

INTERNSHIP REPORT

Scientific Modeling and Resource Discovery in Scientific Workflows

An internship report presented in
partial fulfillment of the requirement
of the Professional Science Master's
in Computational Biosciences

Maliha Aziz

*Computational Biosciences Program
Arizona State University*

Dr. Zoé Lacroix

*Internship Advisor
Department of Electrical Engineering
Arizona State University*

Internship:

From : January, 2007 - May, 2008

THIS REPORT IS NOT CONFIDENTIAL

Technical Report Number: 08-03
May 8, 2008

ACKNOWLEDGEMENT

This work was partially supported by the National Science Foundation¹ (grants IIS 0431174, IIS 0551444, and IIS 0612273).

I would like to thank Dr. Zoé Lacroix for giving me an opportunity to work with her as a Research Assistant in the Scientific Data Management Lab at Arizona State University. Her guidance and support motivated me to explore the field of computational biosciences in new and exciting ways.

I would also like to thank Dr. Nathalie Meurice whose knowledge in her field has continued to impress me since the day I met her during the Helios internship at TGen where she was my mentor. I am grateful for all her advice and support.

¹ Any opinion, finding, and conclusion or recommendation expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

TABLE OF CONTENTS

| | |
|--|----|
| LIST OF TABLES AND FIGURES..... | 4 |
| ABSTRACT..... | 5 |
| GOAL OF THE PROJECT..... | 7 |
| INTRODUCTION..... | 8 |
| METHODOLOGY..... | 10 |
| SCIENTIFIC MODELING..... | 10 |
| RESOURCE DISCOVERY..... | 18 |
| CONCLUSIONS AND FUTURE DIRECTIONS..... | 25 |
| REFERENCES..... | 27 |
| APPENDIX..... | 29 |
| APPENDIX A..... | 29 |
| APPENDIX B..... | 29 |

LIST OF FIGURES

| | |
|--|----|
| Figure 1. Ontological representation of the structure of the PKB KINASE DOMAIN (pdb id:1gzn) | 11 |
| Figure 2. Design Protocol..... | 12 |
| Figure 3. Protocol Implementation- Phase 1..... | 14 |
| Figure 4. Protocol Implementation - Phase 2..... | 14 |
| Figure 5. Case Study: Structural Alignment of two MMPs, namely MMP-7 (PDB ID: 1MMP) and MT1-MMP (PDB ID: 1BQQ)..... | 15 |
| Figure 6. Protocol Implementation - Phase 3..... | 17 |
| Figure 7. Protein Structure Superposition Protocol as entered in ProtocolDB..... | 17 |
| Figure 8. Semantic Map of BioMoby Services (partial view)..... | 24 |

LIST OF TABLES

| | |
|--|----|
| Table 1. Biomoby versus a Simple Web Service..... | 21 |
|--|----|

ABSTRACT

Modeling scientific problems in workflows involves a number of activities. At a very basic level scientists specify the design of the workflow and implement the design using different tools. Issues such as specifying an efficient design, resource discovery, interoperability of the selected resources plague the bioinformaticians while creating a workflow. Tools such as ProtocolDB [1] can assist the scientists in specifying and reasoning about their workflows. One such workflow related to the superposition of the protein structures was analyzed and implemented using the underlying workflow model of ProtocolDB.

The use of domain ontologies to specify the design of the protocol increases its visibility, understandability and reusability by defining it with well accepted standards. Once the concepts have been selected, the design is mapped onto the implementation protocol. In the implementation phase the selection of tools takes place which requires prior knowledge about what the tool does. In an ideal world all tools would be intelligently annotated and placed efficiently according to different concepts in the ontologies thus making them uniquely identifiable according to the tasks they can perform and the type of inputs and outputs they consume and produce respectively. Each tool would either understand all the relevant formats or would be able to convert the incoming data into the desired format without having the scientist to write shims and wrappers to do so. But in reality, the scientists comes across a number of problems during resource discovery and incompatibility between the resource formats that hinder the specification and execution of the workflow. Thus the decision to use a particular tool becomes dependant more on the format than the tools that can perform the task efficiently. In this report we analyze a well known life sciences repository called BioMoby [2] that tries to overcome such

resource discovery and interoperability issues by specifying a common service and data type ontology. We identify the absence of a suitable domain ontology in BioMoby and try to extract one using the existing ontologies.

GOAL OF THE PROJECT

The goal of our project is to alleviate the challenges met by bioinformaticians while performing tasks such as scientific modeling and resource discovery.

INTRODUCTION

Typically while conducting a research a scientist would access a number of tools to obtain the desired results. A single tool would be incapable of fulfilling all their scientific needs. The output of a particular tool has to be made available as an input for the next tool. Thus the scientist becomes involved in the repeated process which in essence resembles a workflow.

Scientific workflows go through a number of construction phases before providing the desired results. Initially the scientist specifies the scientific aim of the protocol using concepts derived from the relevant scientific domain. Once the design had been specified, tools that are capable of performing the design tasks are searched for and selected. After overcoming the dataflow inconsistencies, the workflow is executed.

In the field of life sciences information integration becomes a challenge due to the highly dynamic nature of the data. Therefore it is unrealistic to consider the current scientific model implemented in the workflow is going to persist for a long time. Even after the workflow has been fully implemented it goes through evolutionary changes based on results obtained or as its aim becomes clearer and more refined with new discoveries.

Finally, the newly created workflow would be of importance only if it produces meaningful data or results. It will be considered vital, if it adds value to the environment. Value comes from the increase in knowledge about the environment or the process itself. Thus the overall aim is not only the efficient generation of data but the knowledge that can be extracted.

Data can be turned into knowledge once their meaning is known. Thus we need what is called “meta data” to assist us in the path of knowledge discovery. Another important

aspect is the reusability of both data and the workflows producing the data. The myriad of tools now being made available is reducing the need for writing ones own scripts. A popular way of making resources available nowadays is converting them into web services. A web service model offers advanced automation and application integration [3]. A complete annotation process using the relevant ontologies is the only way that can make these tools and web services uniquely identifiable and efficiently exploitable.

In this report we present the work done in the area of scientific modeling and resource discovery. A workflow regarding the protein structures superpositioning was analyzed as a part of an internship completed at Translational Genomics (TGEN). Further more we explored the different aspects of a famous life sciences resource repository called BioMoby [2] in an effort to extract a domain ontology from its existing ontologies.

METHODOLOGY

SCIENTIFIC MODELING

In silico experiments carried out by the scientist make use of tools that take the shape of workflows.

A scientific protocol is constructed in two phases [4]

- design
- implementation

In the design phase tools such as ProtocolDB [1] can assist the scientist in expressing the scientific aim of their protocols at the conceptual level. A domain ontology is used to identify the scientific objects and tasks for the protocol.

The next phase is the implementation phase. It requires the selection of tools and data sources. These resources should map directly onto the scientific tasks and objects. Scientific protocols can be implemented using applications such as Taverna [5] or Kepler [6] that are specifically aimed at workflow implementation.

During an internship completed at TGEN a protein structure superposition protocol was expressed and analyzed using the underlying model for ProtocolDB.

The aim of the protocol is to superpose protein structures evaluated at the residue level, when both sequences and structures have diverged. The main superpositioning task in the protocol is performed by the GAPS (Gaussian Based Alignment of Protein Structures) algorithm. GAPS uses a Gaussian representation of selected atoms in the proteins and utilizes a 3-D molecular similarity function as a metric for quality of superposition.

The workflow was executed on two proteolytic enzyme structures namely Matrix Metalloproteinases (MMP) such as gelatinase A (MMP-7, PDB ID: 1MMP)² and

² <http://www.rcsb.org/pdb/explore/explore.do?structureId=1MMP>

membrane-type MMP-1 (MT1-MMP, PDB ID: 1BQQ)³. These enzymes regulate various cell behaviors with relevance for cancer biology. Overlaying their structures provides insights into their structurally-conserved regions, which are valuable for drug discovery projects in general.

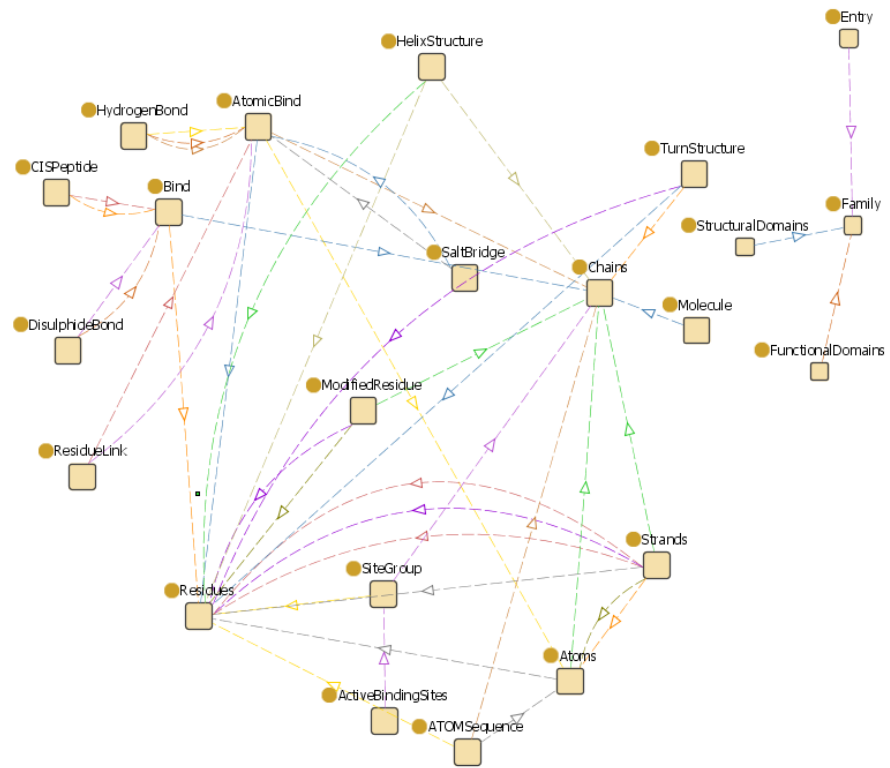


Figure 1. Ontological representation of the structure of the PKB KINASE DOMAIN (pdb id:1gzl)

To express the protocol we make use of the existing domain ontology for proteins i.e., Protein Ontology (PO) [7]. Figure 1 shows the structure of the *PKB KINASE DOMAIN* (PDB ID: 1gzl) downloaded from the Protein Ontology (PO) instance repository⁴ for various proteins and being displayed in Protégé⁵. The language used for annotation is OWL [8]. The figure shows the numerous elements bonds and chains that contribute to the overall structure of the protein. Deriving the concepts from the ontologies removes the need for specifying any kind of formats at this stage. In the implementation phase

³ <http://www.rcsb.org/pdb/explore/explore.do?structureId=1BQQ>

⁴ Protein Ontology ID:PO0000007412 instance can be found at the following site:
<http://proteinontology.info/proteins.php?currentPage=0&sortBy=pofilename&sortOrder=desc>

⁵ <http://protege.stanford.edu>

these concepts can be associated with different formats.

The different phases of protocol construction are shown in the Figure 2-4. As mentioned earlier the ProtocolDB workflow model was used to design and implement the protocols. To express the protocols, the model uses split and parallel operators to connect the different tasks to one another. Integrity constraints prevent any new inputs or outputs to be declared for the intermediate tasks. These inputs and outputs are expressed using the basic types used in the ProtocolDB data model.

Basic Type := Xc where, c is an ontology concept

Using this basic type complex types are made,

Set $\{Xc\}$

List (Xc)

Record $[Xc1, Xc2, \dots, Xcn]$

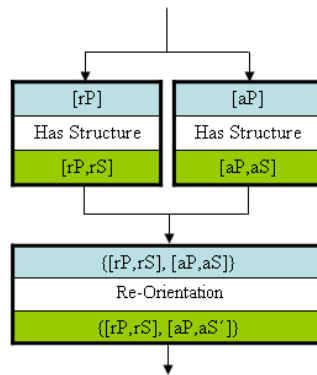


Figure 2. Design Protocol.

The design for our protocol shown in Figure 2 has three tasks, two parallel and one sequential. The parallel tasks retrieve the structures (aS , rS) of the adapting protein (aP) and the reference protein (rP). The sequential task reorients the aS with respect to rS thus

providing us with a final reoriented aS' .

In the implementation protocol the tools selected are (1) PDB [9] used as a resource of protein structural information and (2) the GAPS program [10], selected to compute the protein structure similarity and to generate molecular superpositions. Figure 3 presents the basic protocol implementation. The execution details are as follows.

Reference and Adapting Protein structures (aP , rP) are downloaded from PDB into a file directory. Both aP and rP are fed into a routine called ' α -carbon Extractor' that extracts the α -carbon chains of the protein structure (aS , rS) which are then passed onto the GAPS routine. The GAPS algorithm produces a set of multiple orientations of the adapting protein (aS') with respect to the reference protein. Next the best superposition extractor routine extracts aS' with the best orientation. A script written in Scientific Vector Language (SVL) that interfaces with the Molecular Operating Environment (MOE) platform (SVL and MOE are products of Chemical Computing Group, Montreal, Canada)⁶ converts the best orientation α -carbon structure of the adapting protein (aS') back into its original full protein form. Finally reference and adapting protein structures can be viewed superposed through a molecular viewer and further analyzed by the scientist.

In Figure 3 we can identify a number of format discrepancies as well as other missing connectors needed for the smooth execution of the protocol. We recognize a need for connectors to convert between PDB format and an internal format understood by the GAPS algorithm i.e., a CCC format. A complete picture of the protocols including all the missing connectors is presented in Figure 4.

⁶ <http://www.chemcomp.com/>

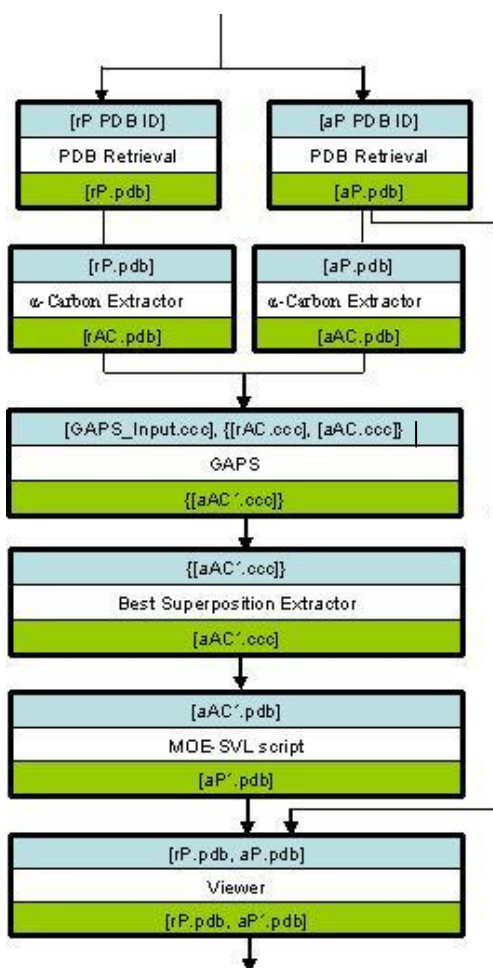


Figure 3. Protocol Implementation- Phase 1.

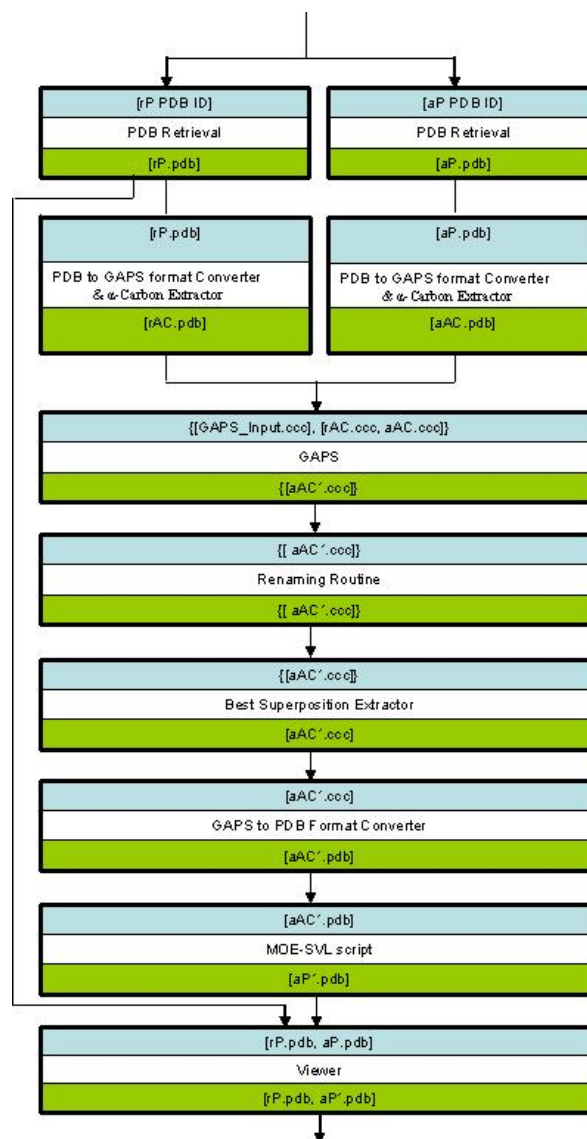


Figure 4. Protocol Implementation - Phase 2.

It took approximately 7 seconds⁷ for the protocols to execute on the two proteolytic enzymes structures namely Matrix Metalloproteinases (MMP) such as gelatinase A (MMP-7, PDB ID: 1MMP) and membrane-type MMP-1 (MT1-MMP, PDB ID: 1BQQ). and present us with the re-oriented adapting protein. 1MMP was selected to be the reference protein and 1BQQ was chosen as the adapting protein. Originally 1MMP

⁷ Processor model: Intel(R) Pentium(R) 4 CPU 3.20GHz

consists of two identical chains A and B and 1BQQ also consists of two identical chains M and T. In each case, duplicate chains point to the same protein sequence. Therefore only one chain per protein is necessary to pursue the analysis. After manual deduplication, the protocol is executed with the two files as inputs. The results are shown in Figure 5.

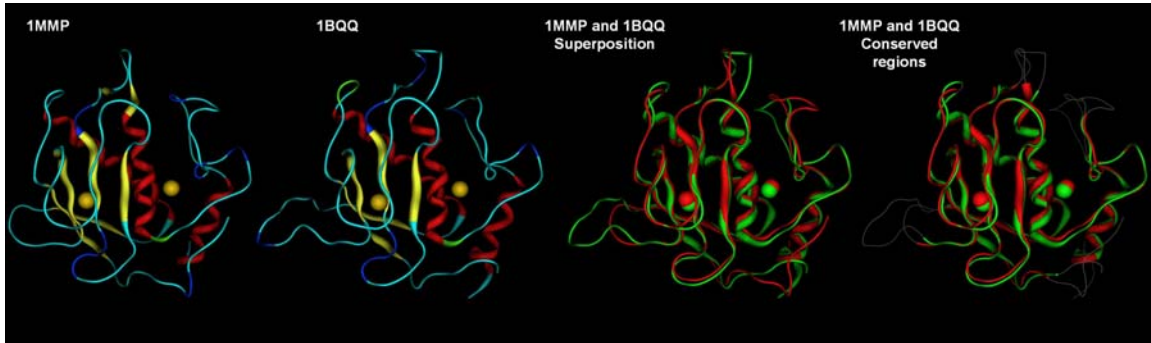


Figure 5. Case Study: Structural Alignment of two MMPs, namely MMP-7 (PDB ID: 1MMP) and MT1-MMP (PDB ID: 1BQQ). Structurally conserved regions correspond to residues with root mean square deviation of mainchain atoms below 1.0 \AA° .

1MMP and 1BQQ are close homologs thus their superposition their best superpositioning result is very accurate.

The performance of the protocol in Figure 4 is limited by a few factors. The limitations mostly regard the steps that remain manual and require some kind of human intervention. These include the retrieval of files from the PDB repository through its website and the cleaning of files to remove any excess chains of the protein structures that would not be involved in the superimposition process. If both steps were automated, the time taken and the effort required by the user to execute the protocol would be considerably reduced. Our analysis of the protocol in ProtocolDB allows the identification of steps that can be optimized.

A more sophisticated version of the protocol is presented in Figure 6. In this version we provide the user an interface to view all the orientations of the adapting protein. The

availability of a browsing function enables the scientist to select the best orientations of the adapting protein based on other considerations including structural ones. This improvement would suppress the need for the Best superposition extractor routine. Hence this empowers the scientist by allowing him to browse through all the possible superimpositions and select what he considers to be the best one instead of relying on the algorithm to extract one for him.

Figure 7 presents the Design phase and one of the implementation phases as entered in ProtocolDB.

A more detailed explanation and analysis of the protocols can be found in the research article attached in the Appendix B of this report.

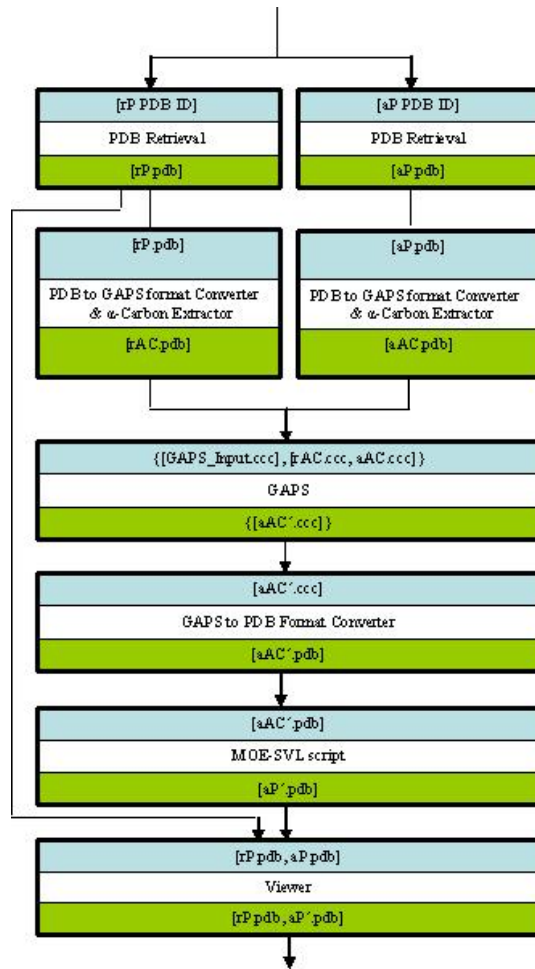


Figure 6. Protocol Implementation - Phase 3.

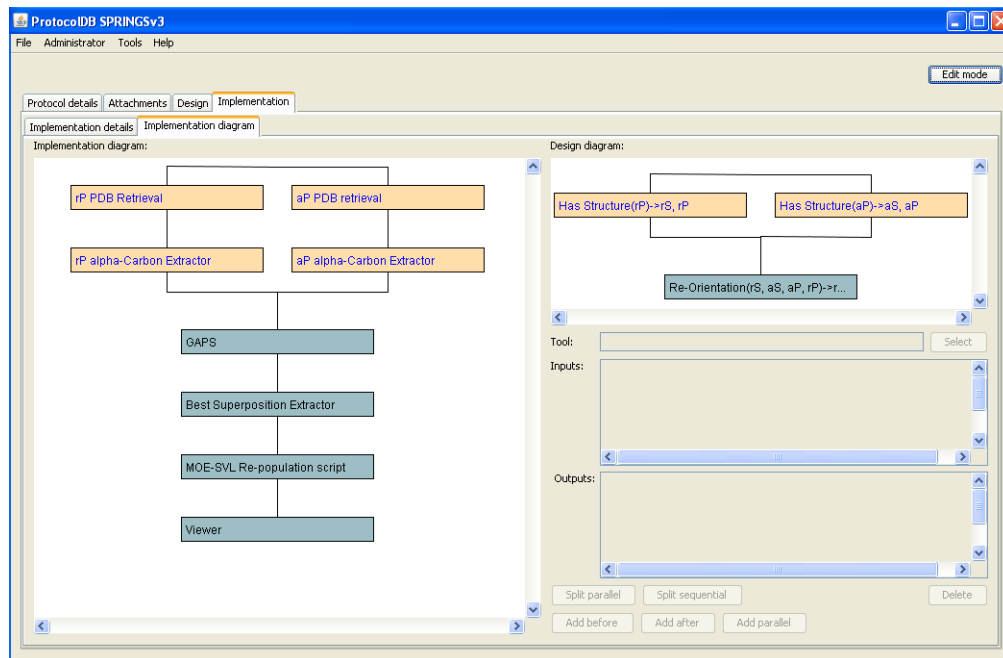


Figure 7. Protein Structure Superposition Protocol as entered in ProtocolDB.

RESOURCE DISCOVERY

When researchers and scientists decide to share their discoveries with the rest of the community or across communities they come across hindrances such as the procedure of disseminating information and making the resources visible to everyone, which is not a trivial task.

Even if they are able to overcome the first hurdle and finally do publish their resource, there still exist problems such the inability to locate and discover the resource. Once discovered basic questions such as how to actually use and communicate with the resource, how to integrate it in different applications and how to handle the data formats of the inputs and outputs arise.

Mainly the search for resources is concentrated in the broad areas of access to databases, retrieval task and analysis. While searching for a particular resource the scientists are either equipped with the knowledge of what resources they are interested in using or are clueless for the most part about the resources. The latter scenario is quite common with the abundance of different types of resources being made available either on the web or privately. Therefore the scientists require robust resource discovery applications to assist them in identifying the resources suitable to implement their scientific tasks.

Resource discovery is facilitated through data centric means such as conforming to specialized formats e.g. FASTA thus enabling discovery on the basis of formats. It is also made possible by having repositories such as the BioMoby central or Grimoires [11] that provide a platform for resource registration thus discovery.

Web services are discoverable through yellow pages or a UDDI. BioMoby provides one such repository which can be searched for a desired web service in life sciences.

BioMoby is an excellent effort at the syntactic and up to a certain extent the semantic levels. However what is missing is a conceptual layer on top of frameworks such as BioMoby that would provide scientific meaning to the resource. This can be achieved through the implementation of a controlled ontology which can be accessed by the resource developers as well as resource users which includes both humans and programs. Thus resource discovery stands facilitated.

We decided to work with the BioMoby ontologies and make an effort to extract semantics thus knowledge from an existing domain instead of starting an ontology from scratch. Furthermore BioMoby repository has generated a lot of interest in the field of life sciences and is growing quite rapidly with time. We focus on the analysis of the BioMoby repository that contains 1,611 services registered in BioMoby Central by 84 different authorities as of April 2008, a dramatic increase since the 2000 release with 140 resources registered [5].

In our efforts we would like to create a repository for the ProtocolDB application that includes services in BioMoby. However, ProtocolDB is semantics-oriented thus relies on the scientific concepts that describe the resources which is found missing in BioMoby. Once the semantics are in place, we hope to be able to download and store web services from repositories such as BioMoby at regular interval of time.

Our aim is to not only perform tasks such as discovering resources based on biological data types they accept or output by exploring the data type ontology like Taverna but also perform searches on the scientific thought level by exploring the conceptual ontology created through tools like SemanticMap [12, 13].

Therefore we see a need for classification of the resources according to the purpose they achieve.

Increasing numbers of resource providers are making their tools available on the web in the form of web services.

A simple web service is described in Web Service Definition Language (WSDL) [14]. The WSDL file is a simple XML file that contains information regarding the operations that the web service can perform as well as their inputs and outputs. As the awareness and need for semantics increased the community decided to utilize constructs such as ontologies and structured meta data while describing a web resources. A number of formats such a OWL-S [15, 16], WSDL-S [17] and SAWSDL [18] to name a few, were used to integrate semantics while describing a web service.

BioMoby is an effort that builds on existing constructs and works towards providing a platform that can be used to make the resource visible and interoperable with other resources. It accomplishes this by providing a datatype as well as a service type ontology to the user. A comparison between a simple web service and a BioMoby web service is provided in Table 1.

A detailed overview and analysis of the ontologies in the BioMoby domain can be found in the paper in Appendix A.

Table 1. Biomoby versus a Simple Web Service

| | Simple Web Service | BioMoby Web Service |
|------------------------------------|---|--|
| Service Registry | UDDI (Universal Description, Discovery and Integration) | Moby Central |
| Inside the Service Registry | In UDDI the web services do not conform to any one standard i.e. Multiple services are made according to the logics of the people who made them | In BioMoby Central standardized interface is made available by all the services |
| Types of Service Registries | A business can deploy its own UDDI for external/internal services [19] <ul style="list-style-type: none"> • Internal registry – for services on intranet • External registry – for services on internet | Here too one can install his own BioMoby central and run his own registry [23] |
| Architecture | No specific architecture | A specific architecture provided by BioMoby in which you have to fit in your service |
| Platform Restriction | Any platform or language can be used to develop a web service as long as that platform provides the functionality to do so. | Only specific languages can be used namely java, Perl and python [24] of which working in java environment has the most documentation. The rest lack documentation. |
| Need For Special Softwares | No specific softwares need to be downloaded or configured to make your web service. The platform on which you develop you web service is sufficient. | One of the three softwares has to be downloaded and configured [24] <ul style="list-style-type: none"> • Jmoby from Moby CVS repository • Moby-s Perl Libraries • Python |
| Steps For developing a Web Service | Design → Implement Business logic → Test → deployment → Register in UDDI | For Java Map/register Data Structure → select/register service type → select/register Namespace → register service → implement business logic in the generated skeleton → Test → deployment For Perl Map/register Data Structure → select/register service type → select/register Namespace → implement business logic in the generated skeleton → Test → deployment → register service |
| Communication | Client and web service talk through XML messages. | Client and web service talk through XML messages |
| Datatype Definition Format | Primitive datatypes e.g. string, integer etc | In BioMoby precise syntax of a datatype is defined by an ontology in an xml format [20] |
| Type of Datatype | Primitive datatypes e.g. string , integer etc | In BioMoby a repository of biological datatypes exists. Data structure mapping i.e. class mapping has to be done for each variable. New datatypes can be declared by the person creating the service if what he is |

| | | |
|--------------------------------|---|--|
| | | looking for is not present in the repository. |
| Datatype of Variables | Inputs and outputs are of primitive type | Inputs and outputs are of specialized types. |
| Namespace | none | A special namespace has to be associated to the service one wants to declare [21] |
| Special Data Structure | None | A special data structure of the input and output for the service one wants to declare [22] (class, namespace, identifier) |
| Syntax and Semantic Separation | Data represented in one format | Syntax is kept separate from semantics. In this way the same data can be represented in a variety of ways when going from one too to another [25] |
| Service Declaration Type | No special service type needs to be declared | A special service type needs to be associated with the service. If not present a new service type can be declared . |
| Metadata | Metadata can be attached about the service in the WSDL. Messages are described by parameter names and datatypes. Therefore a simple web service fails to achieve semantic interoperability [26] | Metadata can be attached to the service at the time of registering the service . |
| Target Users | Anyone | Bench Scientists as well as bioinformaticians |
| Clients | No one specific client | Moby Service Clients [24] <ul style="list-style-type: none"> • Gbrowse moby • Taverna • MOWserv • Remora • Ahab • Seahawk Clients with embedded MOBY functionality [24] <ul style="list-style-type: none"> • BioTrawler • BlueJay • BioFloWeb • AtiDB Client |

As mentioned earlier, the main aim of this effort was to extract a domain ontology from the existing BioMoby ontologies. The details of the techniques adopted, the algorithm used and the output statistics can be found in the paper in Appendix A.

The algorithm was run against 1,611 services, a service classification tree of depth 5 containing 115 nodes, and a datatype classification tree of depth 8 containing 613 datatypes. Running time was approximately 40 seconds⁸.

⁸ Processor model: Intel(R) Core(TM)2 CPU 2.13GHz, 2GB of RAM

Our efforts led to the generation of a first version of the SemanticMap of the resources in BioMoby. This version has its short comings which shall be addressed in the future versions of the map. Figure 8 shows a subset of the generated SemanticMap. The boxes in Figure 8 represent the services and the oval are the concepts. In the map presented in Figure 8 the concepts are not necessarily the actual inputs and outputs of the service but rather the more abstract (super class) data type of the hierarchy. We can see that most of the concepts generated are formats rather than scientific concepts. This is due to the orthogonal motivations of what we are trying to achieve and what BioMoby has to offer. Resource discovery is driven by scientific concepts whereas resource composition efforts concentrate more on the format rather than concepts. BioMoby is inclined more towards resource composition rather than discovery thus its ontologies depict the main driving motivation.

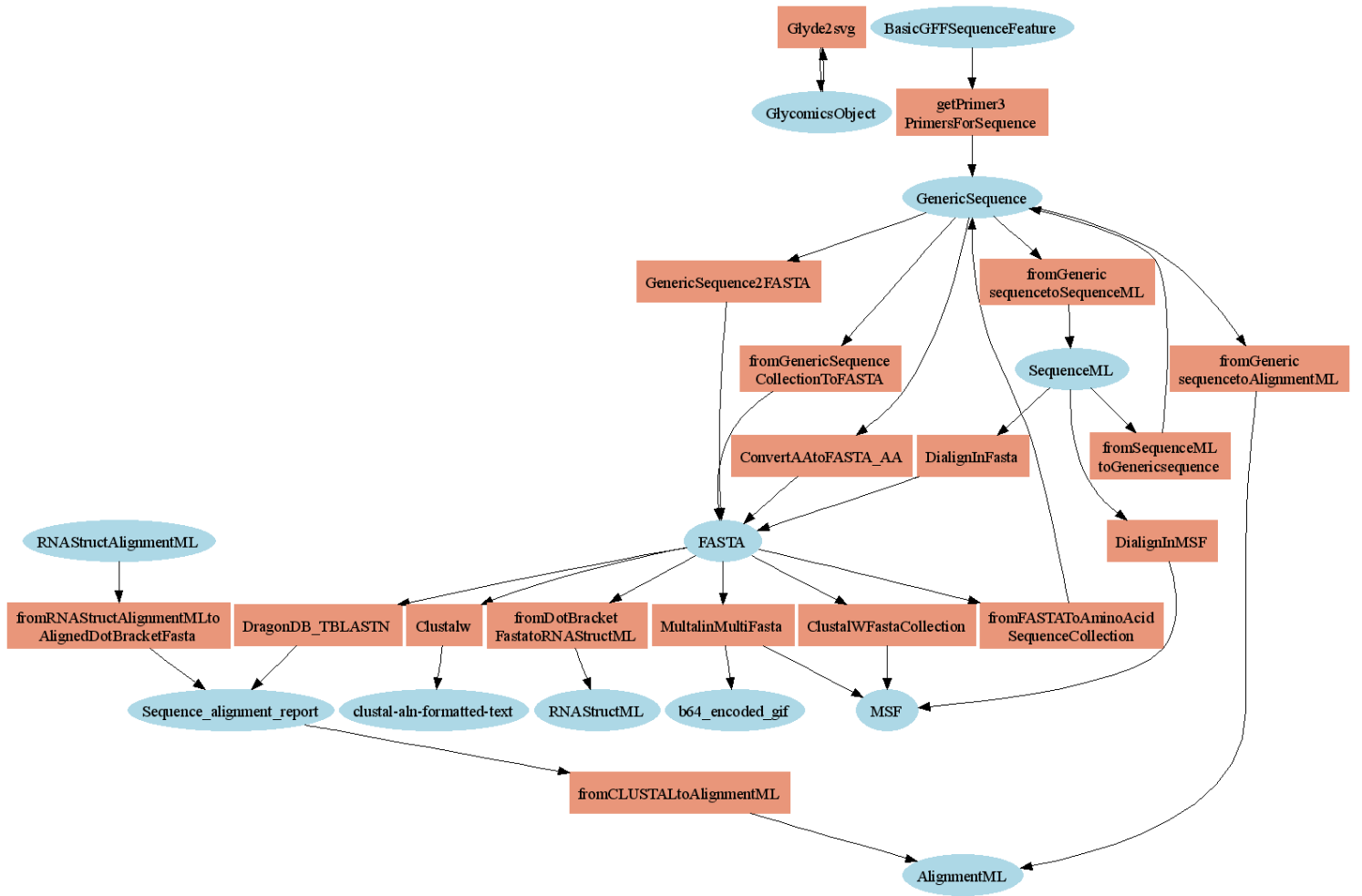


Figure 8. Semantic Map of BioMoby Services (partial view)

CONCLUSIONS AND FUTURE DIRECTIONS

In this report we explored two related scenarios i.e., Scientific modeling and Resource Discovery that are quite common in the field of bioinformatics. We described the implementation and analysis of a protein superposition protocol using a domain ontology and the ProtocolDB workflow model. At the moment, a team is working towards a new more modern version of the GAPS algorithm. Once that is achieved we hope to fix the remaining shortcomings of the workflow and come up with an optimized implementation. This workflow could then serve as an algebraic operator called '*superposition*' in different related workflows while defining a conceptual task.

Next we moved on to shed some light on the challenges being faced by scientists during resource discovery. We analyzed a famous life sciences repository for web services called BioMoby and discussed our approach for extracting a domain ontology from its existing ontologies. A manual curation of the web services in the repository along with their inputs and outputs into a conceptual graph would require a lot time and effort. In this day and age where computational capabilities have evolved many folds, performing a manual curation should not be an option at all. A semi-automatic procedure is a more acceptable approach. Therefore we developed an algorithm and presented our results. The algorithm and the results are still in their preliminary stages. As a future effort the algorithm has to be enhanced by exploring the nature of services even further. We identified a number of inconsistencies in the ontologies as well as the misclassification of services at registration. Therefore we would need to verify if all the services extracted based on the service type ontology even the ones that we have included in our map, do actually belong to that service type category.

At the moment the algorithm is faced by a number of challenges such as generation too

many concepts that do not have any subclasses and overcoming the orthogonal motivation or resource discovery and composition. We would be coming up with rules and axioms that would traverse multiple ontologies in addition to the ones in BioMoby which if satisfied would allow a particular data type to be considered as a concept. These rules would help decrease the level of curation and allow expansion of the conceptual mapping at certain places where the meaning is being lost.

REFERENCES

- [1] Michel Kinsy, Zoé Lacroix, Christophe Legendre, Piotr Wlodarczyk, "ProtocolDB: Storing scientific protocols with a domain ontology," in *International Workshop on Web Data Integration and Mining in Life; Lecture Notes in Computer Science*, 2007, pp. 17-28.
- [2] The BioMoby Consortium, "Interoperability with Moby 1.0--It's better than sharing your toothbrush!" *Brief Bioinform*, pp. bbn003, 2008.
- [3] R. de Knikker, Y. Guo, J. Li, A. Kwan, K. Yip, D. Cheung and K. Cheung, "A web services choreography scenario for interoperating bioinformatics applications," *BMC Bioinformatics*, vol. 5, pp. 25, 2004.
- [4] Natalia Kwasnikowska, Yi Chen, Zoé Lacroix: Modeling and storing scientific protocols. In: OTM Workshops (1). (2006) 730–739
- [5] Tom Oinn, Mark Greenwood, Matthew Addis, M. Nedim Alpdemir, Justin Ferris, Kevin Glover, Carole Goble, Antoon Goderis, Duncan Hull, Darren Marvin, Peter Li, Phillip Lord, Matthew R. Pocock, Martin Senger, Robert Stevens, Anil Wipat, Chris Wroe, "Taverna: lessons in creating a workflow environment for the life sciences," *Concurrency and Computation: Practice and Experience*, vol. 18, pp. 1067-1100, 2006.
- [6] Bertram Ludäscher, Ilkay Altintas, Chad Berkley, Dan Higgins, Efrat Jaeger, Matthew Jones, Edward A. Lee, Jing Tao, Yang Zhao, "Scientific workflow management and the Kepler system," *Concurrency and Computation: Practice and Experience*, vol. 18, pp. 1039-1065, 2006.
- [7] Amandeep S. Sidhu, Tharam S. Dillon, Elizabeth Chang, Baldev S. Sidhu, "Protein ontology: Vocabulary for protein data," in *Third International Conference on Information Technology and Applications (ICITA'05)*, 4-7 July 2005, pp. 465-469 o.1.
- [8] Sean Bechhofer, Frank van Harmelen, Jim Hendler, Ian Horrocks, Deborah L. McGuinness, Peter F. Patel-Schneider, Lynn Andrea Stein. OWL web ontology language reference.
- [9] Berman, H.M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T.N., Weissig, H., Shindyalov, I.N., Bourne, P.E., "The Protein Data Bank," *Nucleic Acids Res.*, vol. 28, pp. 235-242, 2000.
- [10] G. Maggiora, D. Rohrer, and J. Mestres,, "Gaussian-based alignment of protein structures: deriving a consensus," *Journal of Molecular Modeling*, vol. 6, pp. 539-549, 2000.
- [11] Miles, Simon and Papay, Juri and Payne, Terry and Luck, Michael and Moreau, Luc, "Towards a protocol for the attachment of metadata to grid service descriptions and its use in semantic discovery." *Scientific Programming*, vol. 12, pp. p201, 20040101.

- [12] Pierre Tufféry and Zoé Lacroix and Hervé Ménager, "Semantic map of services for structural bioinformatics," in *International Workshop on Database Interoperability (InterDB)*, 2006,
- [13] Hervé Ménager, Zoé Lacroix, Pierre Tufféry, "Bioinformatics Services Discovery Using Ontology Classification," pp. 106-113, 2007.
- [14] Erik Christensen and Francisco Curbera and Greg Meredith and Sanjiva Weerawarana, "Web Services Description Language (WSDL) 1.1," 2001.
- [15] David L. Martin and Massimo Paolucci and Sheila A. McIlraith and Mark H. Burstein and Drew V. McDermott and Deborah L. McGuinness and Bijan Parsia and Terry R. Payne and Marta Sabou and Monika Solanki and Naveen Srinivasan and Katia P. Sycara, "Bringing semantics to web services: The OWL-S approach," in *SWSWPC*, 2004, pp. 26-42.
- [16] OWL-S Coalition, "OWL-S 1.1 Release," [Online]. Available: <http://www.daml.org/services/owl-s/1.1/>
- [17] R. Akkiraju and J. Farrell and J. Miller and M. Nagarajan and M. Schmidt and A. Sheth and K. Verma, "Web service semantics - WSDL-S," IBM, University of Georgia, 2005.
- [18] Kopecký, Jacek and Vitvar, Tomas and Bournez, Carine and Farrell, Joel, "SAWSDL: Semantic Annotations for WSDL and XML Schema," *Internet Computing, IEEE*, vol. 11, pp. 60-67, Nov.-Dec. 2007.
- [19] Universal Description, Discovery and Integration, <http://www.uddi.org/faqs.html>
- [20] Biomoby Object Ontology, <http://biomoby.org/RESOURCES/MOBY-S/Objects>
- [21] Biomoby Namespace Ontology, <http://biomoby.org/RESOURCES/MOBY-S/Namespaces>
- [22] Biomoby Service Type Ontology, <http://biomoby.org/RESOURCES/MOBY-S/Services>
- [23] Installing a local BioMOBY Central, http://biomoby.open-bio.org/CVS/CONTENT/moby-live/Docs/MOBY-S_API/InstallingLocalMOBYCentral.html
- [24] Biomoby For Developers, <http://biomoby.open-bio.org/index.php/for-developers/>
- [25] M. Wilkinson, H. Schoof, R. Ernst and D. Haase, "BioMOBY successfully integrates distributed heterogeneous bioinformatics," *Plant Physiol.*, vol. 138, pp. 5-17, May. 2005.
- [26] U. Radetzki, U. Leser, S. C. Schulze-Rauschenbach, J. Zimmermann, J. Lussem, T. Bode and A. B. Cremers, "Adapters, shims, and glue--service interoperability for in silico experiments," *Bioinformatics*, vol. 22, pp. 1137-1143, May 1. 2006.

APPENDIX

Please refer the attached papers with this document.

APPENDIX A – Maliha Aziz, Zoé Lacroix, Nathalie Meurice "Classifying BioMoby Services in a Domain Ontology to Support Resource Discovery"

APPENDIX B – Maliha Aziz, Zoé Lacroix, Hervé Ménager, Pierre Tufféry, "Analysis and Implementation of a Protein Structure Superposition Protocol"