Globally Distributed Object Identification for Biological Knowledge Bases

Globally distributed object identification for biological knowledgebases

Tim Clark, Sean Martin and Ted Liefeld

Date received (in revised form): 13th December 2003

Tim Clark
led development of the integrated bioinformatics, cheminformatics and knowledge engineering programmes at Millennium Pharmaceuticals from their inception. He was previously at the NCBI, where he helped develop GenBank.

Sean Martin
is a Senior Technical Staff Member with the Advanced Technology Group at IBM Corporation.

Ted Liefeld
is the Senior Software Architect for Cancer Genomics at the Broad Institute. He was previously at Millennium Pharmaceuticals where he helped define Millennium's software integration architecture and led knowledge management and informatics projects.

Abstract

The World-Wide Web provides a globally distributed communication framework that is essential for almost all scientific collaboration, including bioinformatics. However, several limits and inadequacies have become apparent, one of which is the inability to programmatically identify locally named objects that may be widely distributed over the network. This shortcoming limits our ability to integrate multiple knowledgebases, each of which gives partial information of a shared domain, as is commonly seen in bioinformatics. The Life Science Identifier (LSID) and LSID Resolution System (LSRS) provide simple and elegant solutions to this problem, based on the extension of existing internet technologies. LSID and LSRS are consistent with next-generation semantic web and semantic grid approaches. This article describes the syntax, operations, infrastructure compatibility considerations, use cases and potential future applications of LSID and LSRS. We see the adoption of these methods as important steps toward simpler, more elegant and more reliable integration of the world's biological knowledgebases, and as facilitating stronger global collaboration in biology.

Keywords: *knowledgebase, interoperability, semanic web, LSID, identity resolution, database*

INTRODUCTION

Bioinformatics traces its early growth as a discipline to the construction of the first databases of DNA and protein sequences,[1–5] and of programs to search and analyse their contents.[6–10] With time and improvements in molecular technology, the number, range and content of such databases increased dramatically. With this development, came an increased need to cross-reference, compare and share data among the databases.[11]

The web also increased the scientific audience size, and therefore potential usefulness, of valuable specialist databases developed by single individuals or small groups. Some of the key biological knowledge resources available today, such as Swiss-Prot[12,13] and its derivatives,[14,15] began in this way.

From the outset, however, biologists attempting to use these knowledge resources were faced with incompatible object identifiers, data formats and interfaces, because they were developed and assigned locally. Various workshops and ongoing committees attempted to introduce some order.[16] The identifier problem has, however, never been satisfactorily solved at the level of infrastructure, but was left instead to the individual programmer to work out – repeatedly.[17]

Genome-era bioinformatics is now completely dependent upon the advanced technologies of the web. However, integration of multiple knowledge resources has so far remained *ad hoc* and labour-intensive. Functional genomics places even greater demands on the current infrastructure, because of the depth, diversity and distribution of knowledge, which can now appear as functional annotation on any given DNA, protein, RNA or other object. Post-genome bioinformatics requires far simpler, more transparent integration, so that the totality of biological knowledgebases can be effectively viewed as a single resource, by programs as well as by individuals.[18,19]

Tim Clark,
Newton,
MA 02459,
USA

Tel: +1 617 947 7098
E-mail: tim.clark@acm.org

Ontological and technical forms of agreement required for database integration

There are two fundamental forms of agreement required for integrating databases. These are, agreement upon names and the meanings of common object attributes in the domain ('dictionaries', 'descriptors', 'controlled vocabulary', 'ontology'), and agreement upon unique object identifiers.

Independently managed databases must agree, where they overlap, on shared name definitions and the interrelationships of their object attributes, if integration is to be possible. Canonical structured term sets of this nature are called ontologies, of which MeSH,[20] UMLS[21] and GO[22] are prominent examples in bioinformatics and medical informatics. Ontologies are a generalisation of the specialised controlled vocabulary or discriminator list, which may include synonyms, hierarchy, mesh or network structure, and named relationships ('noun' and 'verbs'). Sets of agreed referents for object description are necessary for very common-sense reasons – to converse we must share a common language, however simple.

LSID provides a framework for meeting these requirements

LSID is based on existing IETF and W3C technology

Independent databases must also agree on unique object identification. One and only one object may be specified (and be resolvable by) any identifier – or it is not an 'identifier' at all. Naturally, we can only guarantee this property for a name space under our control. We are not so concerned whether identifiers have synonyms, although we should like a means to express clearly when synonym relationships exist.

For biological objects and their attributes, we want to be able to make agreements such as: 'hypothalamus' is a kind of brain tissue, the 'brain' is an 'organ' and part of the 'CNS', containing both 'glia' and 'neurons' of various descriptions. We should be able to agree that '*Mus musculus*' is a species of 'mouse', and that 'C57/B6' is a strain of '*Mus musculus*'. And for identifying unique individual *Mus musculus*, we should be able to agree that in referring to a particular C57/B6 mouse by some identifying number, we do not become confused about which mouse we mean. If we determine this particular *Mus* to have a mutation in its *Apoe* gene, we would also all want to recognise that '*Apoe*' names the gene for a protein 'apolipoprotein E', identified as 'MGI:88057' by Jackson Labs – but that this was also synonymous with locus NM_009696 in NCBI RefSeq.

Fundamentally, we must be able to solve the following problems to fully integrate web-distributed databases: (a) define the link interfaces formally so that they may be understood programmatically; (b) encapsulate the link interfaces so that they are not addresses, but names; (c) locally specify and control object identifiers while guaranteeing them to be globally unique; (d) describe the object attributes using a formal ontology.[23]

This paper will present Life Science Identifiers (LSIDs) and the LSID Resolution System (LSRS) as the most useful system evolved to date for meeting these requirements. It is based on existing IETF and W3C technologies, with some judicious extension, while being compatible with web services,[24] semantic web[25] and grid services[26,27] models. We claim it fully meets requirements (a)–(c) above, while meeting requirement (d) for the important set of attributes called identifiers – because it provides a hierarchy of Namespaces wherein identities are defined.

DISTRIBUTING IDENTITY: LSID SYNTAX AND SPECIFICATION

Tim Berners-Lee[28] said in 1994:

> The web is considered to include objects accessed using an extendable number of protocols, existing, invented for the web itself, or to be invented in the future. Access instructions for an individual object under a given protocol are encoded into forms of address string. . .the web needs the concepts of the universal set

W3C convention identifies objects on the web by name or by address

of objects, and of the universal set of names or addresses of objects.

These names-or-addresses of objects are formalised as Universal Resource Identifiers (URIs), which in turn are specialised into URLs (Universal Resource Locators) and URNs (Universal Resource Names). URLs are location-dependent object references. URNs are location-independent references, ie persistent object names. URNs take the form

URN identification-by-name provides for subspaces of names

⟨URN⟩ : : = 'urn:' ⟨NID⟩ ':' ⟨NSS⟩

where ⟨NID⟩ is a Namespace Identifier and ⟨NSS⟩ is a Namespace Specific String. An NSS will be resolved as specified by the namespace resolution protocol. The registration document for the namespace will specify, among other matters, how to determine functional equivalence of URNs (synonymy).[29–32]

LSID is a specialised URN namespace

LSIDs are a special form of URN. As such, they have their own resolution protocol, and are persistent, global, location-independent object names. Location independence is important: for example, the file path for a database version may be changed without affecting the ability to resolve the resource. LSIDs may be used to persistently name such resources as individual proteins or genes, transcripts, experimental data sets, annotations, ontologies, publications, biological knowledgebases or objects within them.[33]

The syntax of an LSID is:

⟨LSID⟩ : : = 'urn:' 'lsid:' ⟨AuthorityID⟩ ':' ⟨AuthorityNamespaceID⟩':' ⟨ObjectID⟩[':' ⟨RevisionID⟩]

So, in any LSID, the URN NID will be 'lsid'. The URN NSS is the business end of an LSID. It will identify an authority (eg 'ncbi.nlm.nih.gov') which assigns life science identifiers to objects; an authority-specific namespace within which the identifier lives (eg 'refseq'); a unique object id; and, optionally, a revision ID. To a URN-level resolver, the NSS is opaque. To an LSID resolver, the NSS has syntactic structure in that it specifies the authority and intra-authority namespace; but the LSID's ⟨objectID⟩ itself is opaque. In the TIGR identifier AT1G67550, the first two characters may well signify *Arabidopsis thaliana* according to TIGR's local convention, but for LSID purposes they are opaque – there is no syntactic structure within an LSID's ⟨objectID⟩ that a resolver cares about. An LSID specifying this *Arabidopsis* locus name would look something like this:

URN:LSID:tigr.org:
AT.locusname:AT1G67550

LSIDs are case-insensitive in the urn:lsid:⟨authorityid⟩: portion of the identifier, but the remainder of the string (⟨namespace⟩,⟨objected⟩,⟨revisionid⟩) *is* case sensitive. This is compatible with Resource Description Framework (RDF), which uses equivalence rules as defined in the URI standard. For URNs (a subset of URIs) this equivalence is left to the namespace owner for the NSS (namespace-specific-string).[32] Existing web URLs are also case sensitive for the path portion.

This syntax creates a URN-conformant namespace, divided into sub-namespaces by Authority, that is, by the groups responsible for issuing and controlling the identifiers. The LSID scheme conforms very well to the distributed organisational set-up prevailing among biological knowledgebase providers. It creates a natural distribution of the authority for creating and maintaining identifiers. Providing a sufficient number of database providers adopt this approach, which has a low barrier to entry (see below), we can have the ability to resolve globally what has been defined locally.

Any organisation assigning LSIDs has several responsibilities:

- It must identify itself within the NSS using its AuthorityID, typically an Internet domain name it owns.
- It must ensure the uniqueness of the

LSIDs may represent either 'real' concrete objects, or concepts

string created from the namespace, object and revision identifications within any given authority's domain.
- If the AuthorityID is not an internet domain name, the organisation must ensure it is a globally unique string – a very good reason for using self-owned domain names in the first place. Registration of the AuthorityID with the owner of the LSID NID namespace owner guarantees this uniqueness.

LSIDs are permanent

Here are further examples of well-formed LSIDs:

- URN:LSID:ebi.ac.uk:SWISS-PROT.accession:P34355:3
- URN:LSID:rcsb.org:PDB:1D4X:22
- URN:LSID:ncbi.nlm.nih.gov:GenBank.accession:NT_001063:2

A formal identity resolution protocol is defined for these objects

Once assigned, an LSID is permanent and is never reassigned. Furthermore, as unique identifiers, they can specify only one object – not just at a moment in time, but for all time. Assignment of multiple LSIDs to the same object is not recommended, but is legal within the specification. When an LSID names an object that has a byte representation, these exact bytes are always what is named by the LSID. A change of even a single bit in the underlying object named by an LSID should result in a new LSID name for this new object. It is recommended that the new LSID be based on the original LSID, but contain an increment to the content of the RevisionID field, resulting in a new unique name. The Authority of any particular LSID is under no obligation always to be able to provide a copy of the object named that in the future. They are, however, under an obligation to provide an error and not some other object if that LSID is resolved and accessed, without the proper original underlying object. (The LSID/LSRS cannot guarantee, by itself, that an LSID always resolves to the same data, without the effective cooperation of the data provider. This system is one of distributed responsibility.)

LSIDs may alternatively represent abstract entities or concepts. The LSID resolution service in this situation will resolve `getdata()` to an empty result; `getMetadata()`, however, will be non-empty.

For all LSID-named data objects, whether abstract or concrete, any data retrieval service defined on the object will resolve to the same bitstream using `getdata()`. But `getMetadata()` when called against different retrieval services is allowed to resolve to varying metadata for the same LSID.[34]

RESOLVING IDENTITY: THE LSRS RESOLVER AND RESOLUTION DISCOVERY

Objects identified by LSIDs may be mapped, through LSID Resolution, to services implemented on the object by the object-providing authority. These can include various forms of object metadata, multiple-protocol object data retrieval, and potentially other services.

Resolution works as follows:

- A client has an LSID and knows the appropriate Resolution Service, or uses the LSID Resolution Discovery Service to find the appropriate resolver.
- The client sends the method `getAvailableServices` and the LSID to the appropriate resolution service.
- The resolution service returns information on what services can be provided on the LSID, where they are located and how to call them.
- The client calls the desired Data Retrieval Services (DRS), using `getData` and/or `getMetadata`.
- The service executes the requests and returns results to the client.

In the current RDS implementation, `getData` and `getMetadata` have been mapped to older web protocols such as ftp[35] and http[36] as well as to SOAP[37–39] (see below) web services for data retrieval; additional mappings are possible.

Identity resolution is bootstrapped through Resolution Discovery

Resolution Discovery is a bootstrapping process for clients either (a) not knowing the resolver service for a particular LSID, or (b) wishing to get the most current information on all available services (ie checking for new or modified services) before proceeding. When passed an LSID, it returns URLs of the appropriate Resolution Services.

Figure 1 illustrates the relationship of LSIDs to Resolution Services and Resolution Discovery Services in a Universal Modelling Language (UML) model as jointly proposed by the European Bioinformatics Institute, IBM and the I3C consortium.[34] The LSID/LSRS system as described here is implemented in open source code by the LSID Resolution Protocol Project[40] and freely available for download.

INFRASTRUCTURE COMPATIBILITY

Compatible infrastructure means implementing LSIDs does not require extensive effort

Providers of life sciences information should have very little additional work to name and serve out their own data using an LSID handle. This is because the LSID resolution process builds directly upon existing best practices, technical standards and infrastructures that are already widely understood and deployed by the life sciences community. Wherever possible the resolution protocol also details the smallest set of features that an LSID authority should provide, while establishing a clear technical path for making such a service incrementally more sophisticated and useful.

Much of the life sciences information that might be usefully named by LSID is already made publicly available via WWW server software supporting access protocols such as ftp and http. These two protocols were consequently adopted as the means by which most data named by LSID would be retrieved. We expect that in many cases, providers of data over the internet will be able to supply that same data, named using an LSID, with only minor extensions to the configurations of their existing web-serving operation. An immense amount of software capable of

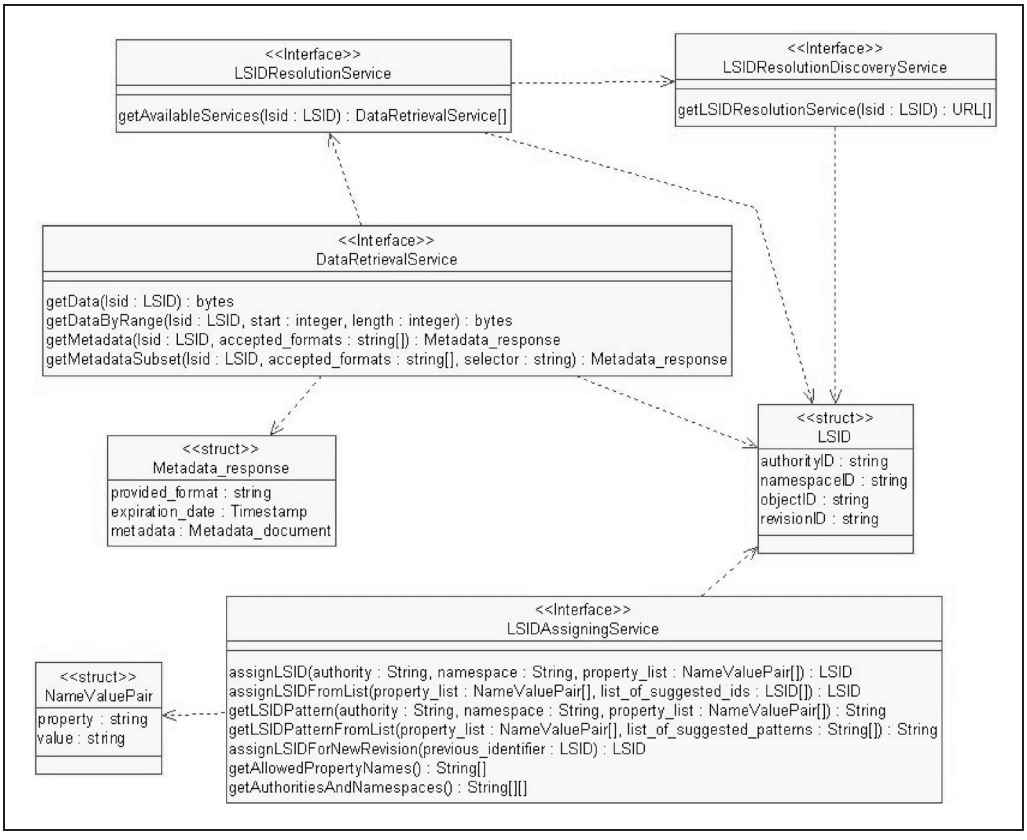

Figure 1: UML diagram showing LSID/LSRS components and their relationships
Adapted from EMBL-EBI, I3C, IBM[34]

supporting these communication protocols is available as both open source and commercially supported packages. In particular, the http protocol together with SSL[41,42] encryption is widely used to support electronic commerce and can similarly be employed with LSID for securing access to information only to authorised users.

The Resolution Protocol itself utilises existing SOAP and WSDL standards

The LSID resolution protocol utilises two important web services[43] standards for the discovery and specification of the sources of data and metadata for any object named by an LSID. The two now most widely adopted industry web service standards are SOAP, an open application based on extensible mark-up language (XML) for application networked communication protocol, and WSDL,[44] an XML standard for defining and communicating a SOAP accessible remote application programming interface (API). The current LSID specification allows for LSID named data and metadata retrieval using the SOAP protocol if a provider wishes to go beyond what the simpler http and ftp access protocols allow. Wide cross-industry support for web services has led to a great deal of new software and technical literature emerging over the past few years to sustain its ongoing deployment. This support includes protocol stacks for most programming languages and web-serving environments and advanced software development tools, both in open source and in commercially supported packages.

Organisations may become LSID naming authorities by adding a record to their DNS server

Before resolution can be performed on any LSID, a Resolution Service with knowledge of where that particular LSID can be accessed must be discovered. This is done by the Resolution Discovery Service. The current open source implementations[40] of LSID Resolution Discovery makes use of the existing Domain Name System (DNS)[45] for this purpose. (Using the DNS for Resolution Discovery, while effective and powerful, is not mandated by the standard, and other approaches are possible.) The IETF RFC (Request for Comments) archive details a number of standards that were designed precisely for such purposes. Service records also known as SRV[46] records were provided for network service discovery using the DNS and the Dynamic Delegation Discovery System (DDDS)[47] specification describes how any URN (of which the LSID is a subclass) might be resolved to the object it names, using the DNS configured to perform DDDS.

An open source implementation uses the Domain Name System (DNS) for Resolution Discovery

The great advantage of using DDDS for Resolution Discovery is that two and potentially more separate directory registries may be used to store the lists of authorities for all LSIDs. In the first instance the authority portion of an LSID can be a unique string registered with the owner of the LSID URN namespace, and if the authority string is not found there, the general DNS registries can be consulted by default to determine if the string is in fact a registered domain name. In either case, the mechanism will provide an IP address for the networked whereabouts of the correct Resolution Service for any LSID that can be contacted to continue the resolution process.

Organisations wishing to establish themselves as an authority for data named by an LSID can choose either to add a single line text record to their own DNS server, similar to adding a host entry using their existing domain name, or to externally register their unique authority string with the central owner of the LSID namespace. In either case, the process will be a quick and simple operation, utilising existing resources and skills. DNS server software is widely available and over the years has proven exceptionally stable and scalable. The DNS is by far the largest; most heavily trafficked, and most widely distributed registry system ever deployed.

One design factor that may turn out to be extremely important to the adoption of the LSID resolution scheme turns on the use of either web or web service protocols for the retrieval of data and metadata named by LSID. In the case of the web, there already exist many gateways and mapping schemes between http access and the underlying storage systems for data. Similarly, web services have characteristics

Access to LSID-identified data can be provided wing web services' 'wrappers'

that make them highly suitable for implementation as a 'wrapper' over pre-existing storage mechanisms or access APIs. Thus the adoption of these protocols means that existing data objects and their metadata can remain in the exact same databases and database schemas or other storage mechanisms for the purposes of a LSID Resolution Service. This very significantly reduces the work of an adopting data provider who will not have to reorganise their data or provision additional databases in order to provide access to their information by LSID. In most deployments the LSID Resolution Service will be another relatively thin layer of software built to operate alongside or on top of the existing serving infrastructure.

Metadata may be defined using RDF

While the format in which to provide additional metadata associated with any LSID has been left open in the LSID specification proposed by EMBL-EBI, I3C and IBM to the Object Management Group (OMG)[34] (an industry standards body), the current open source implementation favours the use of the XML form of RDF[48] with extensive software support. Data providers with metadata can choose to provide that information in its existing 'native' format or they may choose to translate it to XML RDF.

RDF defined metadata supports compatibility with emerging Semantic Web approaches

The use of RDF has many advantages, one of which is that RDF can be used to describe most ontologically encoded information without undue effort. However, information represented in (written to) XML schemas requires an XSLT (Extensible Stylesheet Language Transformation) or other translation. Construction of such XSLTs has been demonstrated to be feasible. Important design considerations here are (a) choosing an ontological representation that integrates well with other sources, and (b) making first-class objects of formerly anonymous objects in the original XML by specifying URIs for them. An example of such an XSLT exists in the I3C's NCBI database LSID server.[49]

Another advantage of RDF is that it was designed to allow the easy merging of information from many sources. This may prove especially interesting to the life sciences community as LSIDs are adopted more widely. Much of the work in life sciences hinges on discovering and recording the relationships between pieces of information. Since RDF can use URIs to identify subjects, objects and their relationships and because LSIDs are a subclass of URIs, they mesh very well with RDF.

Metadata returned as RDF and retrieved from many different sources can use the unique LSID names for life science objects and concepts as a handle on which to hook, with certainty, annotations on information objects that are related to one another. RDF is also extremely good at describing relationships between objects. The merging characteristics of RDF will allow the relationships in the information from many sources to be automatically discovered and that information to be combined for visual presentation to a user or used to perform tasks without human intervention. For example, the automated negotiation with an information store for a piece of data in a particular format with a particular semantic type is made much simpler with RDF.[50–52]

As more life science data providers supply RDF metadata information describing the objects in their information store and the relationships between those objects and other objects (also named by LSID) both in their store and in other life science data stores, the richer the web of immediately accessible interrelated information becomes. Current research including software for specialised RDF databases, parsing, visualisation, ontology creation tools is ongoing and available through the W3C's Semantic Web[53] project and related academic and commercial research efforts.

APPLICATIONS

Identifying identifiers and unpacking them

Current bioinformatics applications and databases each have unique formats for

Parsing and identifying is simplified

Negotiation of data format compatibility is simplified

Service interfaces may be discovered automatically

the identifiers that they generate and maintain. In order to integrate disparate applications it is necessary for bioinformatics developers to include code to parse and identify the identifiers. This problem is made thornier by the fact that identifiers can often not be recognised simply by looking at the identifier itself out of context, eg is GI000197 a GenBank accession number or a GI (GenInfo) number? Syntax is insufficient, since the syntactic format (letter-letter-#### looks like an accession number). Semantics of the identifier or the context of its use needs to be known to be able to computationally determine the nature of the identifier. If we were to write this identifier as an LSID instead, urn:lsid:genbank.ncbi.nih.gov:genbank.gi:000197, then it becomes immediately recognisable to a program: that it is looking at an identifier; what kind of identifier it is; and what authority to contact for resolution, methods queries and so forth.

Identifiers as external object references

One typical use for an LSID is to act as a reference to an external database. For example, we might use it to relate a sequence stored in an internal database to a GenBank record stored at NCBI. Simply storing this relationship in an internal database provides some minimal value but to use it effectively, a scientist would need to retrieve the GenBank record that is represented by the identifier. For this action, the LSID resolver may be used to retrieve the GenBank record. In the case of an academic institution, the resolver could retrieve the record directly from NCBI. In the case of a private company, the systems administrators, if they desire, can configure their systems to retrieve GenBank records from a copy stored inside their firewall.

Negotiation of compatible data formats

An end-user program that chooses to retrieve data from a source (local or remote) using its LSID will be able to obtain metadata about the form the data will take. This allows the client application fetching the object/data referred to by an LSID to negotiate with the service to get it in a format that it understands. For example, many authorities for scientific data such as the Protein Data Bank (PDB) publish data about one biological object (eg a protein) in many different formats (eg HTML, FastA and XML). If we use an LSID to retrieve data to input to an analysis application, the application can interact with the LSID stack using the metadata query routines to identify what data formats are available from the source for the biological object represented by the LSID. The application may then examine the list of data formats and issue a new request to the data source to retrieve that data in the format it prefers. The purpose of LSID resolver metadata queries is to allow the computer to accomplish this task without human intervention. To put this in a concrete example, we should not require the bioinformatician to convert the data they wish to use from a GenBank file to a FastA file before giving it to BLAST if the data are already available in FastA format from the original data source.

Service interface discovery and virtual piping

Web services (eg SOAP services) taking an LSID URN as one of the service method calling parameters will facilitate virtual piping between applications when used in conjunction with service registries such as BioMOBY[54] or UDDI.[55] The applications will know only what they want done, and will use registries to discover services capable of accomplishing the desired actions. These actions may form a series. If the service-implementing programs are LSID-aware, they can negotiate data format compatibility with the data provider, and therefore will be able to load provider data directly into their own address space, rather than transferring it through the client's address space. Issuing an output-data LSID handle to the client will then enable 'virtual

piping' to successively called web services without impacting client address space or communication bandwidth.

Identifier permanency and location – independence support reproducible research on the web

Reproducible research

Use of LSIDs can facilitate the publication of 'reproducible research'. Reproducible research is a movement that espouses having scientific articles publish their data and analysis modules in a manner that allows scientists at other sites to reproduce all computations and analyses exactly, using the same data sets and algorithms. Buckheit and Donoho[56] state the essential principle of reproducible research as:

> An article about computational science in a scientific publication is not the scholarship itself, it is merely advertising of the scholarship. The actual scholarship is the complete software development environment and that complete set of instructions that generate the figures.

Publication of metadata using LSID provides a foundation for Semantic Web in biology

Several institutions have successfully implemented this technology

The LSID and LSID Resolution Service are ideally suited to support this approach. For example, current practice would involve publishing the data sets and analysis pipeline (eg an R script) associated with a scientific publication on a web page. However, authors change their affiliations and web servers change addresses as well. By publishing the LSIDs to the data sets and the LSID to the analysis software, the LSIDs remain valid even when an author moves their publication web page to a different institution or if the data move to a different web server. In addition, the provision for versioning in an LSID would allow the author to publish improvements and corrections to their data and analysis pipeline in a manner that would still allow readers to retrieve the originally published version or to retrieve the newest version with fixes or extensions at their choice.

Semantic web for biology

As authorities begin publishing more extensive metadata for their LSIDs, the promise of the semantic web begins to be realised. The semantic web is defined by Berners-Lee *et al.*[25] as 'an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation.'

Researchers in the bioinformatics and semantic web communities have begun discussing and exploring the notion of a 'semantic web for biology', or for a subspace of biology.[57–59] The LSID/LSRS system is a foundational step towards providing a semantic web for biological computing because it provides locally defined identifiers, and the objects, concepts and metadata to which they refer, a global resolution capability and, with it, the ability to establish rigorously well-defined meaning.

Promising advances in providing such services have been made by various knowledgebase providers already. For example, the PDB authority offers a metadata service that generates comprehensive RDF relating LSIDs to each other, and providing links to external resources.[60] The University of Wisconsin North Temperate Lakes Long term Ecological Research (NTL-LTER) project also provides an LSID authority using LSID to serve their data sets as a web of data and metadata via RDF exposed through the LSRS.[61] Distributed Annotation Service (DAS),[62] a system for exchanging annotations on genomic sequence data based at Cold Spring Harbor Laboratories, has announced that they will support the LSID in their next software version.[23] BioMOBY, a service registry system for biological web services, recently introduced support for access to service descriptive metadata using LSIDs. And the innovative UK-funded project myGrid[63] is also planning to use LSIDs in its infrastructure.[64]

LSID Resolution will become increasingly useful, as more providers use LSIDs to name and expose their data. LSID Resolution Services exist for many well-known life sciences data sources including PDB, GenBank, PubMed,

Swiss-Prot, GeneOntology, LocusLink and ENSEMBL. With a relatively modest amount of integration, any organisation may now provide its existing databases via LSID. This will be a significant advance over the current smorgasbord of identifier formats, will reduce the burden of maintenance coding, and can become foundational infrastructure for a globally integrated semantic web of biological knowledge.

As more authorities begin to publish LSIDs and associated metadata for their data sets, we can see the foundations of a biological semantic web emerge, allowing computational and knowledge mining tools to automatically search, sort, compute upon and discover knowledge based on the relationships that have been published in the metadata by multiple authorities.[65]

SUMMARY AND CONCLUSION

Bioinformatics depends upon a global collaboration framework, which in turn depends upon a successful global identifier resolution

Genome-era bioinformatics, we repeat, is absolutely dependent upon the web as a global collaboration framework. This framework has the potential of unifying and sharing all biological knowledge as it emerges, driving an increasingly productive social organisation of science. Globally resolvable object identifiers are of fundamental concern in this context. In the future technological environment, identifying an object on the web with global resolution will allow us to know what kind of object it is, who originated it, who is responsible for it, how to interface to it and what computations might be carried out on it. And in saying 'allows *us* to know', we really also refer to our agents, the various computer programs upon which we so depend. We will remove a significant impediment to their work as we extract people from their tedious parsing and navigation among them. All the early difficulties among various databases about compatibility of interfaces and data formats, left for years to the ingenuity of individual programmers to solve, are distilled in the problem of global identifier resolution. It is now possible to solve this problem.

Acknowledgments
The authors are grateful to Philip Werner and Rainer Fuchs for helpful discussions during the writing of this paper.

References

1. Dayhoff, M. O., Schwartz, R. M., Chen, H. R. *et al.* (1980), 'Nucleic acid sequence bank', *Science*, Vol. 209(4462), p. 1182.
2. Kneale, G. and Bishop, M. (1985), 'Nucleic acid and protein sequence databases', *Comput. Appl. Biosci.*, Vol. 1(1), pp. 11–17.
3. Burks, C., Fickett, J. W., Goad, W. B. *et al.* (1985), 'The GenBank nucleic acid sequence database', *Comput. Appl. Biosci.*, Vol. 1(4), pp. 225–233.
4. Hamm, G. H. and Cameron, G. N. (1986), 'The EMBL Data Library', *Nucleic Acids Res.*, Vol. 14, pp. 5–9.
5. George, D. G., Barker, W. C. and Hunt, L. T. (1986), 'The protein identification resource (PIR)', *Nucleic Acids Res.*, Vol. 14(1), pp. 11–15.
6. Smith, T. F. and Waterman, M. S. (1981), 'Identification of common molecular subsequences', *J. Mol. Biol.*, Vol. 147(1), pp. 195–197.
7. Kanehisa, M. I. (1982), 'Los Alamos sequence analysis package for nucleic acids and proteins', *Nucleic Acids Res.*, Vol. 10(1), pp. 183–196.
8. Wilbur, W. J. and Lipman, D. J. (1983), 'Rapid similarity searches of nucleic acid and protein data banks', *Proc. Natl Acad. Sci. USA*, Vol. 80(3), pp. 726–730.
9. Pearson, W. R. and Lipman, D. J. (1988), 'Improved tools for biological sequence comparison', *Proc. Natl Acad. Sci. USA*, Vol. 85, pp. 2444–2448.
10. Altschul, S. F., Gish, W., Miller, W. *et al.* (1990), 'Basic local alignment search tool', *J. Mol. Biol.*, Vol. 215, pp. 403–410.
11. Smith, T. F. (1990), 'The history of the genetic sequence databases', *Genomics*, Vol. 6(4), pp. 701–707.
12. Bairoch, A. and Boeckmann, B. (1991), 'The SWISS-PROT protein sequence data bank', *Nucleic Acids Res.*, Vol. 19(Suppl), pp. 2247–2249.
13. Boeckmann, B., Bairoch, A., Apweiler, R. *et al.* (2003), 'The SWISS-PROT protein knowledgebase and its supplement TrEMBL in 2003', *Nucleic Acids Res.*, Vol. 31(1), pp. 365–370.
14. Gattiker, A., Michoud, K., Rivoire, C. *et al.* (2003), 'Automated annotation of microbial

proteomes in SWISS-PROT', *Comput. Biol. Chem.*, Vol. 27(1), pp. 49–58.

15. Gasteiger, E., Gattiker, A., Hoogland, C. *et al.* (2003), 'ExPASy: The proteomics server for in-depth protein knowledge and analysis', *Nucleic Acids Res.*, Vol. 31(13), pp. 3784–3788.
16. Keil, B. (1987), 'Databases in molecular biology: A CODATA task group at work', *Protein Seq. Data Anal.*, Vol. 1(2), pp. 123–126.
17. Clark, T. (2003), 'Identity and interoperability in bioinformatics', *Brief. Bioinf.*, Vol. 4(1), pp. 4–6.
18. Stein, L. D. (2002), 'Creating a bioinformatics nation', *Nature*, Vol. 417, pp. 119–120.
19. Boucelma, O., Castano, S., Goble, C. *et al.* (2002), 'Report on the EDBT 02 Panel on Scientific Data Integration', *ACM SIGMOD Record*, Vol. 31(4), pp. 107–112.
20. National Library of Medicine (2002), 'National Library of Medicine Fact Sheet: Medical Subject Headings (MeSH®)' (URL: http://www.nlm.nih.gov/pubs/factsheets/mesh.html).
21. National Library of Medicine (2003), 'National Library of Medicine Fact Sheet: Unified Medical language System®' (URL: http://www.nlm.nih.gov/pubs/factsheets/umls.html).
22. The Gene Ontology Consortium (2000), 'Gene ontology: Tool for the unification of biology', *Nature Genet.*, Vol. 25, pp. 25–29.
23. Stein, L. D. (2003), 'Integrating biological databases', *Nature Rev./Genet.*, Vol. 4, pp. 337–345.
24. Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. (2001), 'Web Services Description Language (WSDL) 1.1', W3C Note, 15th March (URL: http://www.w3.org/TR/wsdl).
25. Berners-Lee, T., Hendler, J. and Lassila, O. (2001), 'The semantic web', *Scientific American*, May (URL: http://www.scientificamerican.com/article.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21&catID=2).
26. Foster, I., Kesselman, C., Nick, J. M. and Tuecke, S. (2002), 'The Physiology of the Grid' (URL: http://www.globus.org/research/papers/ogsa.pdf).
27. Goble, C. and DeRoure, D. (2002), 'The grid: An application of the semantic web', *ACM SIGMOD Record*, Vol. 31, p. 4 (URL: http://www.semanticgrid.org/documents/sigmod/ami9.pdf).
28. Berners-Lee, T. (1994), 'Universal Resource Identifiers in WWW', IETF RFC1630 (URL: http://www.ietf.org/rfc/rfc1630.txt).
29. Berners-Lee, T., Masinter, L. and McCahill, M. (1994), 'Uniform Resource Locators,, IETF RFC1738 (URL: http://www.w3.org/Addressing/rfc1738.txt).
30. Fielding, R. (1995), 'Relative Uniform Resource Locators', IETF RFC1808 (URL: http://www.w3.org/Addressing/rfc1808.txt).
31. Moats, R. (1997). 'URN Syntax', IETF RFC2141 (URL: http://www.ietf.org/rfc/rfc2141.txt).
32. Berners-Lee, T., Fielding, R. and Masinter, L. (1998), 'Uniform Resource Identifiers (URI): Generic syntax', IETF RFC2396 (URL: http://www.ietf.org/rfc/rfc2396.txt).
33. Werner, P., Liefeld, T., Gilman, B. *et al.* (2002), 'URN Namespace for Life Science Identifiers' (URL: http:// www.i3c.org/wgr/ta/resources/lsid/docs/LSIDSyntax9-20-02.htm).
34. EMBL-EBI, I3C, IBM (2003), 'Life Science Identifiers RFP Response Revised Joint Submission', Object Management Group Document lifesci/2003–10–01, 27th October (URL: http://www.omg.org/cgi-bin/doc?lifesci/2003–10–01).
35. Postel, J. and Reynolds, J. (1985). 'File Transfer Protocol (FTP)', IETF RFC959 (URL: http://www.ietf.org/rfc/rfc0959.txt).
36. Lafon, Y. (2003), 'W3C HTTP – Hypertext Transfer Protocol', W3C.ORG (URL: http://www.w3.org/Protocols/).
37. Mitra, N., Ed (2003), 'SOAP Version 1.2 Part 0: Primer', W3C Recommendation 24th June (URL: http://www.w3.org/TR/soap12-part0).
38. Gudgin, M., Hadley, M., Mendelsohn, M. *et al.* (2003), 'SOAP Version 1.2 Part 1: Messaging Framework', W3C Recommendation, 24th June (URL: http://www.w3.org/TR/2003/REC-soap12-part1–20030624/).
39. Gudgin, M., Hadley, M., Mendelsohn, M. *et al.* (2003), 'SOAP Version 1.2 Part 2: Adjuncts', W3C Recommendation 24th June (URL: http://www.w3.org/TR/2003/REC-soap12-part2–20030624/).
40. LSID Resolution Protocol Project (2003), (URL: http://ibm.com/developerworks/oss/lsid/).
41. Freier, A., Carlton, P. and Kocher, P. (1996), 'The SSL Protocol Version 3.0', IETF Transport Layer Security WG Internet-Draft (URL: http://wp.netscape.com/eng/ssl3/draft302.txt).
42. Rescorla, E. (2001), 'SSL and TLS: Designing and Building Secure Systems', Addison-Wesley, Boston, MA.
43. Haas, H. (2003), 'W3C Web Services Activity', W3C.ORG (URL: http://www.w3.org/2002/ws/).

44. Christensen, E., Curbera, F., Meredith, G. and Weerawarana, S. (2001), 'Web Services Description Language (WSDL) 1.1', W3C Note, 15th March (URL: http://www.w3.org/TR/2001/NOTE-wsdl-20010315).
45. Salamon, A. (2003), 'DNS Related RFCs' (URL: http://www.dns.net/dnsrd/rfc/).
46. Gulbrandson, A., Vixie, P. and Esobov, L. (2000), 'A DNS RR for specifying the location of services (DNS SRV)', IETF RFC2782 (URL: http://www.ietf.org/rfc/rfc2782.txt).
47. Mealling, M. (2002), 'Dynamic Delegation Discovery System' (Parts One to Five), RFC3401, 3402, 3403, 3404, 3405 (URLs: http://www.ietf.org/rfc/rfc3401.txt, http://www.ietf.org/rfc/rfc3402.txt, http://www.ietf.org/rfc/rfc3403.txt, http://www.ietf.org/rfc/rfc3404.txt, http://www.ietf.org/rfc/rfc3405.txt).
48. Miller, E., Swick, R. and Brickley, D. (2003), 'W3C Resource Description Framework (RDF)' (URL: http://www.w3c.org/RDF/).
49. Quan, D. (2003), Personal communication.
50. Buttler, D., Coleman, M., Critchlow, T. *et al.* (2002), 'Querying multiple bioinformatics information sources: Can semantic web research help?', *ACM SIGMOD Record*, Vol. 31, Issue 4.
51. Quan, D. and Karger, D. (2004), 'How to make a semantic web browser' (URL: www.ai.mit.edu/people/dquan/www2004-browser.pdf).
52. Tyrell, G. and King, G. (2003), 'A platform for the description, distribution and analysis of genetic polymorphism data', in Yi-Ping Phoebe Chen, J., Ed, 'Proceedings, First Asia-Pacific Bioinformatics Conference, Adelaide, Australia', Conferences in Research and Practice in Information Technology, Vol. 19.
53. Miller, E., Swick, R., Brickley, D. *et al.* (2003), 'W3C Semantic Web' (URL: http://www.w3.org/2001/sw/).
54. Wilkinson, M. and Links, M. (2002), 'BioMOBY: An open source biological web services proposal', *Brief. Bioinf.*, Vol. 3(4), pp. 331–341.
55. UDDI.ORG (2000), UDDI Technical White Paper (URL: http://www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf).
56. Buckheit, J. and Donoho, D. L. (1995), 'Wavelab and reproducible research', in Antoniadis, A., Ed, 'Wavelets and Statistics', Springer-Verlag, Berlin, New York (URL: http://citeseer.nj.nec.com/buckheit95wavelab.html).
57. Stevens, R. D., Robinson, A. J. and Goble, C. A. (2003), 'myGrid: Personalised bioinformatics on the information grid', *Bioinformatics*, Vol. 19 (Suppl 1), pp. i302–i304.
58. Quan, D. (2003), 'Client-side bioinformatics and RDF', Invited talk, I3C Technical Meeting, 6th May (URL: http://www.i3c.org/mtg/may03/presentations/quan.ppt).
59. Oliver, D. E., Hewett, M., Rubin, D. L. *et al.* (2001), 'Management of Data, Knowledge, and Metadata on the Semantic Web: Experience with a Pharmacogenetics Knowledge Base' Stanford Medical Informatics Research Reports SMI-2001–0901–1, 9th September (URL: http://smi-web.stanford.edu/pubs/SMI_Reports/SMI-2001–0901–1.pdf).
60. Protein Data Bank (2003) (URL: http://www.rcsb.org/pdb/).
61. NTL_LTER (2003) (URL: (URL: http://lsid.limnology.wisc.edu).
62. BIODAS (2003) (URL: http://www.biodas.org).
63. MyGrid Project (2003) (URL: http://www.mygrid.org.uk).
64. Li, P. (2003), 'Requirements for the information repository and management of information in myGrid', MyGrid Requirements Document, 2003-01-27 (URL: twiki.mygrid.org.uk/twiki/pub/Mygrid/MyGridInformationRepository/WP3Requirements0_2.pdf).
65. Guha, R., McCool, R. and Miller, E. (2003), 'Semantic search', in 'Proceedings of the Twelfth International Conference on World Wide Web, Budapest, Hungary, May 2003', ACM Press, New York.