Comparison of Data Access and Integration Technologies in the Life Science Domain

Dr R.O. Sinnott [1], D. Houghton [2],
[1] National e-Science Centre, University of Glasgow, UK
[2] MRC Human Genetics Unit, Edinburgh
ros@dcs.gla.ac.uk
d.houghton@hgu.mrc.ac.uk

Abstract

Dealing with the deluge of data is the key challenge facing the life science community. The OGSA-DAI technology from the Grid community and IBM Information Integrator product were applied within the Biomedical Research Informatics Delivered by Grid Enabled Services (BRIDGES) project to support federated data access and integration of biomedical data sets. This paper outlines the experiences of applying these solutions for this purpose.

1. Introduction

There are many databases in existence throughout the world which contain biological data. The actual data may relate to structure, function, publication, annotated sequences to name but a few categories. Through the profusion of genetic and pharmaceutical research and high throughput capabilities, these datasets are becoming ever larger.

Most of the major genomic databases have browser interfaces through which the user can initially input a set of parameters to receive more detailed information on say a particular gene or sequence. This often results in having to 'drill down' through the web resource often via a multitude of hyperlink options to eventually find the desired information or a link to another website for further details. It quickly becomes apparent that to do this manual task repetitively for each item is quite time consuming. Not only this, but collating all the relevant data into one coherent document can be quite a painstaking procedure.

As a result many of the activities that biomedical scientists undertake in performing their research are done in a time consuming and largely non-automated manner. This is typified through "internet hopping" between numerous life science data sources. For example, a scientist might run a microarray experiment and identify a gene (or more likely set of genes) being differentially expressed. This gene is then used as the basis for querying a remote data source (e.g. MGI in Jackson [13]). Information retrieved from this query might include a link to another remote data source, e.g. on who has published a paper on this particular gene in MedLine [14] or PubMed [15]. Information from these repositories might include links to ensembl [16] where further information on the gene, e.g. its start/end position in a given chromosome can be established. Such sequences of navigations typify the research undertaken by scientists.

Grid technologies offer capabilities to overcome these difficulties through single queries that can be federated across a variety of data resources and subsequent integration of the resulting data sets. The BRIDGES project has explored two leading technologies for this purpose: OGSA-DAI [3] and IBM Information Integrator [4]. This paper compares the experiences in applying these technologies.

2. Background to BRIDGES

The DTI funded BRIDGES project [1] was set up to assist the Wellcome Trust funded Cardiovascular Functional Genomics (CFG) project [2] in developing a Grid infrastructure providing coherent, fast, reliable, and user friendly access to such data. The BRIDGES project began in October 2003 and is due to end at the end of 2005. The CFG project is investigating possible genetic causes of hypertension, one of the main causes of cardiovascular mortality. This consortium which involves five UK sites and one Dutch site is pursuing a strategy combining studies on rodent models of disease (mouse and rat) contemporaneously with studies of patients and population DNA collections. The BRIDGES project has been funded by the UK Department of Trade and Industry to develop a Grid based compute and data infrastructure to support the needs of CFG. In this paper we focus primarily on the data Grid component of BRIDGES.

A key component of the BRIDGES architecture is the Data Hub (Figure 1). This represents both a local data repository, together with data made available via externally linked Grid accessible data sets. These data sets exist in different heterogeneous, remote locations with differing security requirements. Some data resources are held publicly (e.g. genome databases such as Ensembl [16], gene function databases such as OMIM [17] and relevant publications databases such as MedLine [14]); whilst others are for usage only by specific CFG project partners (e.g. quantitative trait loci (QTL) data sets [22]).

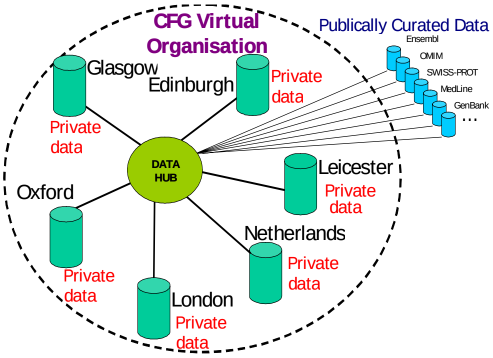


Figure 1: Data Distribution and Security of CFG Partners

We applied IBM Information Integrator and the Grid communities OGSA_DAI product release 4 to develop this data hub.

3. Technology Background

Information Integrator is essentially a suite of wrappers for both relational (Oracle, DB2, Sybase etc) and non-relational (flat files, Excel spreadsheets, XML databases etc) targets which extend the integration capabilities of DB2 database (DB2). Use of the wrappers allows DB2 SQL and procedural languages to be used seamlessly to query foreign data sources and set up of a ‘federated’ view of these resources thus allowing applications access to data as though it were in a single database.

Information Integrator is free for academic use for as long as IBM allows it to be so. For others, a licence fee is required. It is based upon the well established DB2 database. It comes with a suite of tools and utilities with which the database administrator can monitor and optimize the database. You have the option to interact with the database either by command line or graphical interface. There are options to create either Java or SQL stored procedures and customized functions.

OGSA-DAI middleware provides application developers with a range of service interfaces, written in Java, allowing data access and integration via the Grid. It is not a database management system in itself rather it employs a Grid infrastructure to perform queries on a set of relational and non-relational data sources and conveys the result sets and associated metadata ultimately back to the user application via SOAP. Through OGSA-DAI interfaces, disparate, heterogeneous data sources and resources can be treated as a single logical resource.

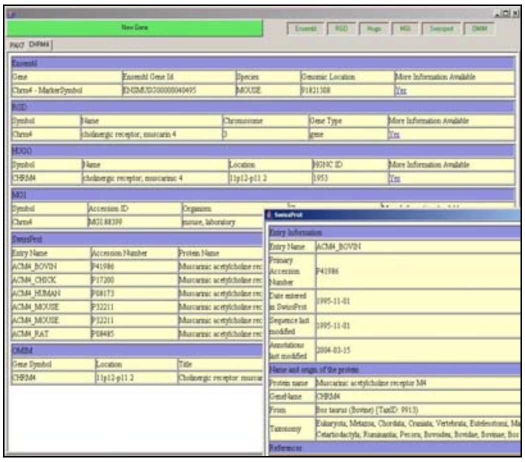
OGSA-DAI is free. It is open source. It is at the forefront of research into grid based data access. It has a relatively small core development team but it is sponsored by IBM. It requires a degree of specialized ‘hands on’ coding especially to customise any extensions to the product. It has limited number of data source types both relational and non-relational with which it can communicate although a good cross section is represented. The documentation is clear and concise and help is forthcoming from the development team.

3.1 Use Case Background

BRIDGES proposed that by creating a federated view of all the commonly accessed data sources with Information Integrator and with OGSA-DAI, user oriented applications could be developed which could take the starting point of the scientists line of enquiry, e.g. a gene name/identifier, issue a query to the federated database and present all available information back to the user in user friendly and configurable perspective. Two applications were developed for this purpose: MagnaVista and GeneVista (figure 2). MagnaVista is a Java application which utilises WebStart technology [24] as its delivery mechanism. GeneVista has been developed based upon portlet technologies.

In essence the functionality of GeneVista is very similar to MagnaVista. However, it does not support the richness of personalisation. We note that this was at the request of the scientific end users. The CFG scientists simply wanted to be able to select a collection of gene names and retrieve all available information. Few of them bothered with personalisation possibilities. The basic front end to GeneVista was designed to reflect this.

The GeneVista portlet simply requires that the scientist input the gene names that they are interested in and selects submit. Following this, HTML based data sets are returned and presented within the browser window as shown in Figure 2.



Ensembl

Gene	Ensembl Gene Id	Species	Genomic Location	More Information Available
Chrm4 - MarkerSymbol	ENSMUSG00000040495	MOUSE	91821308	Yes

RGD

Symbol	Name	Chromosome	Gene Type	More Information Available
Chrm4	cholinergic receptor, muscarin 4	3	gene	Yes

HUGO

Symbol	Name	Location	HGNC ID	More Information Available
CHRM4	cholinergic receptor, muscarinic 4	11p12-p11.2	1953	Yes

MGI

Symbol	Accession ID	Organism
Chrm4	MGI:88399	mouse, laboratory

SwissProt

Entry Name	Accession Number	Protein Name
ACM4_BOVIN	P41986	Muscarinic acetylcholine rec
ACM4_CHICK	P17200	Muscarinic acetylcholine rec
ACM4_HUMAN	P08173	Muscarinic acetylcholine rec
ACM4_MOUSE	P32211	Muscarinic acetylcholine rec
ACM4_MOUSE	P32211	Muscarinic acetylcholine rec
ACM4_RAT	P08485	Muscarinic acetylcholine rec

OMIM

Gene Symbol	Location	Title
CHRM4	11p12-p11.2	Cholinergic receptor: muscar

SwissProt

Entry Information

Entry Name	ACM4_BOVIN
Primary Accession Number	P41986
Date entered in SwissProt	1995-11-01
Sequence last modified	1995-11-01
Annotations last modified	2004-03-15

Name and origin of the protein

Protein name	Muscarinic acetylcholine receptor M4
GeneName	CHRM4
From	Bos taurus (Bovine) [TaxID: 9913]
Taxonomy	Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Euteleostomi; Ma Cetartiodactyla; Ruminantia; Pecora; Bovoidea; Bovidae; Bovinae; Bos

References

Figure 2: GeneVista Basic Usage for Gene Query

4. Experiences and Comparison

Ideally when comparing Information Integrator and OGSA-DAI, identical scenarios would be supported. This is not the case for three main reasons. Firstly, whilst in spirit Information Integrator and OGSA-DAI address the same problems (enabling access to distributed data sets) the technologies do not directly match. For example, one of the genome databases used by the CFG scientists (MGI in Jackson) is based upon Sybase. This is not currently supported in the OGSA-DAI release. Secondly, most publicly available genomic databases do not provide programmatic access. As a result much of the efforts in BRIDGES has been spent on provide a compromise solution: federating access where possible and providing a local data warehouse where not. Finally, BRIDGES had to develop a working solution for the CFG scientists. Hence much of the initial focus was spent on Information Integrator (with the industrial collaborator being IBM). Once a working system was developed, the drive to pursue OGSA-DAI based solutions was influenced by a combination of efforts needed to maintain the existing system and team changes As a result a full and complete comparison of these technologies in the life science domain has not been possible. That said useful comparisons have been made which we now describe.

.

4.1 Setup and Installation Process

The process of accessing, obtaining and installing and configuring IBM Information Integrator is non-trivial. Accessing Information Integrator through the “Scholar’s Program” can be a time consuming procedure and requires authorisation. Advanced knowledge of the vendor clients that the wrappers may use (e.g. Sybase 12.5ASE Client) eases the installation process. This is especially true on Linux as one has to manually edit configuration files and run rebinding scripts if the clients are installed at a later date. It is also worth noting that it helps the installer to run smoothly if all components are unzipped to the same directory level. In the case of IBM’s WebSphere on which the BRIDGES portal is based where the description of the download file is followed by a number in brackets e.g. (04) the unzipped directory should be renamed ‘04’.

Download of OGSA-DAI is by contrast a much friendlier affair. One visits the download site, signs up for access and is issued with a username and password for http authentication to the download area. New releases are advertised by email (submitted during the sign up process) and all downloads are supplied with the obligatory README file which provides adequate guidance as to the setup procedure and what additional downloads one may require (e.g. JDBC drivers, apache commons utilities). With the release of v4 the install process can be done via a GUI which performs reliably.

4.2 Post-installation

Regarding Information Integrator, IBM provides a wealth of Redbooks available from their website. Unfortunately at the time of the BRIDGES work in applying Information Integrator (end of October 2004), these were not descriptively named so it is a matter of opening each one to discover its title and topics dealt with. Whilst informative, it proved to be quite time consuming searching through these documents for specific information. The online documentation was found to be very helpful through its search facility, particularly with syntax questions. In reality though, it is most convenient if there is a well subscribed user group to which one can post, review and reply to topics, e.g.

LazyDB [23]. It should be noted that within the last year, navigation around IBM's website has improved significantly providing easier access to online documentation and resources.

OGSA-DAI comes with its own HTML documentation which can be downloaded separately as required. The content and navigability of this has improved over each release as more detailed coding examples have been given. User support is quick and efficient and in our experiences with a response time typically less than 24 hours.

4.3 Initial Usage Experiences

Attempts were made initially to use Information Integrators XML wrapper to query certain bio-databases, e.g. the Swissprot/Uniprot database [18]. This database in XML format is available for ftp download and is over 1.1GB in size. The wrapper failed in its attempt to work with this file, as, according to an IBM white paper (DB2 Information Integrator, XML Wrapper Performance. October 31, 2003), the whole document is loaded in memory as a Document Object Model (DOM). The suggestion of splitting the file into 40MB chunks seemed a cumbersome solution and so it was decided to parse the file and import it into DB2 relational tables. Since each flat file wrapper has to be specifically configured manually to match the file 'columns' it would seem to be no greater effort to actually write a programme to parse the file according to format or delimiting character and read the data directly into database tables which have all the benefits of indexing, constraints and optimisation associated with them.

A further issue was that the format of some of the downloaded database flat files was not compatible with a provided wrapper (e.g. OMIM [17]), it was decided to adopt the approach of converting all flat files to relational tables. The initial parse of the Swissprot database used table 'Inserts' to commit the data immediately to the database as the file was read by the parsing program. (Java SAX parsing was used and primary and foreign keys were updated using insert triggers). This proved to be incredibly slow (84 hours for the 1.1GB file with around 500,000 inserts to the database. This is obviously not a viable option if the database is in constant use and the latest version of the Swissprot release is to be incorporated into the BRIDGES model.

4.4 Schema Changes

A major problem faced by both Information Integrator and OGSA-DAI is the changes made to the schema design associated with the remote data source. For BRIDGES, the two relational data sources which would allow public programmatic access were Ensembl [16] (MySQL - Rat, Mouse and Human Genomes, Homologs and Database Cross Referencing) and MGI [13] (Sybase - mainly Mouse publications and some QTL data.) Flat files were downloaded for Rat Genome Database [19] (RGD), OMIM [17] (Online Mendelian Inheritance in Man), Swissprot/Uniprot [18] (Protein sequences), HUGO [25] (Human Gene Ontology) and GO [21] (Gene Ontology).

Obviously changes made to the schema of a third part database are completely outwith our control. Ensembl change the name of their main gene database every month! The schema of the database has been drastically altered on at least 3 occasions during the BRIDGES project. MGI have had one major overhaul of all their table structure. In these cases queries to these remote data sources will fail. This is especially costly when large federated queries are made which ultimately fail when perhaps only one or two columns of a particular sub-query are problematic, i.e. are based on an erroneous schema.

The case of flat files is slightly different. Since the flat file has been downloaded onto the BRIDGES data server and either wrapped or parsed into the database, queries on this will survive but only until a newer schema-altered file is incorporated.

It should be noted that Information Integrator insists that the flat file being wrapped exists on a computer with exactly the same user setup and privileges as the data server itself. This is unlikely to be the case with regard to most curated life science data sets. In is also the case that the manual task of mapping flat file columns to database wrappers must still be done in the event of remote changes.

The ability to create Materialized Query Tables (MQTs) in Information Integrator can insulate the queries from these remote changes. An MQT is a local cache of a remote table or view and can be set to refresh after a specified time interval or not at all. Here we have a balancing act of deciding how important it is to have up to the minute data (refreshed frequently) or slightly older data but impervious to schema changes. The MQT can be optimized to try the remote connection first, and if it is unavailable to resort to the local cache but in the event that the remote connection is good but the query throws and exception because the schema has changed, then the query will fail.

BRIDGES has come up with a partial solution to the problem of remote schema changes. An application was developed (*Bridges_wget*) which systematically checks for database connections and if the connection is made runs a sample query specifically naming columns (so not a *select **) to test if the table schema has changed. If all is well, remote flat files are checked for modification dates. If newer ones are found at the remote site they will be downloaded, parsed (if necessary) and either loaded into the database or the db administrator notified that new files are available.

Hopefully this will go some way to help in keeping the BRIDGES database up to date with current data. We note however that the parsers developed are not semantically intelligent so it would require updating the code (Java) to meet with file format modifications

4.5 Data Independence Issues

One of the issues around creating a federated view of remote data sources is the fact that these data sources are largely independent of each other. It is not always possible to find a column which will act as a foreign key over which the joining of the two (or more) databases can occur. When there is a candidate, often the column name is not named descriptively so to give a clue as to which database might be joined to.

Were all the databases developed by the same development team for example within a company intranet, this possibility of creating large scale joins across several homogenous databases would be much clearer.

As it is one bio database may have a row with a gene identifier column with another column holding an accession ID for an entry for this gene in another database. In this way the join can be made.

In the case of Ensembl a row containing a gene identifier contains a Boolean column indicating whether a reference exists in another database. For example, RGD_BOOL=1 would indicate that a cross reference can be made to the RGD database for this gene identifier. We now have to query the Ensembl RGD_XREF table to obtain the unique ID for the entry in the RGD database.

The query to RGD may then contain references to other databases and indeed back to Ensembl and one ends up with a circular referencing problem.

BRIDGES dealt with this problem by caching all the available unique identifiers along with the database in which it is found from all the remote data sources in a local materialized query table. When a match is found, the associated data resource is queried and all results returned to the user. It is then up to the user to decide which information to use.

In addition to the schema checking and file download programme (*Bridges_wget*), BRIDGES has developed a knowledge base for problems and solutions in an attempt at providing a project wide resource to assist developers. It can easily be extended to incorporate other projects and modules so that other projects working with DB2 for example can share their fixes and workarounds.

4.6 Data Return Size Issues

In practice with Information Integrator queries are issued from the MagnaVista or GeneVista application to stored procedures within DB2. Based on the parameters received the appropriate databases are queried. Rather than create one large join across all the remote data sources, with Information Integrator the stored procedure makes individual queries to each database and returns the result sets into a temporary table. It is this temporary table which is returned to the client.

With Information Integrator, selectively populating the temporary table allows us to make sure no duplication of data is returned. To illustrate this problem, an Ensembl gene identifier may be associated with several hundred publications in the MGI database and also a particular Swissprot accession ID and the *taxon* element is required to be returned from Swissprot. The *taxon* table is three relations away from the Accession table. There may be 5 *taxons* to be returned which means that there is no possibility of running a DISTINCT select statement. This would mean that all the publication data would be returned along with each *taxon*.

The fact that large publication data may well be involved in the bio database query could easily exceed the maximum *tuple* size returned from DB2. It is possible to increase the *tuple* size by increasing the page size for the database. This of course could work against performance if the bulk of the queries issued would return a small dataset and therefore there would be a redundancy in the page size overhead.

Using OGSA-DAI to perform the same type of query as has been outlined above is more problematic. To begin with there is no cache or local views available as with Information Integrator through MQTs. Instead there has to be a query made to each data resource to begin with in order to obtain all the unique identifiers. Of course these could be stored in a local file or database by doing a UNION of all identifiers on all databases thus serving as an alternative to the MQT.

The mapping of remote data sources to the 'Integrator' mechanism is always going to have to be done but with OGSA-DAI it is more a manual task rather than the automated '*Discover'* function in Information Integrator.

There is a similarity in the way that the particular stored procedures were constructed within DB2 and the integration procedure in OGSA-DAI in that the former uses temporary tables to store the result sets from each database and the latter requires the existence of a database within which tables or temporary tables can be created to store each result set. It is from this newly constructed table that the client then obtains required data.

It should be noted that if MySQL is to be used as the 'dumping' database by OGSA-DAI, only newer releases support temporary tables and those that do are selective about allowing more than a single SELECT to be performed within a transaction on the temporary table. If actual tables were to be created there would have to be a convention in place regarding table naming otherwise collisions would occur.

The fact that stored procedures can be written in DB2 makes it simpler for the client as all the query processing is done within the body of the procedure but table-based queries can also be implemented in OGSA-DAI and the same results returned.

4.7 Creating Federated Views and Issues

When setting up a federated view in Information Integrator one first chooses a wrapper to use. Then one defines a '*Server'* which contains all the connection parameters. Next '*Nicknames'* are created for the server which are local DB2 tables mapped to their remote counterparts. To aid this process there is a '*Discover'* function available when creating the Nicknames which when implemented will connect to the remote resource and display all the metadata available. One can then choose which of these to use, rename them and alter their schema for local use. Such advanced features are not available with OGSA-DAI.

It should be noted that if one drops any component, every component that is dependent on it will be dropped also. So for example, if a *Server* is dropped all *Nicknames* will be dropped. If a *Nickname* is dropped, all MQTs and Views based on that Nickname will also be dropped. Experience has shown that it is worth saving the original scripts.

We note that there is an unresolved issue regarding MQTs in Information Integrator. MQT's created with the same schema owner as the underlying tables from which they are built, are not accessible to users who do not have administrative privileges.

4.8 Performance Comparison

It is the case from the tests carried out from installation to querying that both applications are able to support the notion of providing an integrated model for data in diverse remote and local data sources. However, in the absence of real response times for multiple queries on large datasets it has been impossible to draw a detailed performance comparison on this level.

As an example of single query response, we ran a search for the *PAX7* gene across the BRIDGES federated view of 7 bio databases. This returned

- One entry from Ensembl Mouse Table. (27 columns)
- One entry from Ensembl Human Table. (27 columns)
- One entry from the HUGO database. (20 columns)

- Eighty five entries from MGI including full abstract and publication details. (11 columns)
- One full entry from the OMIM database including fully annotated publication details. (19 columns)
- Two full entries from Swissprot/Uniprot including full sequence and reference data. (50 columns).

The average response time between MagnaVista and the database including time to rebuild the application perspective GUI was 44 sec. To have the results returned via a Grid Service using OGSA-DAI with the grid service calling the stored procedures on the federated system showed no appreciable difference in the time for the simple example shown above.

Tests are required on a larger initial input number of gene identifiers and also substituting sequences as the input in place of gene identifiers.

5 Conclusions

The BRIDGES project has explored two leading technologies for access to and integration of genomic data sources: OGSA-DAI and IBM Information Integrator. As noted, ideally we would have liked to be able to undertake a full evaluation of these technologies is a common environment (with common data sets and common queries). At the time of writing this has not been possibly for the reasons given previously, namely:

- there are key databases used by CFG scientists which cannot be accessed right now with OGSA-DAI (but can with Information Integrator) since they utilise the currently non-supported Sybase database.
- BRIDGES focus was primarily on developing a working infrastructure for CFG and to a lesser extent to compare associated technologies for this purpose.

Instead, after developing the Information Integrator based solution we simply placed OGSA-DAI between the client and Information Integrator. As a result of this the client became a little fatter as it had to implement the appropriate Grid calls. Specifically, queries from the client were sent to a GridServiceFactory handle which forwarded the request to the relevant DB2 stored procedure and returned the data embedded in a *Perform* document. This provided BRIDGES with a working OGSA-DAI implementation querying and receiving data via Grid Services but actually all the data integration was done by Information Integrator.

We are now working towards a more realistic solution however where OGSA-DAI is used for federated queries between the DB2 data warehouse and the MySQL based ensembl rat, mouse and human databases.

Despite this, a useful comparison has been made between these two technologies. It is clear that a big advantage of using Information Integrator is all the utilities that come with the database management system. This includes amongst others: replication of databases which can be configured to update from single transaction committed to a set time interval for bulk updates; creation of *explain* tables which will graphically show the query author the amount of table scans done as the result of the executed query and thereby allow different solutions to be compared; creation of *tasks* which can be executed immediately or at specified times perhaps when the database is less used, e.g. backing up the whole database image, running statistics and reorganizing tables, or taking *Snapshots* of the database to see where bottlenecks may be occurring.

We note that since our evaluations have been made, IBM have now prototyped an OGSA-DAI wrapper for Information Integrator.

Models of all schemas included in the BRIDGES federated database have been created using IBM's Rational Rose. All use cases have been documented in IBM's Requisite Pro and available via RequisiteWeb.

It is clear that many of the issues faced by all technologies is accessing and using life science data sets are impacted by standards and data models. Changing schemas is indicative of this. A further challenge is often gaining access to the database itself – most often this is not possible, but is essential if Grid technology is to help simplify access to and usage of these data sets. We are currently finalising a report on these issues [6] to be released imminently.

A key challenge in this is ensuring that security is upheld (or not weakened by Grid based solutions). BRIDGES has developed solutions with fine grained authorisation infrastructures based upon PERMIS [7] restricting access to the different data sets and computational resources.

The queries that are supported currently within BRIDGES are fairly simplistic in nature – returning all data sets associated with a named gene. We hope in future to look towards more complex queries, e.g. lists of genes that have been expressed and their up/down expression values as might arise in microarray experiments through a collaboration with Cornell University [10] and Riken Institute [11].

We also expect much of the infrastructure developed within BRIDGES to be refined and extended and used within the recently funded Scottish Bioinformatics Research Network [20].

5.1 Acknowledgements

This work was supported by a grant from the Department of Trade and Industry. The authors would also like to thank members of the BRIDGES and CFG team including Prof. David Gilbert, Prof Malcolm Atkinson, Dr Dave Berry, Dr Ela Hunt and Dr Neil Hanlon. Magnus Ferrier is acknowledged for his contribution to the MagnaVista software and Jos Koetsier for his work on GeneVista. Micha Bayer is acknowledged through his work on Grid based technologies within BRIDGES and Anthony Stell for his involvement on the security aspects of BRIDGES. Acknowledgements are also given to the IBM collaborators on BRIDGES including Dr Andy Knox, Dr Colin Henderson and Dr David White. The CFG project is supported by a grant from the Wellcome Trust foundation.

6. References

[1] Biomedical Research Informatics Delivered by Grid Enabled Services (BRIDGES) project, www.nesc.ac.uk/hub/projects/bridges

[2] Cardiovascular Functional Genomics (CFG) project, www.brc.dcs.gla.ac.uk/projects/cfg

[3] Open Grid Service Architecture - Data Access and Integration (OGSA-DAI) www.ogsadai.org

[4] IBM Information Integrator, http://www-306.ibm.com/software/data/eip/

[5] Minimal Information About a Microarray Experiment (MIAME), http://www.mged.org/Workgroups/MIAME/miame.html

[6] Joint Data Standards Survey (JDSS), http://www.d-archiving.com/JDSS/study.html

[7] D.W.Chadwick, A. Otenko "The PERMIS X.509 Role Based Privilege Management Infrastructure". Future Generation Computer Systems, 936 (2002) 1–13, December 2002. Elsevier Science BV.

[8]MIAMexpress, www.ebi.ac.uk/miamexpress/

[9]MaxDLoad http://bioinf.man.ac.uk/microarray/maxd/maxdLoad

[10] Computational Biology Service Unit, Cornell University, Ithaca, New York, http://www.tc.cornell.edu/Research/CBSU/

[11] Riken Genomic Sciences Centre Bioinformatics Group, Yokohama Institute, Yokohama, Japan http://big.gsc.riken.jp/

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

[13] Mouse Genome Informatics (MGI), www.informatics.jax.org/

[14] US National Library of Medicine, http://www.nlm.nih.gov/

[15] PubMed Central Home, http://www.pubmedcentral.nih.gov/

[16] EMBL-EBI European Bioinformatics Institute, http://www.ebi.ac.uk/ensembl/

[17] NCBI Online Mendelian Inheritance in Man, http://www.ncbi.nlm.nih.gov/OMIM/

[18] UniProt/Swiss-Prot, http://www.ebi.ac.uk/swissprot/

[19] Rat Genome Database (RGD), http://rgd.mcw.edu/

[20] Scottish Bioinformatics Research Network (SBRN), www.nesc.ac.uk/hub/projects/sbrn

[21] Gene Ontology (GO), http://www.ebi.ac.uk/GO/

[22] An Overview of Methods for Quantitative Trait Loci (QTL) Mapping, Lab of Statistical Genetics, Hallym University, http://bric.postech.ac.kr/webzine/content/review/indivi/2002/Aug/1_08_index.html

[23] LazyDB discussion group, http://aspn.activestate.com/ASPN/Cookbook/Python/Recipe/66423

[24] Sun WebStart Technology, http://java.sun.com/products/javawebstart/

[25] Human Genome Organisation, http://www.hugo-international.org/

Author/s:
Sinnott, R. O.;Houghton, D.

Title:
Comparison of data access and integration technologies in the life sciences domain

Date:
2005

Citation:
Sinnott, R. O., & Houghton, D. (2005). Comparison of data access and integration technologies in the life sciences domain. In Proceedings of the UK e-Science All Hands Meeting, Nottingham, UK.

Publication Status:
Published

Persistent Link:
http://hdl.handle.net/11343/28776