehousing in molecular biology, *Briefings in Bioinformatics* **1**(2) (2000) 190–198.
35. P. T. Spellman *et al.*, Design and implementation of microarray gene expression markup language (MAGE-ML), *Genome Biology* **3**(9) (2002) research0046.1-0046.9.
36. J. E. Stajich *et al.*, The Bioperl toolkit: Perl modules for the life sciences, *Genome Research* **12**(10) (2002) 1611–1618.
37. L. Stein, Creating a bioinformatics nation, *Nature* **417** (2002) 119–120.
38. S. Twigger *et al.*, Rat Genome Database (RGD): Mapping disease onto the genome, *Nucleic Acids Research* **30**(1) (2002) 125–128.
39. L. Wang *et al.*, XEMBL: Distributing EMBL data in XML format, *Bioinformatics* **18**(8) (2002) 1147–1148.

40. A. Waugh *et al.*, RNAML: A standard syntax for exchanging RNA information, *RNA* **8**(6) (2002) 707–717.
41. E. M. Zdobnov *et al.*, The EBI SRS server-new features, *Bioinformatics* **18**(8) (2002) 1149–1150.
42. J. Zhang *et al.*, An extensible application for assembling annotation for genomic data, *Bioinformatics* **19**(1) (2003) 155–156.