Integrative Warehousing of Biomolecular Information to Support Complex Multi-Topic Queries for Biomedical Knowledge Discovery

Arif Canakoglu, Marco Masseroli, Stefano Ceri, Luca Tettamanti, Giorgio Ghisalberti, and Alessandro Campi

***Abstract*—Biomedical questions are often complex and address multiple topics simultaneously. Answering them requires the comprehensive evaluation of several different types of data. They are often available, but in distributed and heterogeneous data sources; this hampers their global evaluation. We developed a software architecture to create and maintain updated a Genomic and Proteomic Data Warehouse (GPDW), which integrates several of the main of such dispersed data. It uses a modular and multi-level global data schema based on abstraction and generalization of integrated data features. Such a schema eases integration of data sources evolving in data content, structure and number, and assures provenance tracking of all the integrated data. Thanks to the developed software architecture and adopted data schema, the GPDW has been kept updated easily and progressively extended with additional data types and sources; it is publicly usable at http://www.bioinformatics.dei.polimi.it/GPKB/ .**

I. Introduction

THE life sciences are characterized by an increasingly large amount of valuable but heterogeneous and sparse biomolecular data [1]. Their comprehensive evaluation can support addressing current biomedical questions, which generally are complex and regard multiple different topics. Yet, various information about a given biomolecular entity are often scattered across many diverse sources. Hence combining information from multiple sources is paramount for current biomedical-molecular investigation.

Several approaches have been proposed to integrate data from multiple heterogeneous data sources, including *information linkage*, *multi databases*, *federated databases*, *mediator based solutions* and *data warehousing*. The last one well supports applications where off-line processing of numerous data from various and dispersed sources is required, e.g. in order to comprehensively and efficiently mine the integrated data towards knowledge discovery. Yet, data warehousing may generate maintenance difficulties in keeping the data warehouse up-to-date and expanding it with additional data and data types from new sources [2]. Often in biomolecular databases, data vary frequently; in these cases warehousing requires automatic procedures to retrieve the new data from the distributed sources that have been integrated and to maintain the warehouse updated [2]. In data warehousing, integration is performed off-line, based on a predefined data model that provides a unified reconciled global view of the data. Difficulties mainly arise from the adoption of global schemas proposed for biological data [3-5]. These data schemas generally are very expressive and complex; making it difficult to face the integration challenges of evolving data. Furthermore, they usually do not provide good support for data provenance and version tracking, as well as for integration of different and overlapping sources providing the same data type [6]. Yet, integration of such data sources can improve integrated data coverage and quality, through identification of mismatching information by cross-validation of overlapping data [7].

Leveraging our previous experience with the GFINDer system [5] and [8], we developed a software architecture to create and maintain an updated and publicly available integrative data warehouse of selected genomic and proteomic controlled annotation data of different species. It represents the available biomedical-molecular knowledge and adopts a novel modular and multi-level global schema that we propose for integrated data. Such a data schema supports integration of data sources, also overlapping, which are fast evolving in data content, structure and number, and assures provenance tracking of all the integrated data.

II. Integrated Data Schema

We defined an integrated data schema which is composed of multiple interconnected modules. Each module represents a single feature or topic, with data provided by one or more of the integrated data sources and containing provenance information for each single feature instance (Figure 1). In the case of controlled biomedical-molecular annotation data, which we focused on, a feature can be a biomolecular entity (i.e. DNA sequence, gene, transcript, protein), or a biomedical feature (e.g. pathway, genetic disorder, etc.), as specified in its *feature_type* attribute. Feature attributes common to all, or most of, the data sources that provide data for the feature are drawn on the feature main table. Among them, the *source_id* and *source_name* attributes identify each feature data instance; the *source_name* represents the source to which the *source_id* belong. The *reference* attribute represents the source that provides the data. Together with the *reference_file* attributes in each of the source tables, it allows provenance storing and transparent

Manuscript received July 30, 2013. This work was supported in part by the "Search Computing" (SeCo) project (2008-2013), funded by the European Research Council (ERC), under the 2008 call for "IDEAS Advanced Grants".

A. Canakoglu, M. Masseroli, S. Ceri, L. Tettamanti, G. Ghisalberti and A. Campi are with the Department of Electronics, Information and Bioengineering, Politecnico di Milano, Milano, IT 20133 Italy. (phone: +39-02-2399-3553; e-mail: [canakoglu\masseroli\ceri\tettamanti\ghisalberti\campi]@elet.polimi.it ; fax: +39-02-2399-3411).

978-1-4799-3163-7/13/$31.00 ©2013 IEEE

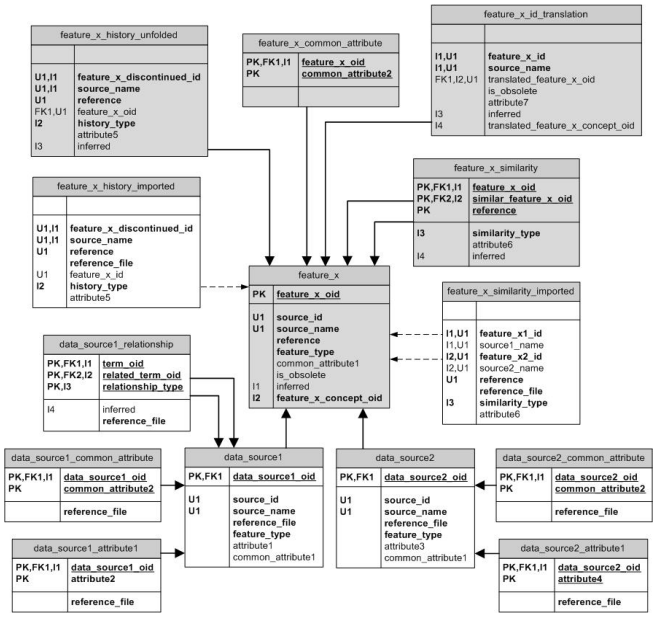

Fig. 1. Multi-level feature module of the defined integrated global data schema. White tables represent import level data, whereas dark tables represent aggregation level data.

tracking of each data. Furthermore, every biomolecular entity instance is characterized by its *symbol* and the *taxonomy_id* of the organism to which it belongs. Similarly, every biomedical feature instance is characterized by its *name* and *definition*.

Each feature of our general schema can also have *history* and/or *similarity* data. The former data represent obsolete discontinued source IDs and the current ID that replaced them, if discontinued ones have been propagated. The latter data describe equivalence of different feature instances (from the same or different sources). These similarity data can be imported from one or more sources, or inserted by expert curators, or inferred automatically by some computational process. Both *history* and *similarity* data are paramount to identifying multiple feature instances, from single or multiple data sources, as representing the same feature concept. Thus they directly support the integration of different feature instances. Finally, our schema can represent controlled descriptions of feature instances expressed through a flat terminology or an ontology. In the latter case, ontological relationships among hierarchically related feature data from the same data source are stored in a *relationship* entity of a data source.

In order to ease maintenance and extension of the global data schema, each feature module is internally organized in two levels: an *import level* and an *aggregation level*. The import level is composed of separated sub-schemas, each one for every single data source considered which provides data for that feature. These sub-schemas are individually structured as in the original data source, i.e. in a global-as-view (GAV) data integration fashion. The import level allows structuring and locating originally distributed data in the same database, while thoroughly checking their consistency and quality [7], and identifying the feature they refer to and their main attributes. This allows automatic aggregation of the main attribute data of each feature source in the aggregation level, where they are associated with a unique OID while duplicate instances are identified. It also eases maintenance and expansion of the global data schema. In fact, if data schema variations occur in the original data sources, they can be easily managed since they affect only the source specific part of the schema. Similarly, integration of an additional data source, which provides data for that feature, only requires adding a sub-schema for the new data source (according to its original data schema), without affecting other parts of the global schema. Then, the aggregation level of the global schema is automatically derived from the modified import level.

Data feature modules can be pairwise associated (through association/annotation data); also these associations are organized in an import and an aggregation level. In the latter one, the association data, which are contained in the import level as feature instance ID pairs, are translated into pairs of unique OIDs and matched to the feature instance OIDs of the two associated features. By leveraging available history data, this also allows reconciling discontinued IDs of external sources to their current ones.

Finally, a third, higher and more general *integration level* (not shown in Figure 1) completes the integrated data schema by individually representing all unique feature concepts and their associations described by the integrated data, regardless of the source(s) that provide them. Yet, each of these unique concepts is related to all its instances from distinct data sources in the lower schema levels, thus keeping all its provenance information.

III. Software Architecture for Data Integration

Leveraging the modularity of our data schema, we created a generalized and parametric software architecture in Java programming language. It supports the automated creation of a data warehouse adopting our schema and makes updating the data warehouse and extending it with new data sources easy. Our approach for the integration of distributed multi-source heterogeneous data is divided in two macro steps: A) *importing* data from their diverse sources in the *source-import* level of the global data schema and B) *integrating* them in the *instance-aggregation* and *concept-integration* levels of the data schema.

A. Data Import

The data import procedure is guided by an *import manager* that instantiates, configures and executes an *importer* for each considered data source. Each source specific importer coordinates a set of *loaders* (a loader for each data file, group of homogeneous data files, or data access API provided by the source) and a set of *parsers* (a parser for each data format). Each parser extracts the data from its associated input file(s) or API(s) and produces data tokens usable by the loader. Each loader is responsible for

associating a semantic meaning to the tokens produced by the associated parser(s) and inserting them into the data warehouse.

In the import process, each actor is independent: the import manager administers the importers via a standard interface based on java reflection API. The parser is aware of data format, but agnostic of data semantics; the loader receives data in a standard format and inserts them in the proper data warehouse tables. The importing is built to be very flexible and allows adding new sources as easily as possible. To reach this goal, the process is guided by a *configuration file*; it contains the list and descriptions of all the registered data sources to be imported, all the features (biomolecular entities and biomedical features) described by the data to be imported and theirs bindings. Such descriptions are used to map each data source and feature to one or more tables in the data warehouse and their bindings are used to populate such tables.

The importing framework assigns to each imported "data record" an OID, which is unique across the data warehouse. It is used as the primary identification of the data entries, since there is no guarantee that the IDs provided by the different sources do not conflict with each other. In order to ensure correctness of imported data, a set of regular expressions has been defined to check and identify IDs [7]. They are used by the *ID matcher*, an additional component of our framework that acts as mediator between the loaders and the data warehouse. The main role of this mediator is to check ID syntactic correctness and identify the semantic type of each ID, in order to insert the correct information in the appropriate data warehouse tables. During this process, each inserted tuple is also enriched with provenance metadata to track its source. Correct ID identification is paramount since data from multiple sources are then linked together thanks to association data provided by the integrated sources as pairs of IDs in different data sources.

At the end of the import process, index, unique, primary and foreign key integrity constraints are defined and enforced upon the data warehouse tables.

B. Data Integration

The main tasks automatically performed in the data integration step can be grouped into an *aggregation* and an *integration* phase. In the former one, data from the different sources, imported in the previous data import step, are gathered and normalized into a single representation in the *instance-aggregation* level of our global data model. In the latter phase, data are organized into informative clusters in the *concept-integration* level of the integrated data schema.

During the initial aggregation phase, integrated tables of the features described by the imported data are created and populated. Then, similar IDs (e.g. aliases of feature IDs) and historical IDs, which are sometimes provided by the data sources, are translated to our internal OIDs. Unfolding of historical IDs is performed before OID translation, so as to associate repeatedly superseded and discontinued IDs with the translated OID of their latest ID. Integrated entries derived from this processing are marked as *inferred through historical data*, in order to keep full track of their generation process. Both similarity and historical ID data are extremely valuable for subsequent data integration tasks. Translation tables for biomolecular entity and biomedical feature IDs are also created by using translated similarity data and unfolded historical ID data. These serve as main entry points to query and explore the data warehouse; they allow the conversion from a number of user-provided identifiers (also obsolete or alias of those in the warehouse feature tables) to a set of current OIDs, which are usable to navigate the warehouse.

Finally, associations (annotations) between pairs of feature entries are created by performing OID translation of the imported association data expressed through the "native" IDs. In doing so, association data are coupled with the related feature entries. Depending on the imported data sources and their mutual synchronization, association data may refer to feature entries, or even features, that have not been imported in the data warehouse (yet). In this case, missing integrated feature entries are synthesized and marked as *inferred through synthesis from association data*. However, if a missing entry has an obsolete ID and through unfolded translated historical data its most current ID can be obtained, the association is transferred to the latest ID and marked as *inferred through historical data*. This association translation policy preserves all associations expressed by the imported association data between the integrated data; thus, it allows subsequent using such associations for biomedical knowledge discovery (e.g. by transitive relationship inference [9], involving also the synthesized entries).

During the final integration phase, through a "similarity analysis", it is checked whether single feature instances from different sources represent the same feature concept. In this case, they are associated with a new single concept OID. Furthermore, new entries can be inferred from the integrated data, e.g. as previously mentioned. The *Inferred* attribute in the integrated tables is used to keep track of the inference method employed, if any, to derive an entry.

IV. GENOMIC AND PROTEOMIC KNOWLEDGE BASE (GPKB)

We use the described general software architecture and integrated data schema, implemented in a PostgreSQL RDBMS, to create, maintain updated and progressively extend a multi-organism integrative Genomic and Proteomic Data Warehouse (GPDW). It currently contains more than 1.84 billion data tuples, which amount to a total of about 383 GB of disk space (included index space). It integrates data of very numerous biomolecular entities and their interactions and annotations with many biomedical-molecular features imported from several distributed databases, including Entrez Gene, UniProt, IntAct, MINT, Expasy Enzyme, BioCyc, KEGG, Reactome, GO, GOA and OMIM. At the time of writing, it contained data about 9,537,645 genes of 9,631 different organisms, 38,960,202 proteins of 338,004 species and a total of 71,737,812 gene annotations and 118,165,516 protein annotations regarding 12 biomedical

controlled terminologies. The latter ones included, among others, 35,252 Gene Ontology terms and their 64,185,070 annotations, 28,889 biochemical pathways and 171,372 pathway annotations, and 10,212 human genetic disorders and their 27,705 gene annotations, together with 38,901 phenotypes (signs and symptoms). These last, which we extract from OMIM clinical synopsis semi-structured descriptions as described in [5], at the time of writing to our knowledge are not included in any other integrative database publicly available. Furthermore, the GPDW also integrates 542,873 very valuable interaction data between several different biomolecular entities, including 539,718 protein-protein interactions.

The GPDW constitutes the backend of a Genomic and Proteomic Knowledge Base (GPKB) publicly available at http://www.bioinformatics.dei.polimi.it/GPKB/ . An easy-to-use Web interface enables the scientific community to access and comprehensively query all the data integrated in the GPDW and to take full advantage of them. For example, the GPKB user can find an answer to a search for the enzymes codified by genes known to be involved in the same pathways and find in which cellular components each of such enzymes is known to be expressed. Alternatively, he/she can also search for all genes whose alterations are known to be associated with given genetic pathologies, find the symptoms of such pathologies and the pathways in which these genes are known to be involved, and check if common pathways and symptoms exist in different related pathologies (e.g. in muscular dystrophy and in amyotrophic lateral sclerosis), etc.. Such complex multi-topic queries cannot be performed in other available systems.

V. Discussion and Conclusions

The modular nature of the integrated data schema that we defined allows creating warehousing integrations according to the features to be represented and the data sources to be imported. In particular, it can be incrementally updated and expanded with additional entities and associations while new data and data sources are becoming progressively available. Thus, our proposed data schema is particularly suited to tackle the difficulties rising in keeping updated and extending integrated life science data collections, which evolve both in content and, although less frequently, in structure and number. Different from previous warehousing proposals (e.g. BioWarehouse [11]), our integrated schema can be iteratively extended in a seamless and modular way to include many biomedical features, biomolecular entities and their associations, with virtually no limitations.

Despite its generality, the modular and multilevel hierarchical structure of our global data schema allows defining automatic algorithms that allow query composition based only on the data attributes that the user wants to include in the query. These algorithms also guide the generation of well-performing queries, even when run on big quantities of data, as thoroughly illustrated in [10]. Such assets distinguish our data schema from usual generic models, which often can cause performance challenges. Through recording of data provenance and modeling of associations among the integrated data, our data schema can support comprehensive feature association-based analyses of data from multiple sources well, thus fostering biomedical knowledge discoveries.

In conclusion, a modular multilevel data schema, such as the one that we propose, eases maintenance and extension of integrated collections of evolving data. It also supports composition of efficient queries though complex and multi-topics. Using such data schema, the GPKB has been created and easily progressively extended and kept updated. It constitutes a unique integrative knowledge base very valuable for data mining algorithms that take advantage of integrated controlled annotation data, e.g. for gene and protein annotation inference and annotation-based functional similarity evaluations and clustering, or for annotation enrichment analyses of gene or protein lists. The GPKB also well supports multi-topic searches that can help complex biomedical question answering and biomedical knowledge discovery.

References

[1] X. M. Fernández-Suárez, M. Y. Galperin, "The 2013 Nucleic Acids Research Database Issue and the online Molecular Biology Database Collection", *Nucleic Acids Res*., vol. 41, D1, pp. D1-D7, 2013.

[2] S. B. Davidson, C. Overton, P. Buneman, "Challenges in integrating biological data sources", *J. Comput. Biol*., vol. 2, pp. 557-572, 1995.

[3] N. W. Paton, S. A. Khan, A. Hayes, F. Moussouni, A. Brass, K. Eilbeck, et al., "Conceptual modeling of genomic information", *Bioinformatics*, vol. 16, 6, pp. 548-557, 2000.

[4] E. Bornberg-Bauer, N. W. Paton, "Conceptual data modelling for bioinformatics", *Brief Bioinform*., vol. 3, 2, pp. 166-180, 2002.

[5] M. Masseroli, O. Galati, F. Pinciroli, "GFINDer: Genetic disease and phenotype location statistical analysis and mining of dynamically annotated gene lists", *Nucleic Acids Res*., vol. 33, pp. W717-W723, 2005.

[6] W. Sujansky, "Heterogeneous database integration in biomedicine", *J. Biomed. Inform*., vol. 34, 4, pp. 285-298, 2001.

[7] G. Ghisalberti, M. Masseroli, L. Tettamanti, "Quality controls in integrative approaches to detect errors and inconsistencies in biological databases", *J. Integr. Bioinform*. vol. 7, 119, pp. 1-13, 2010.

[8] M. Masseroli, "Management and analysis of genomic functional and phenotypic controlled annotations to support biomedical investigation and practice", *IEEE Trans. Inf. Technol. Biomed*., vol. 11, 4, pp. 376-385, 2007.

[9] M. Quigliatti, M. Masseroli, "In silico identification of new biomolecular annotations", *Proc. VIII Annual Meeting of the Bioinformatics Italian Society (BITS 2011),* Pisa, IT, 2011, pp. 19-20.

[10] F. Pessina, M. Masseroli, A. Canakoglu, "Visual composition of complex queries on an integrative genomic and proteomic data warehouse", *Proc. 7th International Conference on Bioinformatics and Biomedical Engineering (iCBBE 2013),* 2013, pp. 1-4, (in press).

[11] T. J. Lee, Y. Pouliot, V. Wagner, P. Gupta, D. W. Stringer-Calvert, J. D. Tenenbaum, P. D. Karp, "BioWarehouse: a bioinformatics database warehouse toolkit", *BMC Bioinformatics* , vol. 7, 170, pp. 1-14, 2006.