SIBIOS Ontology: A Robust Package for the Integration and Pipelining of Bioinformatics Services

Malika Mahoui[1], Zina Ben Miled[2], Sriram Srinivasan[3], Mindi Dippold[1], Bing Yang[2], and Li Nianhua[2]

[1] School of Informatics, IUPUI
{mmahoui, mimeie}@iupui.edu
[2] Department of Electrical and Computer Engineering, IUPUI
{zmiled, niali, yanbing}@iupui.edu
[3] Department of Computer and Information Sciences, IUPUI
srsriniv@iupui.edu

Abstract. The recent technological advancements in biological research have allowed researchers to advance their knowledge of the domain far beyond expectations. The advent of easily accessible biological web databases such as NCBI databases and associated tools such as BLAST are key components to this development. However, with the growing number of these web based biological research tools and data sources, the time necessary to invest in becoming a domain expert is immense. Therefore, it is important to allow for easy user deployment of the wealth of available data sources and tools necessary to conduct biological research. In this paper we discuss an approach to create and maintain a robust ontology knowledge base that serves as the core for SIBIOS, a workflow based integration system for bioinformatics tools and data sources. Further, deployment of the ontology in various components of SIBIOS is discussed.

Keywords: Data integration, scientific workflows, ontologies, fault tolerance.

1 Introduction

Data integration and service discovery in the Life Sciences are key challenges that impede discoveries in biology and bioinformatics. The necessity in retrieval of available data that are generated by the technologically evolving field of biology and bioinformatics has resulted in introduction of more supporting tools. For example, a user may be interested in a particular gene such as *BRAC1 Human Gene* [1]. The user may use *GENBANK* [2], a public nucleotide sequence repository to retrieve the gene sequence. The results of this search can be given to BLAST [3] to find additional genes with similar conserved regions. The next step may be the translation of the gene sequence found into 6 reading frames by *TRANSEQ* [4] to find proteins of interest. Finally, the structure and functional motifs of the protein may need to be studied via services such as *PRINTS* [5] and *FINGERPRINTSCAN* [6] in order to find additional information related to the effects of mutations in the *BRAC1 gene* [1]. The process is known as in-silico experiment and involves a mixture of database and tools deployed in a workflow fashion. In-silico experiments take time and require sophisticated expertise from biologists due to two main reasons. In one hand, data sources and

U. Leser, F. Naumann, and B. Eckman (Eds.): DILS 2006, LNBI 4075, pp. 104 – 113, 2006.

bioinformatics tools hereafter referred as *bioinformatics services* lack a registry mechanism by which researchers are able to retrieve the services needed for their experiments, just by relying on a set of metadata for service description provided as part of service registration. As a result only a handful list of databases such as GenBank [2] and SwissProt [7], and a limited number of bioinformatics tools such as Blast [3] are used by the research community; while hundreds of other services [8] offering valuable quality data and analysis capabilities remain under-utilized. On the other hand, the distributed nature and heterogeneity of bioinformatics services at the syntactic as well as at the semantic levels render their manipulation and deployment cumbersome and time consuming. Hence researchers need to continuously work on the interoperability between services by data copying and pasting, and when necessary performing data formatting and data filtering operations. Therefore there is a great value in automating the process of service selection, service composition and service invocation when working with in-silico experiments.

Several integration systems are being proposed to assist researchers in conducting their analyses [9, 10, 11, 12, 13, 14]. These initiatives can be broadly classified under either a warehouse approach or a wrapper based approach. A global schema is used to reconcile service heterogeneity in warehouse solutions by having copies of bioinformatics services at the server hosting the integrating system. Solutions based on the wrapper approach often use ontologies as the basis for their integration solutions [10, 11, 15, 16, 17]. SIBIOS, the system for the integration of bioinformatics services falls into the latter approach [13, 15, 18].

This paper describes the main challenges faced during the design and the maintenance of SIBIOS ontology. It also describes the ontology features to support easy deployment of SIBIOS system by researchers.

Section 2 briefly describes the main features of SIBIOS. The ontology design will be detailed in Section 3. Section 4 describes how the ontology is deployed within SIBIOS system. Discussion on related and future work is presented in Section 5.

2 Overview of SIBIOS Architecture

SIBIOS operates in a distributed client-server environment in order to facilitate service selection and dynamic execution of workflows [18]. The main components of SIBIOS architecture are highlighted in figure 1. Workflow building is aided by the service composition module. The tasks of service composition are twofold: assist the user in selecting the appropriate services which compose the workflow and ensuring correct pipelining of services. Correct pipelining of services ensures that a service *s2* can be composed after service *s1* only if service s2 is able to use the output of service *s1* as input parameters. Service selection offers two options for the user to select services for the workflow. The semi-automated mode of service selection offers a step by step process where the user selects the services that are needed and assembles the workflow. In the second alternative the user will submit a high level description of the workflow that will be passed on to the automated service composition. This latter module will generate a list of potential workflows from which the user will select the most appropriate one for execution. The fault tolerance module enhances system reliability during workflow execution by the workflow enactment.