Available online at www.sciencedirect.com

SCIENCE DIRECT®

Computer Standards & Interfaces 27 (2005) 579–596

COMPUTER STANDARDS & INTERFACES

www.elsevier.com/locate/csi

The Federated Forums based Integrated Biological Data Mining Broker

Min-Huang Ho[a,*], Yue-Shan Chang[b], Kuang-Lee Li[c], Shyan-Ming Yuan[c]

[a]*Department of Information Management, Leader University, Taiwan*
[b]*Department of Computer Science and Information Engineering, National Taipei University, Taiwan*
[c]*Department of Computer and Information Science, National Chiao-Tung University, 1001 Ta Hsueh Road, Hsin-Chu 30050, Taiwan, ROC*

Received 7 April 2004; received in revised form 22 September 2004; accepted 22 September 2004
Available online 27 October 2004

Abstract

The inherent diversity and complexity of biological data sources make traditional information retrieval technologies unsuitable for handling the problems that arise with the retrieval of biological documents on the Internet. Furthermore, biological data sources in all their variety have specific domains and objectives that differ from other user interface, query formats, result sets and database organizations. To meet the problems of unification in both their syntactic and semantic aspects, we here present a unified, adjustable, and extractable Biological Data Mining-Broker (BDMB). Based on XML technology, the broker provides a federated forum model that handles heterogeneity of sources. It also uses a feedback-based utility for raw and meaningful extracted cache techniques that makes the system more efficient and accurate. The experimental results show that the system performs well and is ideal for biological data mining processes with many different data sources, mining applications, and knowledge analysts. It is very useful for target discovery and bioinformatics research projects.

Keywords: Information retrieval; Biological informatics; Data mining; XML

1. Introduction

1.1. Background and motivation

The past decade has seen an explosive development in biology-related research with much of it being focused on biological data mining [1–8]. Some researchers have made multiple heterogeneous data sources accessible for biological data mining processes [9–11]. However, the formats and structures of bio-

* Corresponding author. Tel.: +886 921222768; fax: +886 62553129.
E-mail addresses: mhho@mail.leader.edu.tw (M.-H. Ho), ysc@mail.ntpu.edu.tw (Y.-S. Chang).

0920-5489/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.csi.2004.09.011

logical data mining queries and result sets differ greatly in relation to data sources. Most biological data mining query languages and processes have complex structures and knowledge analysts with their different preferences have found it hard to agree on formulations. Using traditional mechanisms that were invented for general-purpose meta-search techniques presents great difficulty when the need is for the efficient integration of different biological data sources.

Currently, the major problems in source retrieval are:

- *Poor integration*: many traditional researches do not provide unified full-function query mechanisms, or those provided are too difficult to learn. Nor are the result sets derived from the sources easy to convert to a unified format.
- *Poor performance*: many proposed systems are very inefficient in handling highly diverse documents. That will consequently bring about meaningless browsing from existing biological data sources.
- *Poor flexibility and extensibility*: the major defect of existing systems is that they are not flexible or extensible enough to handle query or retrieval processes. They are hard-coded, and do not provide the user with a useful mechanism for extending new data sources.

The aim of the present study is to design and implement a biological mining-broker for retrieving multiple heterogeneous mining applications, including features relating to biological data mining processes, by the following means:

- User interface unification to support the capabilities of transparency.
- XML technology and the integration both of syntactic and semantic heterogeneity.
- Two-phase and feedback-based cache mechanisms to improve the efficiency and accuracy of the data-mining process.
- Federated forums to support the functionalities of flexibility, interactivity, and adjustability.

1.2. Related works

Because of the importance to scientists of unified interfaces for biological information retrieval, much research has been dedicated to related issues. Target Informatics Net (TINet), for instance, uses federated technology to query multiple heterogeneous data sources as if they were components of a single large database [1]. TINet System Middleware, which is based on Gene Logic's Object Protocol Model (OPM) system, supports many and diverse formats of data sources, including Mouse Genome Database (MGD), SwissProt, GenBank, PROSITE, GeneCards, PubMed, as well as on-the-fly BLAST searches. It also provides a preferable strategy when very fast access at runtime is required. By means of the OPM multi-database middleware system with the CORBA services that underlie TINet, new data sources and class methods may be added dynamically to the system with reliability and ease by multiple developers working in parallel.

TINet performs the postprocessing of subquery results to compensate for the differences in the query facilities supported by different data sources, the heavy load on TINet causes a loss of special query result bonuses from different independent data sources. There are strategies for ignoring semantic heterogeneity and keeping copies of data sources of interest in local caches that follow a simple, instinctive, and syntactic cache mechanism. These are inefficient and meaningless for many biological data mining techniques, such as similarity searches and pattern analysis, but most biological data mining applications and knowledge analysts take semantics and meaningfulness into serious consideration.

Transparent Access to Multiple Biological Information Sources (TAMBIS) proposes primarily semantic heterogeneity. It provides a view of heterogeneous biological data sources by means of TAMBIS Ontology (TaO) [4], which is a kind of ontology of data mining for biological terminology based on a specific description logic language and is one of the major functionalities of TAMBIS. TaO ensures an ambitious undertaking and guarantees that if any biological data source is fully integrated into the TAMBIS system, TaO permits sophisticated reasoning about biological concepts. The critical bottleneck of TAMBIS is that it requests data sources and mining applications to fully integrate with it, which requires considerable effort.

Among other related approaches are: (1) The SRS [17] system, which only supports flat file data sources; (2) The Discovery Link form IBM, which

is an SQL-based heterogeneous database integration system based on the Garlic research prototype; (3) The DB2/UDB DataJoiner federated database management system for relational databases; (4) The P/FDM system, which is a schema-based mediator, and a functional data model approach that uses DAPLEX as its query language; and (5) The Kleisli transforms and integrates heterogeneous data sources using a complex object data model and CPL. This is a powerful query language inspired by recent works in functional programming languages.

1.3. Our approach

To overcome the stated problems in the data mining process with multiple heterogeneous biological data sources, we propose an XML-based federated mechanism that we term the *Biological Data Mining-Broker* (*BDMB*). It is devoted to coordinating the format and transparency of directory information, formulating biological data mining language, planning the extraction format for mining applications and data sources, and to knowledge acquisition among different knowledge analysts. In addition, different mining applications and multiple heterogeneous biological data sources operate a federated model that underlies the interactive discussion mechanism and enhance the meaningfulness and usefulness of information retrieval from the broker. Finally, we use the de facto and well-known XML object model to reduce coding effort of mining applications and data source retrieval applications for designers [13–15].

The rest of the paper is organized as follows. In Section 2 , we present the basic components of the system and the architecture of the mechanism. Section 3 has a description of the workflow of the system and the various scenarios of data query. In Section 4 , we set out the experiments and results, after which our concluding remarks are given in Section 5.

2. System architecture

2.1. The components and architecture of biological data mining-broker

The *BDMB* is designed to allow knowledge analysts to easily and efficiently retrieve meaningful information from numerous and diverse heterogeneous biological data sources. Its basic components are shown in Fig. 1. A federated model-based mechanism based on the *Federated Repository Center* (*FRC*), it can handle *Biological Data Mining Query Language* (*BDMQL*) via *Mining Interface* (*MInter*) and send a command sheet in an accurate query format to each specified biological database via *Bio-Info Interface* (*BIOIInter*). To expand the new supported biological database, the system developer can also use a *Template Designer Kit* (*TDKit*) to automatically or semiautomatically generate appropriate DTD files for any specified newly added database.

The six basic components are the *BIO-Info Interface* (*BIOIInter*), the *Raw Data Cache* (*RD-Cache*), the *Directory and Extraction Cache* (*D&E-Cache*), the *Federated Repository Center* (*FRC*), the *Template Developer Kit* (*TDKit*), and the *Mining Interface* (*MInter*). The *BIOIInter* provides the interface for communicating between the inner components and each biological data source wrapper. Through the *BIOIInter*, the *broker* sends well-formed queries to specified wrapper sources, and then receives recognizable result sets with both raw and meaningful extracted data from the sources. The fetched raw data is stored in the *RD-Cache*, and the extracted data in the *D&E-Cache* for later use by other components. By

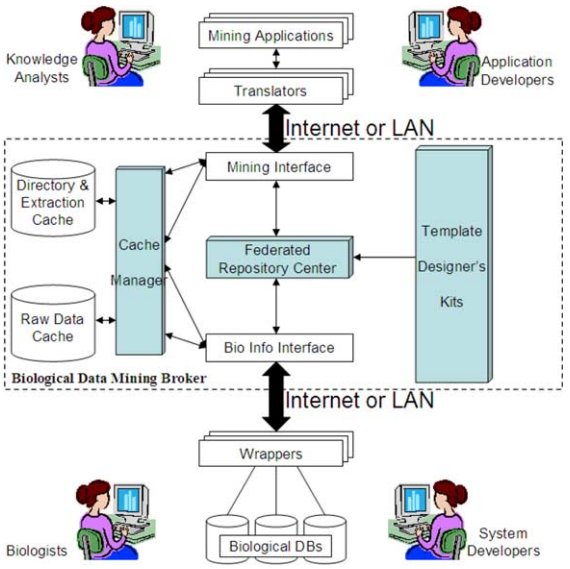


Fig. 1. The system components of our proposed biological mining broker.

applying a rating condition technique in the *D&E-Cache*, the mining application can access meaningful raw data more accurately and efficiently. More detail of the functions and related techniques relating to each of the proposed broker's components is given in the following sections.

2.2. The interfaces of BDMB between the biological data sources and mining applications

To overcome both syntactic and semantic heterogeneity problems, two aspects need consideration. The first is a unified query format and interface, and the other is a uniform data representation of result sets. *MInter* takes care of the first and *BIOIInter* the second.

For mining applications, *MInter* has the most important role in achieving integration of the data mining process. Fig. 2 shows its architecture and that it only serves not only as the interface between the outside mining application and other inner components, but also that it also converts query statements and result sets according to inner federated forums. These include an extraction template, a directory template, a feedback template, and a query template.

MInter is the interface between the biological data mining applications and the mining processing components through which the broker receives and processes structured mining queries, and sends recognizable result sets with meaningful extracted data from the cache databases to the applications. After sending the result sets to the applications, the knowledge analysts must evaluate each data source and rank the rating conditions for all the extracted data with feedback forms via a federated feedback mechanism in the interface. This mechanism influences cache management as to which raw or extracted data is to be cached in databases, and also affects the performance of future mining queries.

Fig. 2 shows the three Minter components, the *Mining Query Processor* (*MQP*), the *Mining Feedback Processor* (*MFP*), and the *Federated Forums* (*FFrm*). The main purpose of *FFrm* is to provide a meaningful, cached-based, unified and transparent interface for *MQP*, which is accessed by the translators of the mining applications.

As mentioned earlier, the XML-based *BDMQL* is acceptable to our proposed broker via *MInter*. Fig. 3 shows a typical simple mining query written by *BDMQL*. It means "use data source of NCBI and GenBank to mine default information featured on lineage with all fields fetched out, based on the nucleotide domain of locus AF323081, and the source should be human".

Although biological mining queries in *BDMB* are well formatted in XML, they contain lots of complicated information. Thus, the *FFrm* of *MInter* is playing an important role. The translator of a mining application can utilize a federated query template in *FFrm* to recognize the information in mining queries. In addition, the *FFrm* also includes a directory template forum, an extraction template forum, and a feedback template forum. These are interfaces constructed to assist communication between *MInter* and

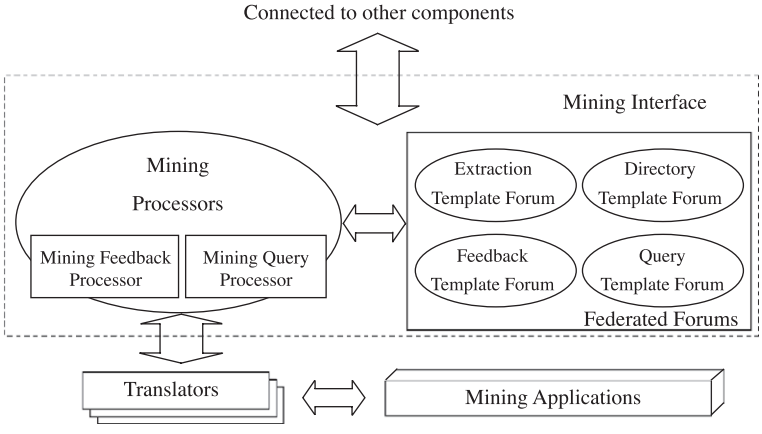


Fig. 2. The architecture of MInter.

```
<?xml version="1.0" ?>
<!--  Biological Data Mining Query Example  -->
- <BDMQL-entry>
  - <DS-list>
    - <DS>
        <Name>NCBI</Name>
      </DS>
    - <DS>
        <Name>GenBank</Name>
      </DS>
    + <Weight-list>
    </DS-list>
  - <Body>
    - <Mining-list>
        <Mining>Default</Mining>
      </Mining-list>
    - <Selection-list>
        <Selection>H.*</Selection>
      </Selection-list>
    - <Source-list>
      - <Source>
          <Name>Human</Name>
          <Alias>H</Alias>
        </Source>
        <Search-expr />
        <Exclusion-expr />
      </Source-list>
    - <Locus-list>
        <Locus>AF323081</Locus>
        <Search-expr />
        <Exclusion-expr />
      </Locus-list>
    - <Domain-list>
        <Domain>Nucleotide</Domain>
        <Search-expr />
        <Exclusion-expr />
      </Domain-list>
    - <Feature-list>
        <Domain>Lineage</Domain>
        <Search-expr />
        <Exclusion-expr />
      </Feature-list>
    </Body>
  + <Rating_Condition>
  </BDMQL-entry>
```

Fig. 3. A typical simple BDMQL example.

multiple heterogeneous data source wrappers. Together, the templates describe two major things: the fields contained in the extraction format and the use and meaning of the fields.

As mentioned above, if the document in the cache cannot fulfill a user's request, the broker activates the *Multi-source Query Processor* (*MSQP*)of *BIOIInter* to retrieve the biological information over the network. Fig. 4 shows the architecture of *BIOIInter*. *MSQP* uses *Raw Data Receiver* (*RDR*) to fetch original raw data directly from data sources and *Extracted Data Receiver* (*EDR*) to fetch any meaningful data set by parsing the returned raw data. *FFrm* also plays an important role as a guideline for *MSQP* in querying and parsing heterogeneous data sources. That is, any queries accepted by *BIOIInter* will be verified by the *Query Template Forum* to ensure the correctness of the query commands with wrappers.

After *BIOIInter* received an XML-based query from other components, *MSQP* sent it in proper format to any interested biological data source wrappers, in accordance with *DIR-entry* of the *Multi-Source Directory*. In our design, the wrappers have to fetch and parse the result sets from the data sources, and then send them to the *Extracted Data Receiver* and *Raw Data Receiver*, respectively, for later use. In the *Extracted Data Receiver*, the original returned result is further converted into extracted data result sets with an XML-based perceivable and unified format. Again, *FFrm* is the guideline for the formatting the data sources relating to queries, extractions, and directories. By combining Federated Forums, Template Designer Kits, and wrapper generating technologies [18,19], much of the generation work can be semi- or even fully automated.

2.3. The unified data format with federated repository center

As explained above, federated forums play a very important role in unifying the format of documents across heterogeneous data sources. Our broker has four template forums stored in the *Federated Repository Center* (*FRC*). The *Federated Extraction Template* (*FET*) stores the definition of meaningful extracted data, the *Federated Directory Template* (*FDT*) that of multi-source directory information, the *Federated Query Template* (*FQT*) that of structured mining queries, and the *Federated Feedback Template* (*FFT*) that of predefined mining feedback forms. These templates are all DTD [12] documents and are referenced by *BIOIInter*, *MInter*, and other components.

The broker components use *FET* to check the correctness of queries. After receiving and processing the queries from the broker, the wrappers retrieve the original raw data result sets. Meanwhile, the *MSQP* of *BIOIInter* accesses FET to parse the sets and produces a meaningful extraction in a preferred data format.

The *FDT* provides definitions for the DIR-entry in the multi-source directory. The system developer offers proper DIR-entries in this directory for every heterogeneous data source, thus providing a unified representation for directory information relating to

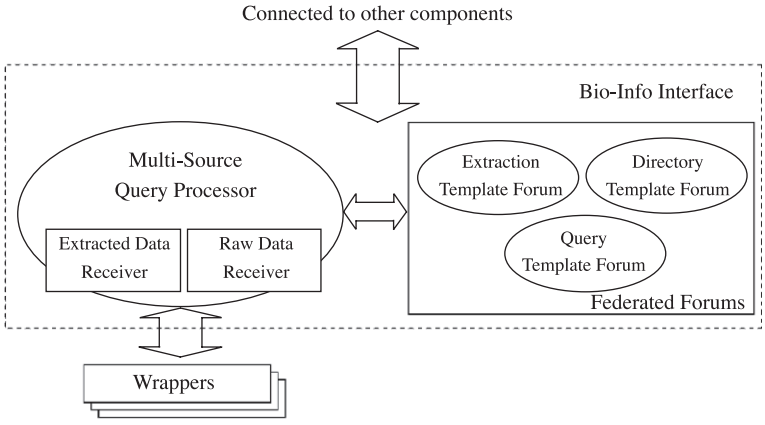


Fig. 4. Architecture of BIOIInter.

biological data sources. Through the FDT, the broker can access directory information relating to all data sources in a uniform way.

FFT provides definitions for well-formed XML-based feedback forms. The mining applications generate validated and readable feedback forms with respect to mining queries for the broker by referring to *FFT*. The broker also requests the *FFT* to recognize the meaning of feedback forms. Thus, after receiving the mining application forms, the broker obtains the related feedback information to tune the rating conditions of the extracted data result sets and the weights of the data sources. This essentially is the task of the semantic cache mechanism in the federated model.

To increase the flexibility of the broker, we also provide *TDKit*, a toolkit for system developers to create and manage mining application templates and newly added biological data sources. The kit includes four useful tools: *Extraction*, *Directory*, *Query*, and *Feedback Template Generator*. Through *TDKit*, data sources and mining applications can be matched to the templates in the broker's federated model and developers can unify the format of documents used in wrappers and applications. This is a flexible way for the user to add new biological information source to the system.

For example, to help specify the location of a data source in *BDMQL* for mining applications, the system developer can use the *Query Template Generator* to add location definition to the *Federated Query Template*. Equally, the *Extraction Template Generator* can be used to generate a proper, customized *Federated Extraction Template* for any heterogeneous data source. Thus, extracted data can be defined to satisfy any mining application requirements or the characteristics of any data source.

2.4. Two-phase caching mechanism

Mining diverse heterogeneous biological data sources is a very time-consuming process. Transforming fetched documents into result sets, for example, can take anything from a few minutes to more than 20 hours [1,8]. We can improve the performance by a two-phase caching mechanism, involving *Data and Extraction Cache* (*D&E-Cache*) and *Raw Data Cache* (*RD-Cache*). The first includes *Multi-Source Directory* (*MSDir*) and *Extraction Cache Databases* (*ECDB*). MSDIR stores directory information about data sources and *ECDB* temporarily stores extracted data in a cache for each source. Unlike *RD-Cache*, *D&E-Cache* is more manageable and meaningful, and always gives higher priority to access by mining applications or other components.

The *Cache Manager* is the interface between caches and the various components. It caches original raw data directly through *BIOIInter* into *RD-Cache*. To retain flexibility, each data source has its own database for caching returned original raw data in its own format.

In the first phase of the mining process, the broker also refers to the rating condition in the *D&E-Cache* and decides which raw data sources to cache. Because no additional processes need be performed on the raw data, the decision is fast and rough but suitable for the first phase of the process. Fig. 5

Locus	Domain	Feature	Keyword	BIO-Sequence
BI018091	Nucleotide	Lineage	M13	GGTACT...
NC001477	Genome	History	Pet	AGTTG...
NP071721	Protein	**NULL**	House	ATGAC...
NC001998	Genome	**NULL**	**NULL**	GGATC...
AF385623	Nucleotide	Human	Africa	ATGAT...
. . . .				

Fig. 5. Raw data table example in RD-Cache database.

shows a portion of the sample raw data cache table in the raw data cache database containing the primary field of biological data, Bio-sequence. All tables are stored in a relational database for indexing purposes.

```
<?xml version="1.0" ?>
<!--  Extracted Data Example  -->
- <Extraction>
  - <Sequence>
      <Data_Source>NCBI</Data_Source>
      <Locus>AF323081</Locus>
      <Title>Homo sapiens resistin mRNA</Title>
      <g.i.>12584201</g.i.>
    - <Molecular-info>
        <Class>mRNA</Class>
        <Value>3</Value>
      </Molecular-info>
    + <Ver-info>
    - <Organism>
        <Taxonomy>Homo sapiens</Taxonomy>
        <Source>Human</Source>
        <Tag>Taxon</Tag>
        <Genus>Homo</Genus>
        <Species>Sapiens</Species>
      + <Lineage-list>
      </Organism>
    - <Seq_Data>
        <Type>Raw</Type>
        <Molecular>RNA</Molecular>
        <Length>476</Length>
        <BIO-Sequence>GTGTG....</BIO-Sequence>
      </Seq_Data>
    + <Domain-list>
    + <Feature-list>
    - <Characteristics>
      - <Range>
          <From>46</From>
          <To>372</To>
        </Range>
      </Characteristics>
    + <Rating_Condition>
    </Sequence>
</Extraction>
```

Fig. 6. A simple extracted data example.

The second phase is *D&E-Cache*. As mentioned earlier, it is composed of *MSDir* and *ECDB*. In *ECDB*, each data source has its own database for storing extracted data in the same XML-based format. Fig. 6 shows a portion of the XML-based extracted data sample in *ECDB*. The element <*Extraction*> . . . </*Extraction*> contains extracted data with <*Sequence*> . . . </*Sequence*>. The second element includes extracted information relating to the respective raw data, the related domains, the special features, the characteristics, and the rating conditions, each represented with a tag.

In addition, the element <*Rating_Condition*> . . . </*Rating_Condition*> plays an important role in the federated model and the feedback-based meaningful cache mechanism of the broker. On one hand, *ECDBs* provide more perceivable and unified result sets for mining applications than those in the original raw data format. On the other hand, with this element, the extracted data contains more useful and meaningful extraction information. Thus, mining applications can access data sets that are more powerful and efficient than the original raw data result sets in *RD-Cache*.

3. System workflow

3.1. Two-phase cache-based biological data mining

According to the caching theory in computer science, by utilizing *BDMB* to mine biological information on the Internet, knowledge analysts find that a two-phase caching mechanism greatly improves performance. Fig. 7 shows the five major steps that the mechanism requires as well as the workflow and system components.

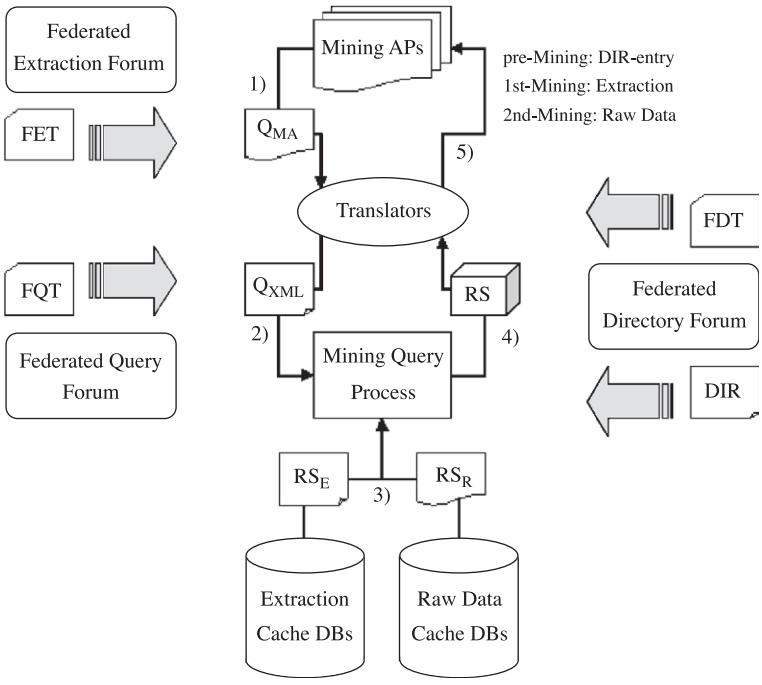


Fig. 7. The five main steps of two-phase cache based biological data mining querying.

(1) Mining applications send queries (Q_{MA}) to their own translators. A simple example of a traditional SQL-based mining query is shown in Fig. 8. It is the same as the example described in Section 2.2.
(2) By means of the federated query forum through *MInter*, mining application translators demand *DTD*-based *FQT* from *FRC*. Then, using *FQT*, they convert the queries into specified formats into *BDMQL* format, Q_{XML}. The queries are than sent through the network to the *Mining Query Process* (*MQP*) of *MInter*. In this example, the Q_{XML} is the same as the contents shown in Fig. 3.
(3) After receiving Q_{XML}, *MPQ* parses it and contacts the *Cache Manager* to perform the query process from both *D&E-Cache* and *RD-Cache*. Similarly, *MQP* invokes *FQT* through the federated query forum, the federated directory template (*FDT*) and the DIR-entry (*DIR*) through federated directory forum to facilitate parsing the query. *MQP* checks DS-list with the weight-list, body and *rating_condition* elements of Q_{XML} and retrieves the respective extracted and raw data result sets (RS_E and RS_R) from the *Cache Manager*.
(4) In this step, *MQP* uses the network sends cache-based result sets (*RS*) and the *DIR*-entry retrieved from the multi-source directory through the federated directory forum to mining application translators. They request the federated extraction template (*FET*) through the federated extraction forum for mining applications and knowledge analysts translate the desired extracted data result sets.

```
use data source NCBI, GenBank
weight with overall >= 80 and domain >= 87.5
mine default
select H.*
source from Human as H
locus for 'AF323081'
domain in Nucleotide
feature on Linage
rate as overall >= 93.5 and feature != 'History'
```

Fig. 8. Simple example of traditional mining query.

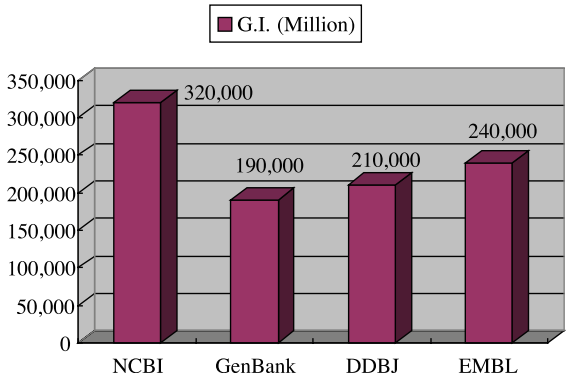


Fig. 9. Simple example of premining with directory information.

(5) Mining applications and knowledge analysts use *DIR-entry* to perform premining with directory information. A simple example of premining with directory information is shown in Fig. 9. Mining applications and knowledge analysts can also evaluate the practicability of cache-based extracted data result sets by first-phase mining in the biological data mining process. First-phase mining can be a semi-automatic preprocessing of the biological data mining process. Fig. 10 shows a simple example of first-phase mining with extracted data result sets [10]. Furthermore, mining applications and knowledge analysts adopt cache-based raw data result sets to perform second phase, or traditional biological data mining processes. If knowledge analysts fail to find these results satisfactory, they can continue the process of fetching from multiple heterogeneous data sources, using *BIOIInter* to retrieve original extracted and raw data results. The process of direct retrieval is described in the next section.

3.2. Raw-based biological data mining

If the result sets fetched from *Cache Manager* are unsatisfactory, the broker can further perform the query process directly from original data sources. The workflow and performance are shown in Fig. 11. The major steps are as the follows:

(1) The *Multi-Source Query Process* (*MSQP*) in *BIOIInter* accepts Q_{XML} from *MQP* of *MInter*. Using the DTD-based *FQT* of *BIOIInter*, *MSQP* verifies and parses Q_{XML} to get the information needed to retrieve the diverse data sources. The main information for correctly retrieving those sources on the Internet is the *DIR-entry* from the multi-source directory, which is verified by *FDT* of the Federated Directory Forum.

(2) The data sources each have their own wrappers, which generate proper query commands (Q_{DS}) to the specified data sources according to the

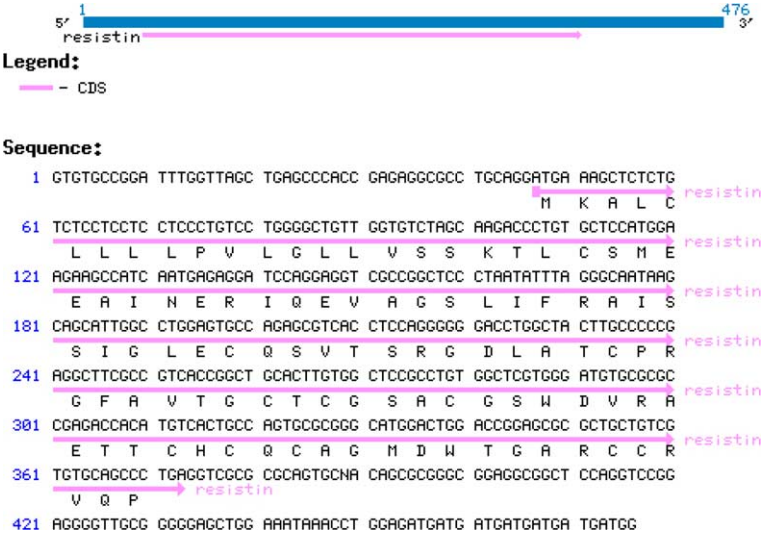


Fig. 10. Simple example of first phase mining with directory information.

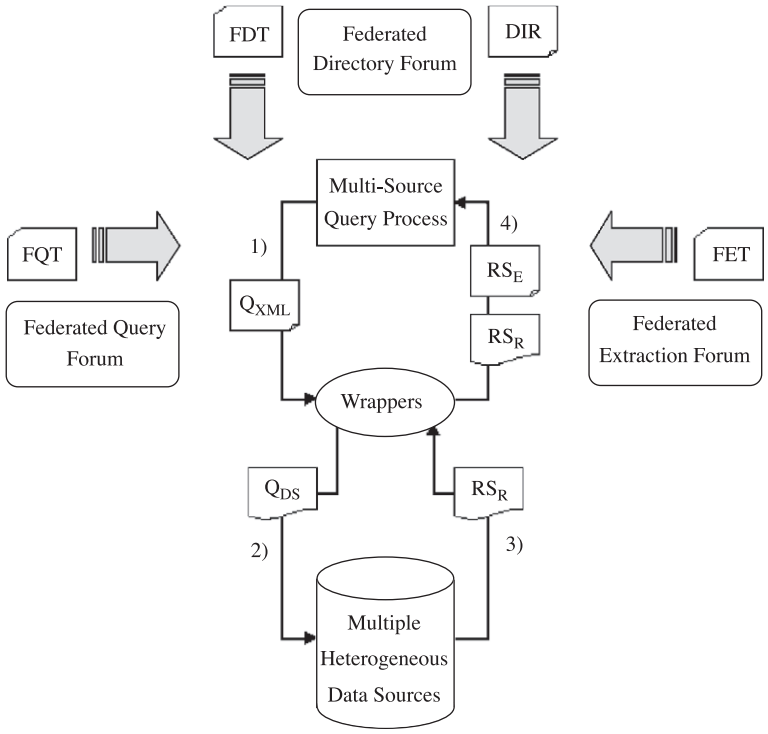


Fig. 11. Scenario of mining querying through data sources.

information in the Federated Query Forum. The broker can here contact all the heterogeneous data sources through the wrappers.

(3) The wrappers also handle results from data sources. The data sources can here generate result sets in the format of original raw data (RS_R) to the specified wrapper. The wrappers each request the *DTD*-based federated extraction template (*FET*) through the federated extraction forum in *BIOIInter*, after which the original raw data result sets are converted into a perceivable and unified format of extracted data result sets (RS_E) and stored as XML documentation. The same applies also to multiple heterogeneous data source wrappers.

(4) Multiple heterogeneous data source wrappers send original extracted and raw data results back to *MSQP* over the network. These sets are retained by the extracted and raw data receiver of *BIOIInter* and treated as supplementary result sets by the query process in the mining interface of the biological mining-broker. The query process satisfies the need for such query results by sending the original sets to the mining applications over the network.

3.3. Scenario of federated model

The federated model in our broker plays the most important role in coordinating the unified information for heterogeneous biological data mining. It provides the coordination in formatting directory information, the transparency of directory information, the formulation of biological data mining language, and in the planning of the extraction format for mining applications and data sources. Fig. 12 illustrates the federated model of the broker. With this model, the broker establishes an interactive mechanism within the discussion community for different mining applications, knowledge analysts, and multiple heterogeneous data sources. The model provides flexibility, interactivity, transparency, maintenance and enhances meaningfulness and usefulness for directory information, *BDMQL*, extraction information, and feedback

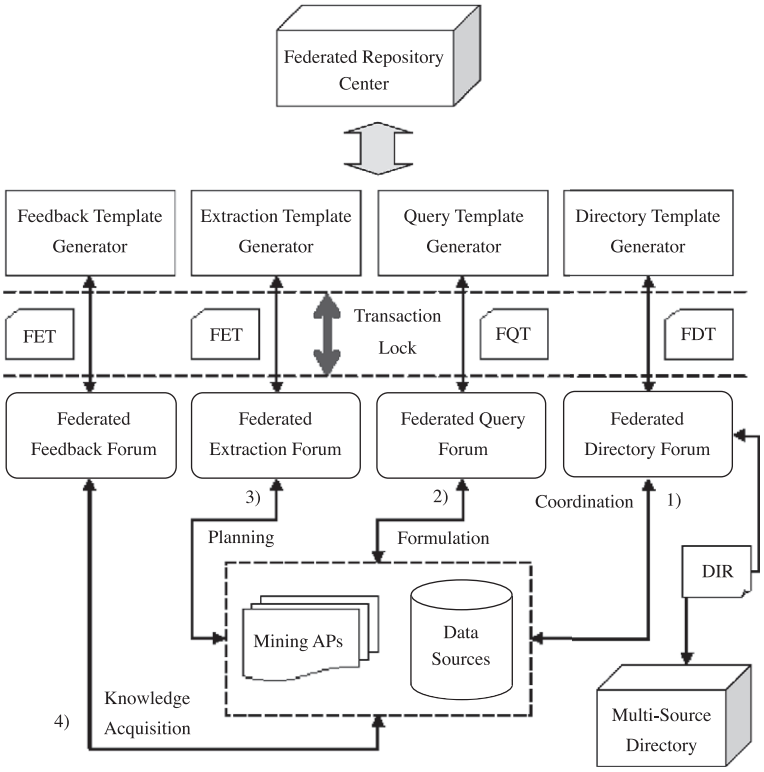


Fig. 12. Unification supporting in the scenario of federated model.

information. The major interactive federated forum mechanisms in the working of the model are as follows:

(1) *Transparency and coordination with the format of directory information*: As mentioned earlier, the process that includes cache-based and raw-based biological data mining queries needs to refer to *DIR-entry* from the multi-source directory through the federated directory forum to find the appropriate location information among the many heterogeneous data sources. Directory information of *DIR-entry* is the foundation of transparency for multiple heterogeneous data sources in the mining process. It belongs to the federated directory forum in *BIOIInter*. Thus, the federated directory forum integrates this kind of directory information into the multi-source directory, and agrees with mining applications in finding the diverse biological data on the network.

In the federated model, the *FDT* provides the definition for *DIR-entry* in the multi-source directory. The system developer constructs an appropriate template for a specified data source through a *GUI*-based directory template generator from a template developer's kits. Mining applications and knowledge analysts can also join the interactive discussion community on coordination and suggest which kinds of directory information data sources can be provided. Thus, the directory information can support the coherence and coordination of directory information formatting for the multiple heterogeneous data sources that underlie the federated model of the biological mining broker.

(2) *Formulating biological data mining query language*: Biological data queries usually contain lots of complicated information. To ease the formulation of the queries, *BDMB* provides *DTD*-based *Federated Query Template* (*FQT*) as the definition for well-formed and structured

XML-based mining queries. The system developer can use a specified *FQT* through a *GUI*-based query template generator from the template developer's kit. Then, the *DTD*-based definition of the queries can be suited to the individual specifications of the mining applications and knowledge analysts that underlie the federated model of the broker. In addition, multiple heterogeneous data sources can also join the interactive discussion community regarding formulation and propose their own special and data source dependent mining queries.

(3) *Planning of extraction format*: Syntactic and semantic heterogeneities are both critical problems for information retrieval from diverse heterogeneous data sources. We use the federated extraction forum to solve heterogeneity problems of related document. In the federated model, the *DTD*-based federated extraction template supplies the format definition for well-formed and structured *XML*-based extracted data from multiple heterogeneous sources. The system developer can construct an appropriate federated extraction template through a *GUI*-based extraction template generator from the template developer's kit. Thus, the planning of extraction format can be suited to individual knowledge analyst's requirements.

(4) *Knowledge acquisition among different knowledge analysts*: A feedback mechanism improves efficiency and precision of information retrieval from multiple data sources. With our broker, users can evaluate the importance of data sources at every stage of the data mining process. Since data sources each have their own characteristics, and knowledge analysts their own preferences, detailed information has to be constructed and maintained by the *DTD*-based federated feedback template. That is, the federated feedback template in the federated repository center provides a unified, estimative, and meaningful feedback form in a format suited to meet diversity. The system developer can design a federated feedback template through a *GUI*-based feedback template generator. Thus, the definition of feedback forms can be adapted to the particular mining feedback requirements of the knowledge analysts that underlie the federated model of the broker. According to the theory of knowledge acquisition [8,16], the broker provides the committee with knowledge analyst experts in the domain of biological data mining and can perform knowledge acquisition from feedback form. Thus, this federated model can serve as an essentially needed feedback-based meaningful cache mechanism.

This federated model promotes the power of the biological mining-broker, allowing it to establish an interactive mechanism in the discussion community for the various mining applications, knowledge analysts and heterogeneous data sources. It also promotes flexibility, interactivity, transparency, maintenance and enhanced meaningful and useful directory information, *BDMQL*, extraction information, and feedback information.

4. Experimental results

4.1. Hardware and software basics

The *BDMB* is mainly implemented using the Microsoft *Visual Basic 6.0 Enterprise Edition* with a *Windows 2000 Server*. All the *XML* and *DTD* files follow the specifications of *XML V1.0.* The biological sequence test data sets are managed in *Microsoft SQL 2000 server*. Moreover, the *XML* parser used in the system is *MSXML V4.0*, and the PC has one *Pentium III 800 MHz CPU* with 327 MB RAM and 100 Mbps Ethernet card.

We used the broker to perform a number of different but related biological data mining queries with an empty cache database, and then with three different data source samples. For the mining queries and related works, we simulated mining application translators and data source wrappers. The test data sources were *NCBI*, *DDBJ*, and *EMBL*, denoted as DS_1, DS_2, and DS_3, respectively. These were randomly sampled 1000 biological data, at a cost of 700 to 800 each KB. Thus, the total size for the three sources was about 2.1 to 2.4 GB. In addition, there were three extraction cache databases for DS_1, DS_2, and DS_3, with a cache size set to 500 each. The simulation parameters were designed to fit the restrictions of the experimental environment.

4.2. The design of experiments and simulations

Every specific execution plan had 10 rounds each, in which the broker performed eight different but related runs of biological data mining queries, denoted as R_1, R_2, . . .and R_8. The iterations in each round are as follows:

(1) *Iteration 1*, *denoted* I_1: The broker performed a two-phase cache based biological data mining query process through Cache Manager. The query is denoted as Q_i, where i means the current run number.
(2) *Iteration 2*, *denoted* I_2: The broker performed a mining query process directly from data sources DS_1, DS_2, and DS_3 with specific mining query Q_i.
(3) *Iteration 3*, *denoted* I_3: The broker performed a feedback-based meaningful cache mechanism process with a specific feedback form denoted F_j, where j means the current run number.
(4) *Iteration 4*, *denoted* I_4: The broker performed the two-phase cache-based biological data mining query process again with the same specific mining query Q_i.

According to traditional techniques [8], biological data mining queries Q_i are designed to perform in three specific domains, protein, genome, nucleotide, denoted P, G, N, respectively, and have three specific features, history, evolution, lineage, denoted as H, E, L. However, according to knowledge acquisition theory [8,16], the feedback forms F_j are designed to ensure and to enhance feedback-based semantics and the meaningful cache mechanism to rate overall conditions. They have four specific domains, protein, genome, nucleotide, structure, denoted P, G, N, S, and four specific features, history, evolution, lineage, breed, denoted H, L, E, B. The brief specifications of the mining queries Q_i and feedback F_j of each experimental are shown in Fig. 13.

4.3. Experimental results and analysis

Fig. 14 shows the evaluation of the simulation statistics after 10 rounds and eight runs. The format of the data item is "{*ExecutionTime*, *Amount*}", which refers to how long a specific iteration of the run took, starting from when a simulated mining application sent a specific mining query until when the related result sets were received. *Amount* refers to the size or number of the fetched data result sets. It deserves mentioning that the durations and amounts are presented as rounded mean values over the ten rounds spent on a specific execution plan. The unit of time measurement is a second with a maximum standard deviation of 8% of the mean value for all rounds.

Fig. 15 compares the time complexity I_1, I_4 and (DS_1, I_2), (DS_2, I_2), (DS_3, I_2) in each run of 10 rounds. It is obvious from the curves shown there and the data values shown in Fig. 14 that the average time complexity of cache-based biological data mining queries was far less than those taken directly from data sources.

Mining Query	Domain	Feature
Q_1	P, N	N/A
Q_2	N/A	H, L
Q_3	N	L
Q_4	P	H
Q_5	G, N	N/A
Q_6	N/A	E, L
Q_7	G	E
Q_8	N	L

Feedback Form	Overall	Domain	Feature
F_1	+7	P: +5	H
F_2	+5	P	H: +3
F_3	+8	P: +4	H: +5
F_4	+6	G	E
F_5	-5	N: -3	B: -4
F_6	-8	S: -4	L: -5
F_7	-7	N: -5	L: -3
F_8	-6	S: -4	B: -3

Fig. 13. Brief specifications of mining queries Q_i and feedback form F_j.

		DS_1	DS_2	DS_3	
Run	I_1	I_2	I_2	I_2	I_4
R_1	{0.732s, 0}	{150.117s, 376}	{121.837s, 395}	{121.34s, 349}	{0.093s, 1120}
R_2	{1.11s, 450}	{121.657s, 381}	{112.303s, 383}	{122.207s, 367}	{0.083s, 830}
R_3	{1.189s, 785}	{121.38s, 327}	{114.423s, 352}	{120.946s, 332}	{0.09s, 1011}
R_4	{1.136s, 679}	{120.2s, 340}	{115.06s, 345}	{124.993s, 304}	{0.083s, 701}
R_5	{1.156s, 983}	{143.107s, 372}	{116.256s, 380}	{125.037s, 366}	{0.094s, 983}
R_6	{1.174s, 694}	{126.12s, 375}	{114.804s, 405}	{123.927s, 399}	{0.07s, 694}
R_7	{1.19s, 619}	{120.39s, 336}	{111.297s, 349}	{122.806s, 341}	{0.067s, 619}
R_8	{1.194s, 908}	{121.684s, 327}	{113.753s, 352}	{123.62s, 332}	{0.093s, 937}

Fig. 14. Evaluation of simulation statistics.

In addition, Fig. 16 shows the performance of the cache hit-rate for I_1 and I_4 at each run, which is $C_{\mathrm{hr}}(R_m, I_n)$, computed as follows:

$$C_{\mathrm{hr}}(R_m, I_n) = \frac{\mathrm{ExAmount}_{Q_m I_n}}{\mathrm{RawAmount}_{Q_m I_2}} \times 100\%$$

where $ExAmount_{Q_m} I_n$ refers to the size or number of the extracted data result sets for the specific mining query Q_m at I_n, and $RawAmount_{Q_m} I_2$ means the sum of the amounts of the raw data result sets for Q_m at I_2 through DS_1, DS_2, and DS_3.

C_{hr} of I_1 at R_1 is 0.00% because initially extraction cache databases were empty and C_{hr} of I_4 at R_1 is 100.00% because the extraction cache databases were still not full after the first operation of the feedback-based meaningful caching.

Fig. 13 shows that Q_1, Q_2, Q_3, and Q_4 performed biological data mining queries on one group of related domains and features. Similarly, F_1, F_2, F_3, and F_4 rated the overall conditions of one group of related domains and features in positive ways, while Q_5, Q_6, Q_7, and Q_8, however, performed queries on another group. Then, F_1, F_2, F_3, and F_4 rated the overall conditions of the group in negative ways. That means that the experiments concerned two major yet different groups of test cases with some intersection in interests. They are (R_1, R_2, R_3, R_4) and (R_5, R_6, R_7, R_8).

Obviously, C_{hr} of I_1 is always higher than C_{hr} of I_4 for each run in the (R_1, R_2, R_3, R_4) group except for R_4, because the feedback-based meaningful cache mechanism ensures and enhances the meaningfulness of extraction cache databases in the broker. C_{hr} of I_1

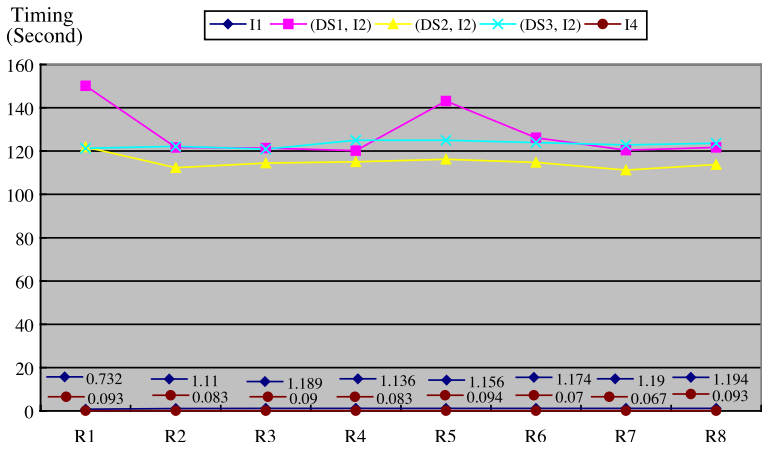


Fig. 15. Time complexity comparison in our simulation.

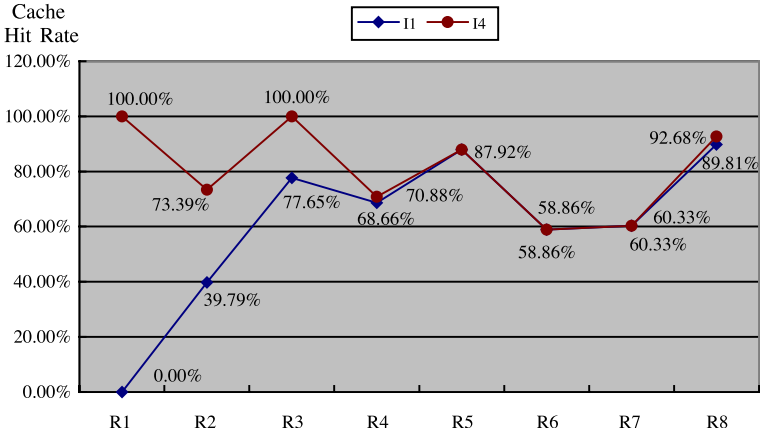


Fig. 16. Cache hit rate performance in simulation.

and I_4 at R_4 are similar because Q_4 is similar to those previous three mining queries with related domains and features. Thus, F_4 has less effect on overall rating conditions in extraction cache databases because of the similarity between Q_4 and the previous three mining queries. In addition, C_{hr} of I_i is consistently higher than that of I_{i-1} for each run in the (R_1, R_2, R_3, R_4) group except for R_2 and R_4, because feedback-based the meaningful cache mechanism works for this group with similar and related runs. C_{hr} of I_4 at R_2 is lower than C_{hr} of I_4 at R_1 because extraction cache databases are still not full after the first operation of feedback-based meaningful caching. C_{hr} of I_1 and I_4 at R_4 are lower than those at R_3 because in this group of runs, the difference between Q_3 and Q_4 is greater than any pair of mining queries.

The cache hit-rate in (R_5, R_6, R_7, R_8) is similar to that in (R_1, R_2, R_3, R_4). However, C_{hr} of I_1 is almost equal to or smaller than that of I_4 for each run in (R_5, R_6, R_7, R_8) because the feedback forms in (R_1, R_2, R_3, R_4) have much greater effects on overall rating conditions in extraction cache databases. Extracted result sets of I_1 for each run in (R_5, R_6, R_7, R_8) depend on the intersection of result sets with the group (R_1, R_2, R_3, R_4). Because of the much greater effect of the feedback forms of (R_1, R_2, R_3, R_4), the feedback forms of (R_5, R_6, R_7, R_8) have a smaller effect on the workings of the feedback-based meaningful cache mechanism until we reach R_8.

In addition, after the above eight runs with eight feedback forms over 10 rounds, two specific example test cases, (denoted TC_1 and TC_2), intended for rating the conditions of extracted data for meaningfulness (according to the theory of knowledge acquisition) were performed to evaluate the hit rate, confidence, and provide support [8] for biological data mining

Mining Query	Description for Test Case
X_1	Domain in 'Protein' and Feature in 'History' from Data Sources
X_2	Domain in 'Protein' and Feature in 'History' from Caches
X_3	Overall >= 0 for Rating Condition
X_4	Protein >= 0 and Feature >= 0 for Rating Condition
Y_1	Domain in 'Genome' and Feature in 'Evolution' from Data Sources
Y_2	Domain in 'Genome' and Feature in 'Evolution' from Caches
Y_3	Overall >= 0 for Rating Condition
Y_4	Domain is 'Genome' and Feature is 'Evolution' for Rating Condition

Fig. 17. Specific examples of test cases with their own mining queries.

queries. Test cases TC_1 and TC_2 with their own four mining queries X_1, X_2, X_3, X_4 and Y_1, Y_2, Y_3, Y_4, respectively, are shown in Fig. 17.

Fig. 18 shows the evaluation of the test cases. Each value relates to the number of extracted or raw data result sets for a specific mining query. The statistics for the execution time and the analysis of the time complexities are similar to those in Figs. 14 and 15. Fig. 18 shows, C_{hr} of TC_1=(the amount of extracted data result sets for specific mining query X_2)/(the sum of the amounts of raw data result sets for specific mining query X_1 through all data sources DS_1, DS_2 and DS_3)×100%=69.70%. X_2 is the same as Q_4 and C_{hr} of TC_1 is similar to that of R_4, because the feedback forms of (R_1, R_2, R_3, R_4) have much greater effect on the overall rating conditions in the extraction cache databases than (R_5, R_6, R_7, R_8). Similarly, C_{hr} of TC_2=60.26% is similar to the cache hit rate for R_7 with mining query Q_7 being the same as Y_2 for the same reason. The results for the test cases with mining queries X_3 and Y_3, respectively, were highly reliable. This is because of the effect of the feedback forms on the overall rating conditions in the extraction cache databases. In addition, the results for the test cases with mining queries X_4 and Y_4, respectively, were similarly satisfactory. Because the estimates in the rating conditions of the biological data referenced by X_4 and Y_4 were assigned to F_1, F_2, F_3, and F_4 (with the feedback forms being similar to each other), these forms were found to belong to (R_1, R_2, R_3, R_4) as was the case with similar mining queries Q_1, Q_2, Q_3 and Q_4. This resulted in overall rating conditions for similar extracted data result sets that can be cached in extraction cache databases.

	DS_1	DS_2	DS_3			
Test Case	**X_1**	**X_1**	**X_1**	**X_2**	**X_3**	**X_4**
TC_1	357	331	347	720	1074	781

	DS_1	DS_2	DS_3			
Test Case	**Y_1**	**Y_1**	**Y_1**	**Y_2**	**Y_3**	**Y_4**
TC_2	348	328	338	611	1074	702

Fig. 18. Evaluation of specific examples of test cases.

5. Conclusions and discussions

Our proposed Biological Data Mining-Broker (*BDMB*) was designed to undertake various fundamental tasks relating to multiple heterogeneous data sources and mining applications, where knowledge analysts engage in biological data mining queries. We investigated two-phase cache-based biological data mining queries and mining queries through multiple heterogeneous data sources. We sought to support the functionalities and capabilities of a unified format for the queries and the result sets and provide biological data mining processes with efficient and meaningful cache-based extracted data result sets, a semiautomatic mechanism for premining, and two-phase mining. The feedback-based meaningful cache mechanism and the federated model provided flexibility, interactivity, transparency, maintenance, and enhanced meaningfulness and usefulness.

The goal of the broker was not to replace human interpreters (such as knowledge analysts) or the functionalities of mining applications and data sources, but rather to make biological data mining easier, faster, and more efficient for domain experts. This was to be through an integrated, flexible, transparent, interactive, federated, semantic, and meaningful view supported by a unified, adjustable, and extractable broker. The experiments with several test cases and analysis by experimental simulation showed that with the customized discussion mechanism, the broker met the specific needs of a range of different mining applications and knowledge analysts. It is thus a natural choice for biological data mining processes with multiple heterogeneous data sources and different mining applications and knowledge analysts. It is highly useful for the purposes of target discovery and for bioinformatics research.

Acknowledgements

We are grateful for the many excellent comments and suggestions made by the anonymous referees. This work was supported in part by the National Science Council of the Republic of China under Grant No. NSC93-2752-E-009-006-PAE.

References

[1] B.A. Eckman, A.S. Kosky, L.A. Laroco Jr., Extending traditional query-based integration approaches for functional characterization of post-genomic data, Bioinformatics 17 (7) (2001) 587–601.

[2] The TAMBIS (Transparent Access to Multiple Biological Information Sources) System, http://bagel.cs.man.ac.uk/.

[3] N.W. Paton, R. Stevens, P. Baker, C.A. Goble, S. Bechhofer, A. Brass, Query processing in the TAMBIS bioinformatics source integration system, Scientific and Statistical Database Management, 1999, Eleventh International Conference, 1999, pp. 138–147.

[4] The Ontology of TAMBIS System (TaO); http://www.ontologos.org/OML/..%5COntology%5CTAMBIS.htm.

[5] R. Stevens, N.W. Paton, P. Baker, C.A. Goble, S. Bechhofer, A. Brass, TAMBIS Online: a bioinformatics source integration tool, Scientific and Statistical Database Management,1999, Eleventh International Conference, 1999, pp. 138–147.

[6] P. Bertone, M. Gerstein, Integrative data mining: the new direction in bioinformatics, IEEE Engineering in Medicine and Biology Magazine 20 (4) (2001 July–Aug.) 33–40.

[7] D.J. Cook, L.B. Holder, S. Su, R. Maglothin, I. Jonyer, Structural mining of molecular biology data, IEEE Engineering in Medicine and Biology Magazine 20 (4) (2001 July–Aug.) 67–74.

[8] J. Han, M. Kamber, Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, 2001.

[9] National Center for Biotechnology Information (NCBI), http://www.ncbi.nlm.nih.gov/.

[10] DNA Data Bank of Japan (DDBJ), http://www.ddbj.nig.ac.jp/.

[11] European Molecular Biology Laboratory (EMBL), http://www.embl-heidelberg.de/.

[12] World Wide Web Consortium (W3C), Extensible Markup Language (XML) Specification, DTD, http://www.w3c.org/XML/1998/06/xmlspec-report.

[13] C. Reynaud, J.-P. Sirot, D. Vodislav, Semantic integration of XML heterogeneous data sources, Database Engineering and Applications, 2001 International Symposium, 2001, pp. 199–208.

[14] Alin Deutsch, Mary Fernandez, Daniela Florescu, Alon Levyd, Dan Suciu, A query language for XML, Computer Networks 31 (1999) 1155–1169.

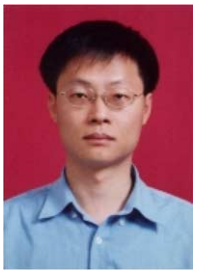
[15] E. Bertino, B. Catania, Integrating XML and databases, IEEE Internet Computing 5 (4) (2001 July–Aug.) 84–88.

[16] V. Estivill-Castro, Collaborative knowledge acquisition with a genetic algorithm, Tools with Artificial Intelligence, 1997, Proceedings, Ninth IEEE International Conference, 1997, pp. 270–277.

[17] T. Etzold, P. Argos, SRS: an indexing and retrieval tool for flat file data libraries, Comput. Appl. Biosci. 9 (1993) 49–57.

[18] Y.-S. Chang, M.-H. Ho, W.-C. Sun, S.-M. Yuan, Supporting unified interface to wrapper generator in integrated information retrieval, Journal of Computer Standards and Interfaces 24 (4) (2002) 291–309.

[19] Chun-Nan Hsu, Chia-Hui Chang, Chang-Huain Hsieh, Jiann-Jyh Lu, Chien-Chi Chang, Reconfigurable web wrapper agents for biological information integration, accepted by Journal of the American Society for Information Science.



Min-Huang Ho was born on February 1, 1969 in Kaohsiung, Taiwan, Republic of China. He received the BS and MS degree in Industrial Education from National Taiwan Normal University in 1993 and 1995, respectively, and the PhD degree from the Department of Computer and Information Science at National Chiao Tung University in 2004. Dr. Ho joined the Department of Information Management of Leader University as an assistant professor in the same year. His research interests are in Distributed Systems, Internet Technologies, Information Retrieval, and Mobile Agent Technologies.

Yue-Shan Chang was born on August 4, 1965 in Tainan, Taiwan, Republic of China. He received the BS degree in Electronic Technology from National Taiwan Institute of Technology in 1990, the MS degree in Electrical Engineering from the National Cheng Kung University in 1992, and the PhD degree from Computer and Information Science at National Chiao Tung University in 2001. Dr. Chang joined the Dept. of Electronic Engineering of Ming Hsing University of Science and Technology (MUST) as a lecturer in August 1992. Since August 2001, he had been an associate professor. He had been the Director of Computer Center of MUST in August 2002. Since August 2004, he joined the Dept. of Computer Science and Information Engineering, National Taipei University, Taipei county, Taiwan. His research interests are in Distributed Systems, Object Oriented Programming, Information Retrieval and Integration, and Internet Technologies.

Kuang-Lee Li was born on August 14, 1978 in Taipei, Taiwan, Republic of China. He received the BS and MS degrees from the Department of Computer and Information Science at National Chiao Tung University (NCTU) in 2000 and 2002, respectively. Mr. Li joined the Biomedical Engineering Center, Industrial Technology Research Institute as an Assistant Research Member in January 2003. Since December 2000, he had been the lecturer at Institute for Information Industry (III) and Tze-Chiang Foundation of Science and Technology (TCFST). He also had been the Computer Science related consultant and lecturer for several companies at Hsinchu Science Park, Taiwan. His research interests are in Data Mining, Object-Oriented Software Engineering, Bioinformatics and Software System Integration.

Shyan-Ming Yuan was born on July 11, 1959 in Mauli, Taiwan, Republic of China. He received the BSEE degree from National Taiwan University in 1981, the MS degree in Computer Science from University of Maryland Baltimore County in 1985, and the PhD degree in Computer Science from University of Maryland College Park in 1989. Dr. Yuan joined the Electronics Research and Service Organization, Industrial Technology Research Institute as a Research Member in Oct. 1989. Since September 1990, he had been an Associate Professor at the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. He became a Professor in June, 1995. His current research interests include Distributed Objects, Internet Technologies, and Software System Integration. Dr. Yuan is a member of ACM and IEEE.