BioFuice: Mapping-Based Data Integration in Bioinformatics

Toralf Kirsten and Erhard Rahm

University of Leipzig, Germany
tkirsten@izbi.uni-leipzig.de, rahm@informatik.uni-leipzig.de

Abstract. We introduce the BioFuice approach for integrating data from different private and public data sources and ontologies. BioFuice follows a peer-to-peer-like data integration based on bidirectional mappings. Sources and mappings are associated with a domain model to support a semantically meaningful interoperability. BioFuice extends the generic iFuice integration platform which utilizes specific operators for data fusion and workflow-like script programs. BioFuice supports explorative data analysis and query and search capabilities. We outline the integration approach by an illustrating scenario, the architecture of BioFuice and its query interface.

1 Introduction

Many biological and medical applications require access to a variety of molecular-biological objects, such as genes, proteins, their interrelationships and functions, and their correlations with phenotypical effects. These objects are maintained in a high number of diverse web-accessible data sources [Ga05] as well as in local (private) data sources, e.g. specific analysis results such as a particular list of genes or medical data on patients participating in clinical trials. Typically, such data is highly diverse so that their integration is laborious and error-prone and difficult to perform by domain experts.

Traditional data integration approaches like data warehousing and mediators are often applicable but also time-consuming to deploy and may lack sufficient support for features such as explorative data analysis. These integration approaches typically require a unified global schema to obtain a consistent view over data from different sources. However, creating such a schema for more than a few data sources is almost impossible due to the high diversity, complexity and fast evolution of sources. Each new source to consider may require adapting the global schema as well as applications built upon this schema.

A promising alternative to the traditional data warehousing and mediator solutions using a global schema are so-called peer-to-peer approaches for data integration [Ha03]. They are based on bilateral mappings between autonomous data sources, called data peers, instead of mappings between data sources and a global schema. Adding a new data source can thus be achieved by mapping it to only one existing peer instead of adapting the global schema and mapping the source to it. In bioinformatics, a peer-to-peer approach seems especially appropriate since bilateral mappings can often be derived from existing cross-references between objects of different

U. Leser, F. Naumann, and B. Eckman (Eds.): DILS 2006, LNBI 4075, pp. 124 – 135, 2006.

sources. Such cross-references refer to so-called accessions, i.e. unique object identifiers, and are omnipresent in public data sources. The cross-references are typically maintained by domain experts and thus of high quality. However, they are currently used mostly for manual web navigation which is unsuitable for evaluating large sets of objects, e.g. for gene expression analysis. Moreover, the semantics of the cross-references is typically not made explicit making it difficult for the user to find and correctly use all relevant sources and mappings for a given application task.

iFuice (information Fusion utilizing instance correspondences and peer mappings) [Ra05] is a recently proposed approach for peer-to-peer data integration. It utilizes mappings, e.g. sets of cross-references, to combine or fuse information from different sources. Sources and mappings are related to a domain model to support semantically meaningful information fusion. The iFuice architecture incorporates a mapping mediator offering both interactive and script-driven, workflow-like access to the sources and their mappings. The script programmer can use powerful generic operators to execute and manipulate mappings and their results. iFuice is a generic data integration approach which is not targeted for a specific application domain. An initial use case of iFuice was to combine bibliographic data for a citation analysis of database publications [Ra05, RT05].

In this paper we describe how iFuice and its extension BioFuice can be used for data integration in bioinformatics applications. Key characteristics of BioFuice include:

- *Peer-to-peer integration:* By following the iFuice paradigm BioFuice aims at utilizing instance-level cross-references which already exist, e.g. as web links, or can be generated by bioinformatics tools, such as BLAST. New sources can be dynamically integrated as needed by mapping the new source to (at least) one already integrated source.
- *Semantic integration:* To address semantic integration, BioFuice utilizes a high-level domain model containing domain-specific object types and mapping types. The domain model is used to categorize specific sources and mappings so that they can be selected and accessed according to current application requirements.
- *Comprehensive query capabilities:* BioFuice utilizes the high-level operators and scripting facility of iFuice to perform data access, mapping execution and data fusion. This infrastructure makes it possible to react to new application needs and to support complex data integration and analysis workflows. BioFuice substantially extends the generic iFuice facilities by providing a graphical query interface for explorative analysis and automatically generating script programs from interactively specified queries. Both predefined queries as well as keyword searches are supported.
- *Local data sources:* BioFuice integrates both public and local (private) data sources. In particular, query and script results or copies of entire sources may be stored within a local database for later reuse. BioFuice can also be operated in an offline mode (e.g. on a notebook) by only evaluating local data sources.

The rest of the paper is organized as follows. In the next section we introduce the basic idea of the BioFuice approach by using an illustrating scenario. We also outline selected high-level operators and their usage. In Section 3, we introduce the BioFuice