Bioinformatics Application Integration and Management in GeneGrid: Experiments and Experiences

P.V.Jithesh, Noel Kelly, David R. Simpson, Paul Donachy, Terence J. Harmer, Ron H. Perrott

Jim Johnston, Mark McCurley, Paul Kerr

Shane McKee

Belfast e-Science Centre
www.qub.ac.uk/escience

Fusion Antibodies Ltd
www.fusionantibodies.com

Amtec Medical Ltd
www.amtec-medical.com

Abstract

GeneGrid is a collaborative industrial R&D project initiated by the Belfast e-Science Centre, under the UK e-Science Programme, with commercial partners involved in antibody and drug research and development. It aims to provide a platform for scientists, especially biologists, to access collective skills, experiences and results in a secure, reliable and scalable manner through the creation of a 'Virtual Bioinformatics Laboratory' that will eventually expedite the discovery of new drugs targeted against cancer and infectious diseases. Since Bioinformatics applications are quite heterogeneous in their requirements, formats of input data, parameters, nature of outputs and platform binding, application integration and management forms a key area of the project. The GeneGrid Application Manager (GAM) component integrates diverse Bioinformatics applications on various resources with the rest of the environment through OGSI-compliant grid services developed based on the Globus Toolkit version 3. This paper presents the architecture, functionality and interactions of GAM and the implementation details and experiences of a functional prototype.

1. Introduction

The requirements of storage and the analysis of the large volume of data generated by genome sequencing projects as well as post-genomic technologies such as microarrays have pushed Bioinformatics to the forefront of disciplines that need huge computing power. However, the prohibitive cost of High Performance Computing systems have made the large scale processing of the data out of reach of scientists with modest computing equipment. The emergence of grid computing [1] technologies has opened up an unprecedented opportunity for biologists to have data from multiple resources, in spatially distant locations which can be integrated seamlessly for comprehensive analysis leading to a greater chance of knowledge discovery.

There has been a few initiatives within the UK e-Science community [2,3] and worldwide [4] to address the Bioinformatics problems using grid computing principles. GeneGrid differs from much of these projects in its motivation and orientation. It is a collaborative industrial R&D project initiated by the Belfast e-Science Centre, under the UK e-Science Programme, with partner companies involved in antibody and drug research and development, namely, Fusion Antibodies Ltd and Amtec Medical Ltd. The background and motivation for this project are discussed in detail in [5].

The principal aim of GeneGrid is to provide a platform for the scientists and biologists from these companies to access the collective skills, experiences and results in a secure, reliable and scalable manner through the creation of a 'Virtual Bioinformatics Laboratory'. Harnessing

the avalanche of data from both public and private repositories in such an environment can help in the discovery of new drugs targeted against cancer and infectious diseases.

GeneGrid has three major areas of functionality, which work in a co-ordinated manner. These are the Workflow & Process Management, Data Management and Application & Resource Management. This paper focuses on the last of the three components. However, a brief description of the other two components, in the context of the whole architecture, will be presented in the next section.

2. GeneGrid: Architecture

GeneGrid is a database-managed, workflow-driven and application-oriented project based on the Open Grid Services Architecture (OGSA) model [6]. It provides the functionality through a number of co-operating OGSI-compliant grid services [7] interacting in a predefined order.

There are a number of components that make up the core of GeneGrid (Figure 1). Users can access the GeneGrid through a portal known as the GeneGrid Environment Interface. This interface masks the underlying complexity of GeneGrid components from the scientists while allowing them to submit and query workflows of their interest. A workflow in GeneGrid, for example, may involve tasks to search a nucleotide database for homologues using a sequence analysis program like Basic Local Alignment Search Tool (BLAST) [8], followed by formatting the results to translate the best scoring result nucleotide sequence to amino acid sequence and then submitting this sequence as input for a transmembrane prediction program.

In GeneGrid, the user-defined workflows are created based on a template Master Workflow present in a Workflow Definition Database. Once created, the workflow is then processed by the GeneGrid Workflow and Process Manager (GWPM), which also updates the status of the workflow in a Workflow Status Database. The Workflow Manager Service handles the workflow by breaking it into a number of tasks. Each of these tasks is further processed by a GeneGrid Process Manager (GPM) service. GPM is responsible for finding the appropriate resource to execute the task in hand. Discussion on the details of the architecture and functionality of the GeneGrid Workflow and Process Manager is beyond the scope of this paper and is described in detail in [9]. Briefly, the GPM after finding a suitable resource, contacts the GeneGrid Application Manager (GAM) Factory on that resource and creates an instance of the GAM service, which in turn executes the task. Detail of how this is achieved is discussed in the later sections.

Databases like the Workflow Definition Database, Workflow Status Database, Result Database etc. are managed by the GeneGrid Database Manager (GDM). GDM also integrates a number of publicly available biological databases like SWISSPROT, EMBL etc. along with the private databases of the partner companies. How this tedious task of integrating databases in diverse formats is achieved using OGSA-DAI [10] and its modifications is described in [11].

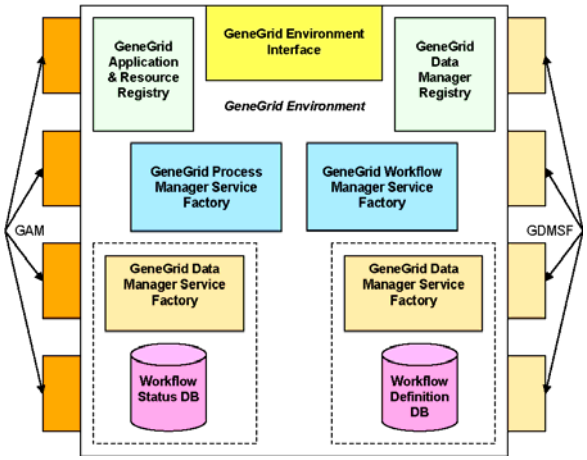

Figure 1. Overview of the GeneGrid architecture. The GeneGrid Environment, which is specific to a GeneGrid implementation, consists of a number of components including the Application & Resource Registry, Data Manager Registry, Workflow & Process Management components, Data Manager components etc. Application Manager and Data Manager Factory can be shared across different GeneGrid implementations

As mentioned in the previous paragraphs, there are a number of services that interact to provide the functionality of GeneGrid. Two centralised registries help in publishing and hence

discovering of these services. The Workflow Manager, Process Manager and Application Manager Factories register with the GeneGrid Application & Resource Registry (GARR) on startup. Other services involved in database management register with the GeneGrid Data Manager Registry (GDMR).

It is possible to have multiple implementations of GeneGrid and also to share the applications and resources in a GeneGrid implementation with another GeneGrid implementation physically located elsewhere. This can be achieved by registering the GAM and GDM Factory services which interface the applications and public databases in the first implementation with the GARR and GDMR respectively of the second one.

3. Application Integration in GeneGrid

GeneGrid provides access to various Bioinformatics applications available on different resources in an integrated environment. The component in GeneGrid which is responsible for this integration is 'GeneGrid Application Manager' (GAM). GAM achieves this functionality through two types of OGSA-compliant grid services: GeneGrid Application Manager Factory Service (GAMFS) and the GeneGrid Application Manager Service (GAMS) (Figure 2).

The factory service is a persistent service, which comes up along with the container which hosts the service. There is usually a single such factory service on a resource which interfaces all the applications available on that resource. The OGSI-compliant GAMFS extends the standard interfaces or portTypes like GridService and Factory [7]. A factory, in OGSI terms, is an abstract concept, corresponding to a grid service instance that is used by a client to create another grid service instance. As with the standard factory services, a client can invoke a create operation on the GAMFS and receive back a locator for the newly created service instance in response. Such services created by the GAMFS are called GeneGrid Application Manager Services (GAMS).

The GAMS, like any other OGSI-compliant grid service, implements the mandatory GridService portType, which forms the base interface definition in OGSI. The GAMS has a finite lifetime as opposed to the Factory service and hence is referred as a transient service. In addition to the standard operations provided by the GridService portType, GAMS exposes an operation specific to the GeneGrid, i.e. to execute an application on the resource. Each of the GAMS is specific to an application request by the client which invokes the factory to create the service. The GAMS is responsible for receiving the request and processing the input parameters and other data to generate the actual command and execute it on the resource. These inputs are transferred from the client service to the GAMS in the form of an XML document through Simple Object Access Protocol (SOAP).

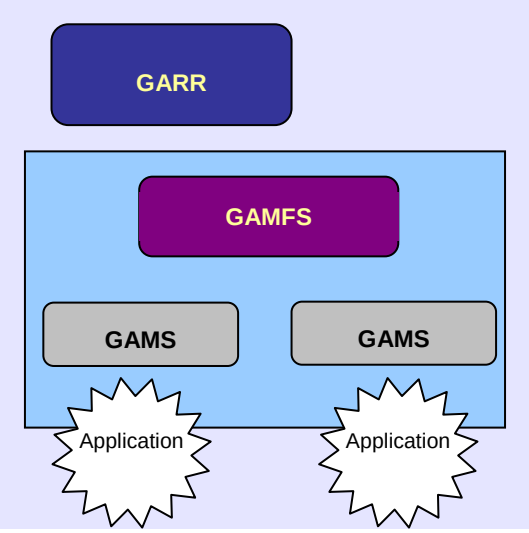

Figure 2. Overview of the GeneGrid Application Manager components. The GAM Factory Service (GAMFS) is registered with the GeneGrid Application & Resource Registry (GARR), which may reside on another resource. GAMFS creates transient GAM services (GAMS) which in turn access specific applications on the resource.

The GAM factory services on various resources publish their availability to other services or clients by registering with a centralised registry known as the GeneGrid Application & Resources Registry (GARR). Services which need to access an application will query this registry to get the Grid Service Handle (GSH) of the appropriate GAM Factory Service. Figure 2 gives an overview of the components that provide the GAM functionality.

3.1 Interaction with the GeneGrid Process Manager

The GeneGrid Process Manager (GPM) service is responsible for the dispatch of tasks to the GAM. See Section 2 for more information about GPM and other components of GeneGrid and how they interact. Each GPM service is responsible for the invocation of the GAM Factory Service for creation of a new GAM service. Therefore each task is handled by a different GAM service.

Initially, once the GPM service instance has been created for the processing of a specific task, it queries the GARR for finding suitable resources where the requested application is present (See Section 4 for more details). Once a resource is identified, the GPM service retrieves the GSH for the GAM Factory Service on that resource from the GARR. This allows the GPM service to communicate with the selected GAMFS and create an instance of the GAM service. The parameters for the application and other data like the unique task identification number, time of creation of the task etc. are communicated to the GAMS by the GPM in the form of an XML file. This XML follows a predefined schema, which helps in the proper validation and parsing at the receiving end.

The GAMS instance parses the task XML file and generates the appropriate command to execute on the resource. A configuration file present on the resource provides GAM with the details of applications, like the name of the executable, actual path on the resource, availability of a scheduler etc. Reading of such a file helps not only in making GAM more generic but also handling of new applications which are added to the resource quite easy.

The GAMS instance sends a notification XML file to the calling GPM service once the job is finished or in the case of a failure. This XML contains the status of the job and any reason for failure if it had failed. The GPM service communicates this information to the GeneGrid Workflow Manager (GWM), which is responsible for updating the Workflow status database with the details of the task, which in turn can be queried by the user to get the status of his job. Events that occur during the processing of a workflow through the interaction of GPM with GAM are depicted in Figure 3.

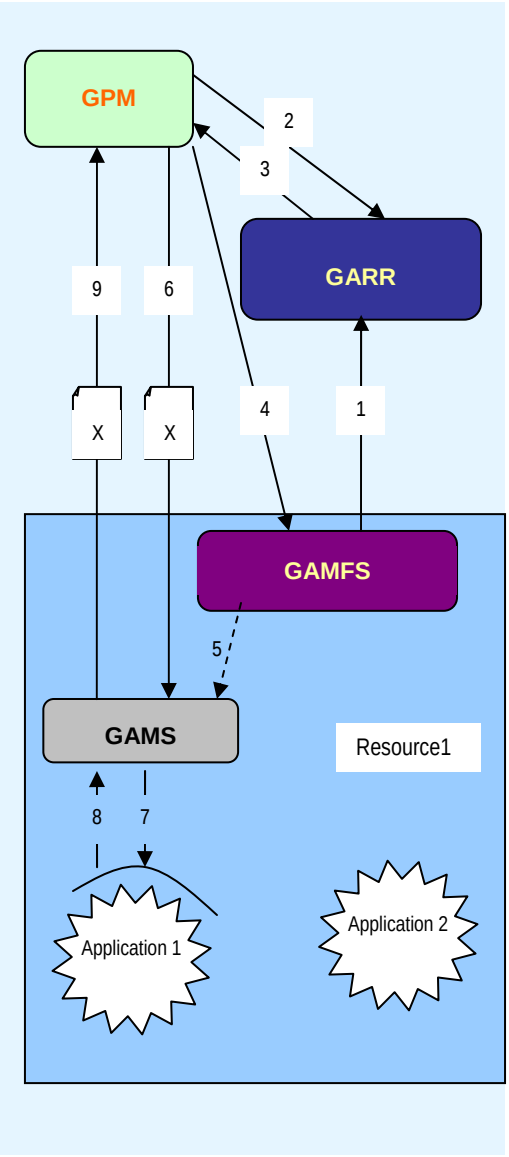

Figure 3. Interaction of GeneGrid Process Manager (GPM) service with GeneGrid Application Manager (GAM) components. 1. The GAM Factory Service (GAMFS) registers with the GeneGrid Application & Resource Registry (GARR) during startup. 2. GPM requests the GARR for GSH of a GAMFS 3. GARR returns the GSH of GAMFS 4. GPM requests GAMFS for creation of an instance 5. GAMFS creates a GAMS instance specific to the task 6. GPM contacts the GAMS with the task XML file 7. GAMS executes the application with parameters from task XML 8. GAMS receives the status of the job 9. GAMS sends a notification XML to GPM service which invoked it

4. Application and Resource Management

The main objective of the Application and Resource Management component of GeneGrid is to provide the client services with up-to-date information about the availability of applications on various resources and the status of the resources.

All the resources which are available to GeneGrid will have a lightweight agent residing on them. These agents are responsible for monitoring the resources and communicating the status of the resource to one or more node monitoring services. The agents update the resource status such as the load average in a timely manner. The node monitoring services in turn will be registered with the GeneGrid Application & Resources Registry (GARR) to make the information available to other client services. So the GeneGrid Process Manager (GPM), which is the GeneGrid service that usually requires information about resources and application availability, will contact the GARR to find out the GSH of a node monitoring service and then contact the selected service. The node monitoring service will provide the resource and application status of all the GeneGrid resources at that instance and also the GSH of the corresponding GeneGrid Application Manager (GAM) Factory service. The GPM then decides on the resource suitable for dispatching the task in hand and directly contacts the GAM Factory service on that resource. GPM then creates an instance of the GAM service (GAMS) by requesting the GAM Factory service with the input parameters and other data.

5. GeneGrid Testbed

The GeneGrid testbed was developed at the Belfast eScience Centre using Globus Toolkit 3 [12], which provided the base for development and initial deployment of OGSI-compliant grid services with the required functionality. GAM implementation was coded in Java while the interfaces that expose the functionality of GAM to other services were defined in GWSDL. A Web Service Deployment Descriptor (WSDD) file described the deployment parameters for the service. Finally, the services were deployed in Tomcat [13] container using the Java-based build tool, Ant [14].

The GAMFS available on testbed were capable of creating GAMS instances upon request by external services like GPM. The GAMS instances interfaced with Bioinformatics applications on the resource such as the sequence analysis program, BLAST. The actual application programs were not modified in anyway. The implementation of the service took care of the differing formats of commands to be executed for the different applications. To make the implementation more generic and flexible for the addition of new applications as they become available, most of the application specific details like the executable name and location on the resource were read in from a configuration file residing on the resource. This configuration file also provided information about the availability of scheduler on the resource. The execution command for the requested application was constructed based on the information read from the configuration file. Other parameters for the command, which were specified by the user, were parsed out from the task XML passed on to GAMS by the GPM. These parameters were then added to the above to generate the complete execution command. The final command was executed on the resource and outputs from the terminal or error messages were recorded by the GAMS. The job status information such as success or failure was communicated back in a notification XML file to the GPM which initiated the GAMS (Figure 4).

Various components of GeneGrid, such as the Data Manager, Workflow & Process Manager and Application Manager were deployed on different resources for test purposes. Highly heterogeneous resources ranging from single processor Pentium machines to six-processor Sun SMP machine with Solaris operating system to 32-node Linux cluster form the GeneGrid testbed. These resources are distributed across various administrative domains including the Belfast eScience Centre, Northern Ireland Technology Centre and other international sites.

GAM integrates different types of Bioinformatics applications. Following is a non-exhaustive list of the categories of applications:

- Genome sequence analysis: To search genome sequences against biological sequence databases for identifying homologues.

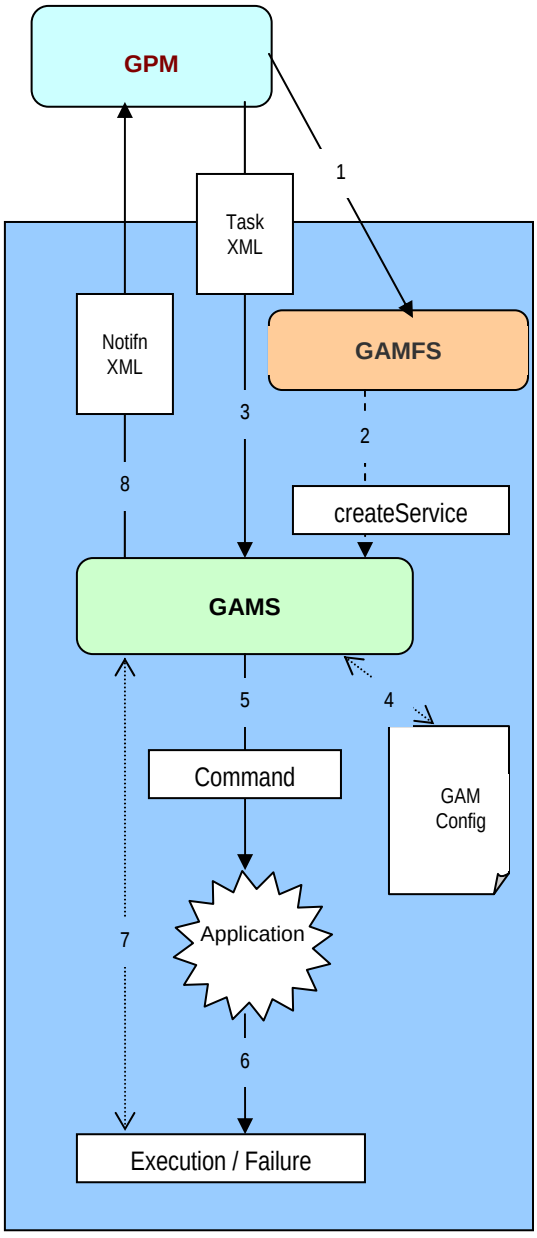

Figure 4. Steps involved in executing a task by the GeneGrid Application Manager (GAM). 1. GeneGrid Process Manager (GPM) service requests GAM Factory Service (GAMFS) for creation of a service instance 2. GAMFS creates a GAMS instance specific to the task 3. GPM sends the task XML containing parameters to the GAMS 4. GAMS reads a configuration file for application details 5. GAMS creates the command to execute from task XML and configuration file and invokes the application 6. Application is executed successfully or fails 7. GAMS receives the status of the job 8. GAMS sends a notification XML to GPM service which invoked it.

- Transmembrane prediction: To find out whether the amino acid sequence of interest contains domains specific for proteins spanning the length of the cell membrane.

- Signal peptide prediction: To find out whether the peptide sequence of interest is part of a secretory protein, by searching for amino acids that function as signal sequence in transporting newly formed secretory proteins to their final destination.
- Protein function prediction: To identify the function of an unknown protein by structural and sequence similarity.

A number of simple test cases were executed using the prototype successfully. One of the case scenarios is briefly presented here.

5.1 Identification of new protein family members

Identifying proteins which can act as suitable targets for the action of drugs is an important stage in the discovery of new drugs. Proteins which are located on the cell membrane spanning the entire length of it, known as transmembrane proteins, lend themselves as favourable targets.

The Immunoglobulin family is a large protein family containing a number of subfamilies. One such subfamily, Siglec, has lot of features in common among the members. These include the presence of transmembrane domain, certain motifs which are highly homologous among the members of the family. So these features are very much helpful in identifying new members of the family by computational analysis.

In the initial stage, scientists identified representative protein sequences with well characterised features from the siglec family as input to the processing pipeline of GeneGrid. At the GAM-end, these sequences were submitted to appropriate Bioinformatics applications on available resources, with the parameters chosen by the scientist. Steps followed in this experiment are briefly described below.

Input protein sequences were searched against various sequence databases including non-redundant protein (nr) and human genome sequences [15] utilising the subprogram options in BLAST like tblastn, blastp etc. Homologous sequences identified in this step were extracted into a database removing the duplicate entries. In the next step, a transmembrane prediction

program was used to find out whether these sequences contained the domains characteristic of transmembrane proteins. After isolating the positive results, these sequences were further processed using the signal prediction program to check for the leader sequences that help the nascent protein to reach its destination. After this elimination step, the remaining sequences were aligned with initial query sequences for identification of the characteristic position of some of the motifs. The final pool of sequences obtained from the pipeline contained either already characterised members of the siglec family or potential new members of the family. Detailed results of this experiment will be published elsewhere.

This experiment was conducted on the GeneGrid testbed by allowing the user to input the initial sequences and parameters through a client interface. This interface masked the underlying complexity of GeneGrid system from the user and allowed the scientific analysis of the result. The workflow based approach helped the scientists to avoid repeated access of different applications on multiple servers for the experiment.

6. Conclusion and Future Work

GeneGrid integrates various Bioinformatics programs available on different resources allowing the scientists to easily access the diverse applications without bothering to visit many web servers. As the system takes care of monitoring and selection of appropriate resource for the task requested, it relieves the user of such selections and more importantly, utilises the resources in an efficient way, reducing the overall time of the job.

Development of a functional prototype of GeneGrid has clearly illustrated the viability of utilising OGSI-compliant grid services for integrating heterogeneous Bioinformatics programs with diverse requirements on different resources while following a workflow based approach. However, it may be warned that the utilisation of Globus Toolkit, which itself is in developmental stage, as the base for creation of the services hindered the smooth development and deployment of GAM services through issues ranging from technical to ones arising from lack of proper documentation.

The project is about to reach the half-way mark. Work on the Application and Resource management as well as the development of GeneGrid Application & Resources Registry is currently underway. Security at the GAM level is another important functionality to be implemented. More Bioinformatics applications will be integrated and new programs will be developed for formatting the results from the application programs, leading to the development of pipelines for processing the workflows.

It is anticipated that demonstration of the functionality of GeneGrid will elevate the interests of scientists in utilising the framework and expertise leading to fruitful collaborations within the country and worldwide.

7. References

[1] Foster I., Kesselman C., Tuecke S. (2001) The Anatomy of the Grid: Enabling Scalable Virtual Organizations. *International J. Supercomputer Applications*, 15(3).

[2] Gray W.A. and Thomson C. (2003) Bioinformatics and eScience. *Proceedings of the UK eScience All Hands Meeting 2003*, 66-71.

[3] http://www.nesc.ac.uk/projects/escience_projects.html

[4] http://eol.sdsc.edu

[5] Donachy P., Harmer T.J., Perrott R.H. *et al*. (2003) Grid Based Virtual Bioinformatics Laboratory. *Proceedings of the UK eScience All Hands Meeting 2003*, 111-116.

[6] Foster I., Kesselman C., Nick J., Tuecke S., The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. *Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002*

[7] Tuecke S., Czajkowski K., Foster I., Frey J., Graham S., Kesselman C., Maguire T., Sandholm T., Vanderbilt P., Snelling D., Open Grid Services Infrastructure (OGSI) Version 1.0. *Global Grid Forum Draft Recommendation, 6/27/2003*.

[8] Altschul, S.F., Gish, W., Miller, W., Myers, E.W. & Lipman, D.J. (1990) Basic local alignment search tool. *J. Mol. Biol.* 215:403-410.

[9] Simpson D., Kelly N., Jithesh P.V., Donachy P. *et al*. (2004) A Practical Workflow Implementation for a Grid Based Virtual Bioinformatics Laboratory. *Proceedings of the UK eScience All Hands Meeting 2004*.

[10] OGSA-DAI, http://www.ogsadai.org.uk

[11] Kelly N., Jithesh P.V., Simpson D., Donachy P. *et al*. (2004) Bioinformatics Data and the Grid: The GeneGrid Data Manager. *Proceedings of the UK eScience All Hands Meeting 2004*.

[12] The Globus Toolkit, http://www-unix.globus.org/toolkit

[13] The Apache Jakarta Project, http://jakarta.apache.org/tomcat/index.html

[14] The Apache Ant Project, http://ant.apache.org

[15] Available for download at the National Centre for Biotechnology Information, ftp://ftp.ncbi.nlm.nih.gov/blast/db/