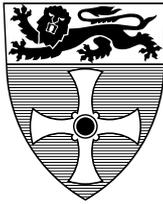**University of Newcastle upon Tyne**

COMPUTING
SCIENCE

Implementing the FuGE Object Model: a Systems Biology Data Portal and Integrator

A. L. Lister, A. R. Jones, M. Pocock, O. Shaw, A. Wipat.

TECHNICAL REPORT SERIES

No. CS-TR-1016   **April, 2007**

Implementing the FuGE Object Model: a Systems Biology Data Portal and Integrator

Allyson L. Lister, Andrew R. Jones, Matthew Pocock, Oliver Shaw, and Anil Wipat.

Abstract

The Centre for Integrated Systems Biology of Ageing and Nutrition has developed a Data Portal and Integrator (CISBAN DPI) that is based on the FuGE Object Model and which archives, stores, and retrieves raw high-throughput data. Until now, few published systems have successfully integrated multiple omics data types and information about experiments in a single database. The CISBAN DPI is the first published implementation of FuGE that includes a database back-end, expert and standard interfaces, and utilizes a Life Science Identifier (LSID) Resolution and Assigning service to identify objects and provide programmatic access to the database. Having a central data repository prevents deletion, loss, or accidental modification of primary data, while giving convenient access to the data for publication and analysis. It also provides a central location for storage of metadata for the high-throughput data sets, and will facilitate subsequent data integration strategies.

Bibliographical details

LISTER, A. L., JONES, A. R., POCOCK, M., SHAW, O., WIPAT, A.

Implementing the FuGE Object Model: a Systems Biology Data Portal and Integrator
[By] A. L. Lister, A. R. Jones, M. Pocock, O. Shaw, A. Wipat.

Newcastle upon Tyne: University of Newcastle upon Tyne: Computing Science, 2007.

(University of Newcastle upon Tyne, Computing Science, Technical Report Series, No. CS-TR-1016)

Added entries

UNIVERSITY OF NEWCASTLE UPON TYNE
Computing Science. Technical Report Series. CS-TR-1016

Abstract

The Centre for Integrated Systems Biology of Ageing and Nutrition has developed a Data Portal and Integrator (CISBAN DPI) that is based on the FuGE Object Model and which archives, stores, and retrieves raw high-throughput data. Until now, few published systems have successfully integrated multiple omics data types and information about experiments in a single database. The CISBAN DPI is the first published implementation of FuGE that includes a database back-end, expert and standard interfaces, and utilizes a Life Science Identifier (LSID) Resolution and Assigning service to identify objects and provide programmatic access to the database. Having a central data repository prevents deletion, loss, or accidental modification of primary data, while giving convenient access to the data for publication and analysis. It also provides a central location for storage of metadata for the high-throughput data sets, and will facilitate subsequent data integration strategies.

About the author

Allyson L. Lister is a Research Associate (RA) and PhD student in CISBAN and the School for Computing Science at Newcastle University.

Matthew Pocock is an RA in the School for Computing Science at Newcastle University.

Oliver Shaw is an RA in the School for Computing Science and CISBAN at Newcastle University.

Andrew R. Jones is an RA in the School of Computing Science at the University of Manchester.

Anil Wipat is a Reader in the School for Computing Science and is a P.I. in CISBAN at Newcastle University.

Suggested keywords

FUNCTIONAL GENOMICS,
HIGH-THROUGHPUT EXPERIMENTS,
FUGE,
LSID,
EXPERIMENTAL WORKFLOWS,
DATABASES,
DATA STANDARDS,
DATA SHARING,
METADATA,
DATA INTEGRATION.

Implementing the FuGE Object Model: a Systems Biology Data Portal and Integrator

Allyson L. Lister[1], Andrew R. Jones[2], Matthew Pocock[1], Oliver Shaw[1], and Anil Wipat[1]

[1]CISBAN & School of Computing Science, Newcastle University, Newcastle upon Tyne NE1 7RU
[2]School of Computer Science, University of Manchester, Manchester, UK
a.l.lister@newcastle.ac.uk, anil.wipat@ncl.ac.uk

Abstract. The Centre for Integrated Systems Biology of Ageing and Nutrition has developed a Data Portal and Integrator (CISBAN DPI) that is based on the FuGE Object Model and which archives, stores, and retrieves raw high-throughput data. Until now, few published systems have successfully integrated multiple omics data types and information about experiments in a single database. The CISBAN DPI is the first published implementation of FuGE that includes a database back-end, expert and standard interfaces, and utilizes a Life Science Identifier (LSID) Resolution and Assigning service to identify objects and provide programmatic access to the database. Having a central data repository prevents deletion, loss, or accidental modification of primary data, while giving convenient access to the data for publication and analysis. It also provides a central location for storage of metadata for the high-throughput data sets, and will facilitate subsequent data integration strategies.

Keywords: Functional Genomics, High-Throughput Experiments, FuGE, LSID, Experimental Workflows, Databases, Data Standards, Data Sharing, Metadata, Data Integration.

1 Introduction

Systems approaches to biological research lead to new challenges in data management and integration. By their very nature, such projects can generate a large number of diverse datasets that must be archived, correctly marked up with appropriate metadata, and made available in a suitable form for subsequent analysis and integration. It is desirable that this process be implemented as early as possible after data generation to prevent data loss and ensure traceability throughout the data management cycle. Whilst repositories have been developed previously for single experiment-types and their associated metadata[1-4], the lack of suitable data standards has hampered the development of a data storage system capable of storing multiple data types in combination with a uniform set of experimental metadata. Recently, the FuGE project was formed with the aim of standardizing the experimental metadata for a range of 'omics' type experiments[5]. The FuGE

standard contains a model of experimental objects such as samples, protocols, instruments, and software, and provides extension points for the creation of technology-specific data standards. The availability of FuGE makes the development of such a portal feasible not only to perform data capture, but also for the purpose of integrating metadata from a range of experimental data sets.

In this paper we describe the development of a data portal and archive for use as part of the data management infrastructure within a Systems Biology centre. The Centre for Integrated Systems Biology of Ageing and Nutrition (CISBAN) aims to advance understanding of the complex mechanisms underpinning the ageing process and of how these mechanisms are affected by nutrition. The CISBAN Data Portal and Integrator (CISBAN DPI) is a generic archive for high-throughput (HT) and quantity-rich data sets, such as transcriptomics and proteomics data, that aims to integrate information about the experimental procedures generating the data. The DPI utilizes Milestone 3 of the open-source, community-driven Functional Genomics Experiment Software ToolKit (FuGE STK). The structure of the DPI is based on the FuGE Object Model (FuGE-OM) and accepts input and provides output formatted in FuGE Markup Language (FuGE-ML). The main entry point for end-users is a web-based graphical user interface that displays a simplified view of the underlying data structure and allows upload of HT data and experimental information. The DPI STK provides expert users with access to the fully implemented FuGE back-end. The DPI also uses the Life Sciences Identifier (LSID) standard[6] to identify all components of an experiment, including the HT data sets themselves. The final component is an LSID server accepting programmatic queries via SOAP[7] and HTTP[8]. On the submission of a LSID to this server by an appropriate client, the requested object is returned in raw form for the HT data sets and in FuGE-ML format for experimental metadata.

2 Standards Employed

As the use of omics techniques by the scientific community has grown, the diversity, number, and size of HT data sets has increased. Minimum reporting requirements such as MIAME[9], MIAPE[10] and MIBBI[1], and data exchange formats such as PSI-MI[11] and MAGE-ML[12], provide consistent methods of exchange and storage. Around these standardization efforts are a number of public databases able to store and exchange data in formats that comply with their community's standard, including ArrayExpress[4] and GEO[1] for MIAME-compliant databases, and IntAct[3] providing output in PSI-MI formats. A number of journals require deposition to these databases prior to acceptance of submissions.

FuGE. Many journals and research bodies require gene, genome, protein and transcriptome data to be submitted according to *de facto* standards. FuGE was developed as the standard for storing information about a functional genomics experiment, though the object model is generic enough to be useful for more than functional genomics, and could model any biological experiment requiring the storage

---

[1] http://mibbi.sourceforge.net

of data, high-throughput or otherwise. By modeling information including samples, protocols, instruments, and software, FuGE provides an exchange format that allows experiment sharing and comparison between groups. The microarray[2] and proteomics standards[3] groups have already adopted the FuGE standard.

Universal Modeling Language (UML). UML[4] helps developers create and visualize models of systems, usually software systems. While on its own, UML is a useful syntax to aid the correct structuring of a large project or application, greater functionality can be drawn from UML diagrams when used in conjunction with Model Driven Architecture (MDA) generators, which transform UML documents into deployable systems. FuGE utilizes such an application in the creation of its XML Schema Definition (XSD)[5] and STK.

eXtensible Markup Language (XML). The bioinformatics community makes heavy use of XML in data exchange and in programmatic access to databases. XML can be heavily structured to make tokenizing and retrieval of the information simple. The template with which FuGE-ML documents are created, and against which they may be validated is an XML Schema (XSD). The XSD informs the programmer as to the content and structure for a corresponding correctly-formatted XML document.

Life Science Identifiers (LSIDs). Universal Resource Names (URNs)[13] are a combination of Namespace Identifier (NID) and Namespace Specific String (NSS) in the format "'urn:' <NID> ':' <NSS>", and provide a useful way of uniquely describing objects. Contrary to location-dependent Universal Resource Locators (URLs)[14], a URN's main benefit is its location independence: an object may still be resolved via its URN, even if its physical location has changed. In an LSID, the NID is 'lsid', and the NSS holds a domain name identifying the assigning namespace. An LSID must also include an authority-specific namespace and a unique identifier, in that order[6]. An example of an CISBAN-formatted LSID is: *urn:lsid:cisban.cisbs.org:Investigation:1*. Here, "cisban.cisbs.org" is the NSS, "Investigation" is the authority-specific namespace, and "1" is the unique identifier.

3 Architectural design and implementation

3.1 Requirements

Until now, there have been no published implementations of the FuGE STK due to the relative youth of the project. The use of FuGE-ML on its own meets the basic requirement of the standard: to completely describe an experiment, including data,

---

[2] Microarray Gene Expression Data (MGED) society, http://www.mged.org/

[3] Proteomics Standards Initiative, http://www.psidev.info/

[4] http://www.uml.org

[5] http://www.w3.org/XML/Schema

analyses, and information about the structure of the experiment. However, if storage and modification history are required, the issue of versioning becomes dominant. The core FuGE structure can record simple audit information but does not allow storage of entire histories of changes to experimental objects. An archive or storage database based on FuGE requiring the full history of the recording of an experiment must implement versioning. A small addition to the FuGE-OM, implemented with hand-written code, provides this capability. For versioning to be implemented properly, all objects must have stable and globally unique identifiers so they may be released publicly without fear of overlap. Additionally, while the MDA approach can auto-generate a web front-end, we found the resulting application too complex to be useful to the targeted users within CISBAN. It was more efficient and more useful to manually develop an easy-to-use web application.

3.2 Architecture

An overview of the architecture of the CISBAN DPI is shown in Figure 1. The core component of the FuGE Milestone 3 STK is the FuGE-OM. AndroMDA[6], an MDA generator, is used to transform the UML into:

- a Hibernate[7]-compatible database
- a Hibernate+Spring[8] persistence and query layer, application framework
- Plain Old Java Objects (POJOs) as representations of FuGE objects
- an XSD

Hibernate is a service that allows the creation of an object/relational database persistence and query layer, thus abstracting the low-level database code from the programmer interface. Using AndroMDA-generated classes in conjunction with Hibernate and Spring lets the developer work only with POJOs, and when ready to populate, query or modify the database, send those objects to a service layer. The service layer controls connections to Hibernate sessions, where standard database operations are performed. AndroMDA can be configured such that any database management system compatible with Hibernate may be used.

While all code discussed so far is generated automatically by AndroMDA, there are sections which must be manually written. The current version of Hibernate has limited XML support, but does not support sufficient features of the FuGE XSD to generate FuGE-ML that can be validated against it. However, there are many freely available XML processing libraries that create marshaling and unmarshaling classes, given an XSD. In an addition made by the authors, the FuGE Milestone 3 STK[9] now includes JAXB2[10]-generated code to help the user load and unload XML files into the generated database. In addition to the JAXB2 code, conversion classes have been manually written that map the JAXB2 objects to the POJOs created by AndroMDA.
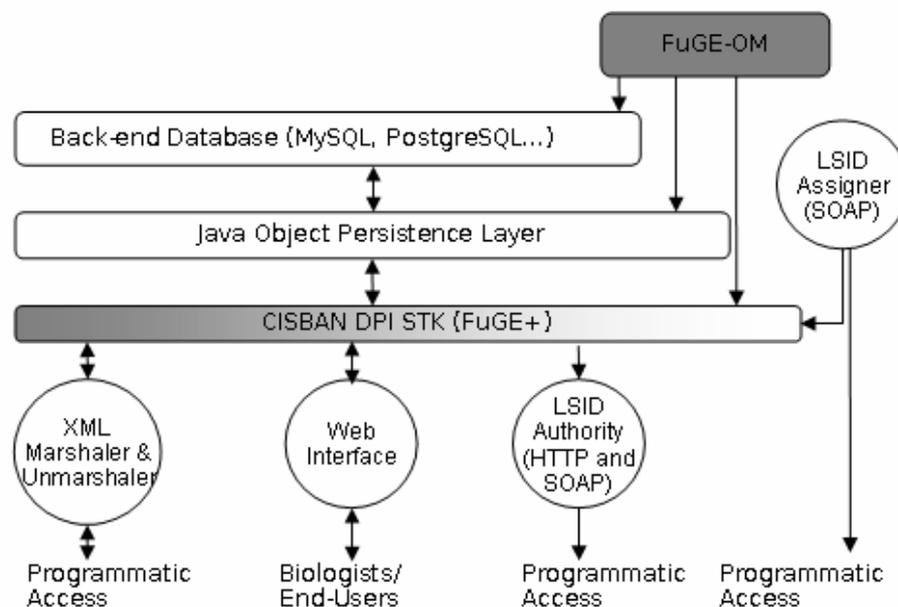
---

[6] http://www.andromda.org

[7] http://www.hibernate.org

[8] http://www.springframework.org

[9] http://fuge.sourceforge.net/dev/index.php#beta_STK_m3

[10] https://jaxb.dev.java.net/

Fig. 1. An architectural overview of the DPI showing the flow of data between users and the database. Dark grey boxes signify hand-written code, white code that is automatically generated. The layer that is shaded from grey to white contains both types of code. Circles mark access points.

Additions to the FuGE STK. A number of additions were made to the FuGE-OM and to the FuGE STK to produce the CISBAN DPI, which fully includes and extends the official Milestone 3 release. One of the main aims of the DPI is to layer changes over the existing OM, maximizing the compatibility of the DPI with FuGE. All described additions add only a single attribute to `Identifiable` elements in the FuGE XSD. The LSID service and Web GUI presented here are the first to be used in conjunction with FuGE.

CISBAN Identifiers are LSIDs. The DPI functions both as an archive and as a source for data integration research, and as such it is imperative that the object referenced by a CISBAN identifier never changes. A specific type of URN, the LSID, was chosen to meet CISBAN requirements. The LSID specification requires object-identifier permanence: once assigned, it must always resolve to an identical byte string. If the data changes, a new LSID must be assigned. Additionally, a publicly-available toolkit[11] for implementing LSID resolution makes LSID usage straightforward.

A more complex versioning system. The ability to store a complete, versioned history of all identifiable FuGE objects has been created with the addition of a set of entities

---

[11] http://lsid.sourceforge.net

layered on top of the existing OM. These additions are summarized in Figure 2, a simplified diagram of the new entities.
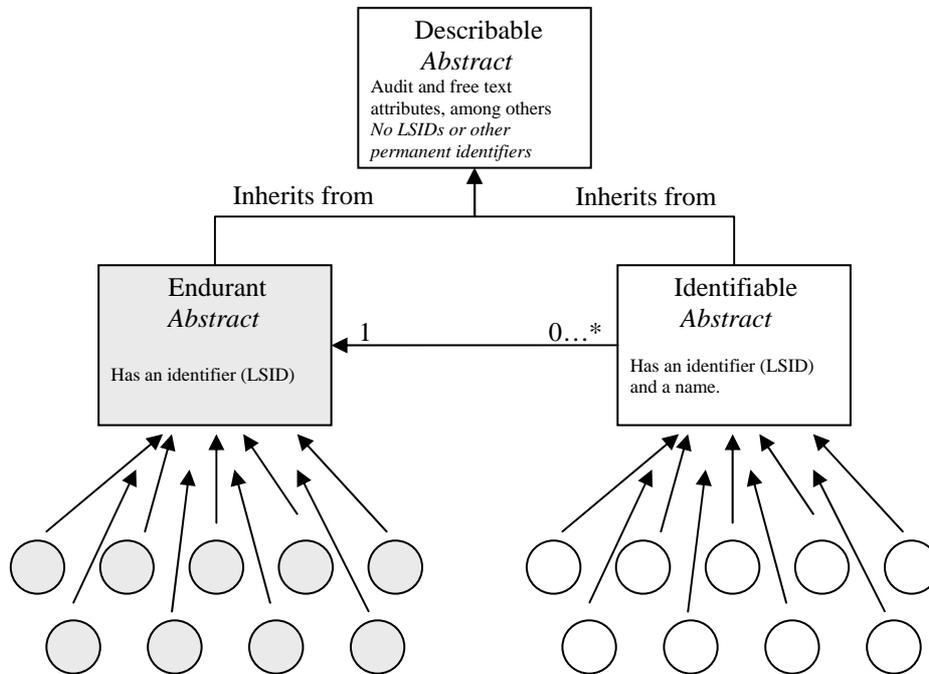
Fig. 2. A simplified view of the additions made to the FuGE-OM when creating the CISBAN DPI. The dark grey objects represent those entities added to create the CISBAN DPI. The white objects are representations of some of the entities included in the FuGE-OM. `Endurant` entities are children of `Describable`, as are `Identifiable` entities. Additionally, an association between `Identifiable` and `Endurant` has been added to link a "permanent" `Endurant` object to its `Identifiable` "snapshot" objects.

All objects in the standard FuGE-OM inherit from `Describable` allowing audit trails, additional ontology terms and textual descriptions to be added to all objects. Entities that inherit directly from `Describable` have no LSIDs or other identifiers. Most objects in FuGE also inherit from `Identifiable`, allowing a unique identifier and human-readable name to be attached to an object. The abstract entity `Endurant` is a new child of `Describable` and sibling of `Identifiable`. Both `Identifiable` and `Endurant` inherit from `Describable` and have an identifier attribute, which in the CISBAN DPI is an LSID, therefore all children of these two entities are identified by an LSID.

`Endurant` objects do not change their intrinsic value over time, and are identified with an LSID. They may point to many versions of non-enduring objects, one version for each modification and are, by CISBAN definition, unresolvable. This prevents the LSID Authority from breaking the LSID specification, as the authority must not allow the same LSlD to resolve to different objects. All other objects identified with an LSID are children of the `Identifiable` entity, which identify a particular version

of the object associated with the `Endurant`. Because of this, such LSIDs will always point to the same, single, version of the object, and are resolvable.

The `Endurant` classes allow every `Identifiable` in the FuGE-OM to represent a single revision in the history of the `Endurant`: new objects are added to the database with every change, however minute. The state of an object at any point in its history is easily retrievable, as is the latest version of each object. Error-correction or addition of information, such as fixing a typo in the name of a robot, will create an additional `Identifiable` associated with the same `Endurant`. A fundamental change in an object (such as the addition of a new camera on a robot) will mean the creation of a new `Endurant` with a new LSID.

LSID Authority and Assigner. Based on the LSID toolkit available from the open-source LSID project, a set of jars allowing access to CISBAN LSID services is integrated into the CISBAN DPI's UI and STK. The CISBAN LSID Authority works both as assigner and resolution service. According to the specification, LSID resolution services must implement two methods: `getMetadata()` and `getData()`. `getMetadata()` may return anything, and is not required to always return the same thing. It is the `getData()` method which must indicate an unresolvable error condition or return a byte–identical value for each successful invocation. Table 1 describes the return types for the CISBAN resolution.

Table 1. Summary of the resolution of CISBAN LSIDs. The header column lists the type of LSID, and the header row lists the LSID Authority's available method calls. The "HT Data Set" is Identifiable, but is dealt with differently and thus is summarized separately.

	`getMetadata()`	`getData()`
Endurant	LSID of latest version	null / non-resolvable
Identifiable	LSID of latest version	FuGE-ML for that object
HT Data Set	LSID of latest holding experiment	original file + DPI identifier
Timestamped	LSID of latest version	FuGE-ML as at the stated time

Any valid LSID may have a timestamp appended to it, resulting in retrieval of the FuGE-ML document corresponding to that time point. HT data may also be retrieved with these constructed LSIDs though, being an archive, the DPI only stores one version for each HT data set. To ensure that FuGE-ML formatting changes do not result in a different object being returned in a `getData()` call, any whitespace changes will be resolved by running the FuGE-ML-formatted object through a XML formatter. However, even if whitespace errors are removed, changes in the UML could propagate to the database table name, Java class name and XML element name. Because a single change has a far-reaching effect, attempting to minimize this problem by, for example, returning the Java object, does not help and limits the users. Therefore, whenever such format changes occur, affected LSIDs will become unresolvable, and `getMetadata()` will return an LSID to the latest version.

Large object storage for HT data. An additional blob store was integrated into the database schema created by the STK. This new UML entity, `RawData`, was created

as a concrete child of `Data`, an abstract UML entity already present within the FuGE-OM. Of the other two children of `Data`, `ExternalData` cannot be used as it refers to its data item via a URI and CISBAN data must be stored within the database, and `InternalData` requires a certain internal structure for the data sets that cannot be guaranteed for CISBAN data sets. `RawData` stores the HT data as a blob, though in future other options will be examined such as storing file references to the HT data.

Simple graphical user interface. From requirements-gathering sessions between bioinformaticians and experimentalists within CISBAN, the qualities deemed most important in the CISBAN DPI were ease-of-use and, more importantly, speed-of-use. CISBAN researchers currently require only a few sections of the exhaustive FuGE standard, therefore while the DPI back-end is capable of storing everything modeled in the OM, the front-end provides a simpler view consisting mainly of HT data sets, person and organization details, protocols, and experimental hypotheses and conclusions. The front-end is written in a combination of Java and Java Server Pages (JSPs)[12] and deployed on a Tomcat[13] server. Many sections of the input forms are pre-filled with information on known instruments and methodologies of the CISBAN laboratory groups. While AndroMDA can generate a web front-end, it is a visualization of the OM and therefore not intuitive for the end-user.

4 Availability

Slides, graphics, and the DPI OM and STK are available from the CISBAN DPI project page at http://www.cisban.ac.uk/cisbanDPI.html. A working "sandbox" version of the CISBAN DPI and STK, including an entirely independent example database, LSID application and web interface will be available from the CISBAN website by March 2007. The STK is still at an early stage, and does not come with any guaranteed user support, though comments and questions are encouraged and should be sent to the CISBAN helpdesk at helpdesk@cisban.ac.uk.

5 Conclusion

Systems biology projects, and especially centers, produce a large amount of raw HT data in the course of their research. A central archive provides:
- *a straightforward capture method* for essential experimental metadata,
- *a safe location* to store original, unmodified data to prevent accidental loss or modification,
- *a central location* from which users can view all of their data in a uniform fashion, as well as data from others working on the same project
- *a complete history* for all metadata associated with the HT data sets in the form of versioned objects,

---

[12] https://jsp-spec-public.dev.java.net/
[13] http://tomcat.apache.org/

- *the first complete implementation of FuGE* Milestone 3 from database back-end through to web front-end, including a full library of code to manipulate and export/import information from FuGE-ML input files and the FuGE-OM-based database, which will speed data preparation and ease the submission process,
- *retrieval mechanisms* via a web interface and via an LSID server utilizing HTTP and SOAP protocols, and
- *a single resource* from which bioinformaticians can draw for their own research, and which acts as a first step in the data integration process within CISBAN.

The CISBAN DPI provides uncomplicated experimental data and metadata integration and storage, irrespective of data type. The amount of experimental information stored may be as much or as little of the FuGE standard as is required by end-users. The importance of utilizing standards in database and application development is growing as stringent, standardized, format requirements for published data are set by publishing and research bodies. Capturing data in a standard format is an essential part of simplifying integration efforts with external data sources, and helps to promote a systems-level approach to research. With the realization of a complete start-to-finish implementation FuGE database, STK and web interface, the CISBAN DPI is at the forefront of data standards implementation.

6 Future Plans

The CISBAN DPI's successful implementation of FuGE shows how straightforward and practical the use of such an all-encompassing standard can be. The HT data sets and their associated experimental information will be crucial inputs in a variety of bioinformatics research projects within the Centre. We plan to expand the web front-end to feature more complex search functions and to conform to community minimum reporting requirements. Additionally, using terms from ontologies such as the Ontology of Biomedical Investigations (OBI)[14] will add richness to the stored experimental information as well as the potential to programmatically compare experiments. To cater for the large amount of automatically generated data, we will develop a batch loader to automatically load HT data sets into the DPI. It may also be possible to provide certain aspects of the DPI via applications such as Taverna[15]. We hope that other project with similar data management requirements will find this work useful and applicable to their own needs.

Acknowledgements

---

[14] http://obi.sourceforge.net

References

1. Edgar, R., Barrett, T.: NCBI GEO standards and services for microarray data. Nature biotechnology 24 (2006) 1471-1472
2. Jones, P., Cote, R.G., Martens, L., Quinn, A.F., Taylor, C.F., Derache, W., Hermjakob, H., Apweiler, R.: PRIDE: a public repository of protein and peptide identifications for the proteomics community. Nucleic acids research 34 (2006) D659-663
3. Kerrien, S., Alam-Faruque, Y., Aranda, B., Bancarz, I., Bridge, A., Derow, C., Dimmer, E., Feuermann, M., Friedrichsen, A., Huntley, R., Kohler, C., Khadake, J., Leroy, C., Liban, A., Lieftink, C., Montecchi-Palazzi, L., Orchard, S., Risse, J., Robbe, K., Roechert, B., Thorneycroft, D., Zhang, Y., Apweiler, R., Hermjakob, H.: IntAct--open source resource for molecular interaction data. Nucleic acids research 35 (2007) D561-565
4. Parkinson, H., Kapushesky, M., Shojatalab, M., Abeygunawardena, N., Coulson, R., Farne, A., Holloway, E., Kolesnykov, N., Lilja, P., Lukk, M., Mani, R., Rayner, T., Sharma, A., William, E., Sarkans, U., Brazma, A.: ArrayExpress--a public database of microarray experiments and gene expression profiles. Nucleic acids research 35 (2007) D747-750
5. Jones, A.R., Miller, M., Aebersold, R., Apweiler, R., Ball, C.A., Brazma, A., DeGreef, J., Hardy, N., Hermjakob, H., Hubbard, S.J., Hussey, P., Igra, M., Jenkins, H., Julian, Jr. R.K., Laursen, K., Oliver, S.G., Paton, N.W., Sarkans, U., Sansone, S., Stoeckert, Jr. C.J., Taylor, C.F., Whetzel, P.L., White, J.A., Spellman, P. and Pizarro, A.: The Functional Genomics Experiment model (FuGE): an extensible framework for standards in functional genomics. Nature Biotechnology in community consultation (2007)
6. Clark, T., Martin, S., Liefeld, T.: Globally distributed object identification for biological knowledgebases. Briefings in bioinformatics 5 (2004) 59-70
7. Gudgin, M., Hadley, Marc , Mendelsohn, Noah , Moreau, Jean-Jacques, Nielsen, Henrik Frystyk SOAP Version 1.2 Part 1: Messaging Framework. Vol. 2007 (2003)
8. Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T., " Hypertext Transfer Protocol -- HTTP/1.1. RFC 2616 (June 1999)
9. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C.A., Causton, H.C., Gaasterland, T., Glenisson, P., Holstege, F.C., Kim, I.F., Markowitz, V., Matese, J.C., Parkinson, H., Robinson, A., Sarkans, U., Schulze-Kremer, S., Stewart, J., Taylor, R., Vilo, J., Vingron, M.: Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. Nature genetics 29 (2001) 365-371
10. Taylor, C.F.: Minimum Reporting Requirements for Proteomics: A MIAPE Primer. Proteomics 6 Suppl 2 (2006) 39-44
11. Hermjakob, H.: The HUPO Proteomics Standards Initiative - Overcoming the Fragmentation of Proteomics Data. Proteomics 6 Suppl 2 (2006) 34-38
12. Spellman, P.T., Miller, M., Stewart, J., Troup, C., Sarkans, U., Chervitz, S., Bernhart, D., Sherlock, G., Ball, C., Lepage, M., Swiatek, M., Marks, W.L., Goncalves, J., Markel, S., Iordan, D., Shojatalab, M., Pizarro, A., White, J., Hubley, R., Deutsch, E., Senger, M., Aronow, B.J., Robinson, A., Bassett, D., Stoeckert, C.J., Jr., Brazma, A.: Design and implementation of microarray gene expression markup language (MAGE-ML). Genome biology 3 (2002) RESEARCH0046
13. Moats, R.: URN Syntax. RFC 2141 (December 1997)
14. Berners-Lee, T., Masinter, L., McCahill, M. : Uniform Resource Locators (URL). RFC 1738 (December 1994)
15. Hull, D., Wolstencroft, K., Stevens, R., Goble, C., Pocock, M.R., Li, P., Oinn, T.: Taverna: a tool for building and running workflows of services. Nucleic acids research 34 (2006) W729-732