Editorial: Identity and interoperability in bioinformatics

It was claimed some time ago that the Human Genome Project could not have been completed without Perl, as a ready language-based means for resolving interoperability problems. At a more fundamental level, if we accept this claim for Perl, we might then have a retrospective look at the total interoperability framework for scientific collaboration on genomics. In so doing we might ask, ‘what are the fundamental components?’ ‘what essential facilities do they provide?’ and ‘are they adequate for the next generation of post-genome research?’

From the informatics perspective, the basic components in our framework are, at least: the Internet, your favourite browser, an adequate collection of bioinformatics databases and algorithms, and Perl or some reasonable facsimile (eg Python). Internet, browser, objects, services, ‘glue’: that is our basic framework for constructing online scientific collaboration on a very large, distributed and data-intensive project.

The (mandatory) use of this framework in doing bioinformatics is now commonplace. It has receded almost into unconsciousness for the ordinary researcher, though as will be discussed, not for the bioinformatician. The ‘first draft’ publication of the human genome, for example, cites 20 or so distinct URLs in its Methods section; the words ‘internet’, ‘browser’, ‘URL’, ‘protocol’ do not appear. The facilities these words describe are as culturally invisible to the researcher as they are utterly pervasive. Neither individual researchers nor institutions could do or apply genomics without them.

But for the bioinformatician concerned with integrating and computing upon distributed information, our framework falls substantially short of adequacy. Its biggest weak spot is the browser (interoperability) – lovely for humans, a nightmare for program-to-program communication. In second place is perhaps naming (identity), with all the gloriously idiosyncratic embedded semantics of local identifiers in disparate forms.

The variant ad hoc approaches to object identifiers, the location-dependence of URLs, and the instability of any particular browser interface over time must be dealt with, or we will continue to invest too heavily in programmatic interfaces and their maintenance to properly track the science or to integrate broad enough sets of information. ‘Best practices’ suggestions intended to stabilise the browser interface for screen-scraping applications identify the correct problem. They are a positive step, but insufficient. Something more is needed if we are to construct an integrated and fundamentally interoperable ‘bioinformatics world’.

In essence, our collaboration framework must do a better job of supporting identity and interoperability. This is a consequence of our choice to distribute computational objects and services among many researchers and places, and our simultaneous need to collaborate and reason across the whole set. We must be able to name all the things (concepts, tools, services, data, conclusions, etc.) in our universe of discourse, in a way comprehensible to programs. Plus our programs must be able to compute on and talk with one another about the things named, with both reliability and stability in the face of constant change. These requirements have become more stringent with the transition now underway to post-genome bioinformatics.

Post-genome bioinformatics will focus on computational application of the wealth of information derived from the genome in multiple contexts, from basic research to...
clinical applications. In biopharmaceuticals, there will be a de-emphasis on novel target selection. Analysis of biochemical pathways, clinical candidate selection, drug mechanism-of-action determination, analysis of pharmacodynamic markers, and stratification of patient populations – the more so-called ‘downstream’ research applications of genomics will be emphasised. The movement will be towards using multi-perspective genomic information in a broadened biological and clinical context.5,6

Bioinformatics, therefore, requires more and more sophisticated programmatic interoperability at every step, because of the increasing simultaneous requirements to combine distributed information, to treat this information as foundational, and to track the science. Adequate interoperability today comes at a significant programming cost to those who must implement it. Meanwhile the science is moving rapidly and we would much rather use our programming resource to track it than rebuild application interfaces and parsers.

In the post-genomic era, a larger and larger portion of the value of any software application will consist of its ability to interoperate with other programs so that it reasons across a wider range of results. This additional value must be provided in a lightweight, agile way, to keep up with the science, which continually redefines the problem.

The LSID (Life Science Identifier)7,8 approach developed by I3C may help meet this challenge. LSID and its associated services models show how to provide federated identity and distributed resolution and services definition with much reuse of existing Internet technology. The approach is simple, lightweight, reusable and based largely on widely deployed current technology.

LSIDs are specialised URNs (Uniform Resource Names).9 Consider them as a sort of ‘far pointer’ across the Internet. They are designed principally to support programmatic interoperability via Web Services.10

The LSID syntax is:

urn: LSID:(AuthorityID):(NamespaceID):(ObjectID):(RevisionID)

Some examples are:

- urn:LSID:ebi.ac.uk:SWISS-PROT/accession:P34355:3
- urn:LSID:rcsb.org:PDB:1D4X:22

In LSID, each authority (ebi.ac.uk, rcsb.org, ncbi.nlm.nih.gov, etc.) is responsible for defining its own namespaces, object IDs and versions. Object IDs may have meaning locally but globally in terms of LSID and its services, they are opaque, semantically void.

Resolution works as follows:

- LSID client software resolves the authority for an LSID to an authority server.
- LSID client SOAP11 query to authority server finds the access method for LSID object.
- LSID authority server returns WSDL12 record defining methods available on LSID object.
- LSID client invokes desired methods remotely using web services.

LSIDs are location independent, persistent and unique. They are consistent with web
services standards such as XML, SOAP, WSDL, UDDI, OGSA and so forth. LSID is also compatible with a number of significant consortia-based efforts at bioinformatics infrastructure development, including BioMOBY, DAS, myGrid, OmniGene, MAGE-ML and GO; as well as the public bioinformatics databases in general, which are the simplest use case.

DAS and OmniGene are already committed to using LSID; OmniGene demonstrated web services integration using the Draft 1 LSID APIs at BIO2002; a version of IBM’s open source LSID resolver was prototyped at the Protein Data Bank in November 2002.

The LSID syntax definition is currently under review by the I3C Science and Technology Board and is expected to be issued as a finished recommendation in the first quarter of 2003. Two early implementations of the resolver code are now available under open source licence, one of which was installed as a prototype at the Protein Data Bank in November.

Research organisations working with genomic information have already made huge investments in infrastructure to enable mass computation to be performed across the relevant data, external and internal. A significant part of the investment has been to support methods to identify, access and interconnect the data. Current public methods are not efficient, maintainable or robust enough and each organisation invents its own infrastructure to fill in the gaps. The LSID services approach offers a promising way forward out of this dilemma.

Tim Clark, Editorial Board Member, Briefings in Bioinformatics
Millennium Pharmaceuticals, Inc., USA

References