

Creating a bioinformatics nation

A web-services model will allow biological data to be fully exploited.

Lincoln Stein

During the Middle Ages and early Renaissance, Italy was fragmented into dozens of rival city-states controlled by such legendary families as the Estes, Viscontis and Medicis. Though picturesque, this political fragmentation was ultimately damaging to science and commerce because of the lack of standardization in everything from weights and measures to the tax code to the currency to the very dialects people spoke. A fragmented and technologically weak society was vulnerable to conquest, and from the seventeenth to the nineteenth centuries Italy was dominated by invading powers.

The old city-states of Italy are an apt metaphor for bioinformatics today. The field is dominated by rival groups, each promoting its web sites, services and data formats. Unarguably, this environment of creative chaos has greatly enriched the field. But it has also created a significant hindrance to researchers wishing to exploit the wealth of genome data to its fullest.

Despite its shaky beginning, the nation of Italy was eventually forged through a combination of violent and diplomatic efforts. It is now a strong and stable component of a larger economic unit, the European Union, with which it shares a common currency, a common set of weights and measures, and a common set of rules for national and international commerce. My hope is that bioinformatics will one day achieve the same degree of strength and stability by adopting a universal code of conduct along the lines I propose here.

Screen scraping: mediaeval torture

The promise and peril of the bioinformatics landscape is clear to any bench biologist attempting to mine the human genome for information on, say, a favourite genetic region. The online sources of these data each provide remarkable user interfaces and deeply interconnected data sets of great richness. Yet each interface is different, both in the subset of data presented and in organization. The researcher may find herself devoting as much time adjusting to differences in presentation of the data as she does actually thinking about them. The situation is worse when comparing a human gene to its orthologue in another species. This brings the model organism databases into play, each of which has its own type of user interface and format. (See www.expsy.org/alinks.html; www.stat.wisc.edu/biosci/genome.html; and mbcf.dfc.harvard.edu/cmsmbr/biotools/biotools10.html for an idea of the scale of the problem.)

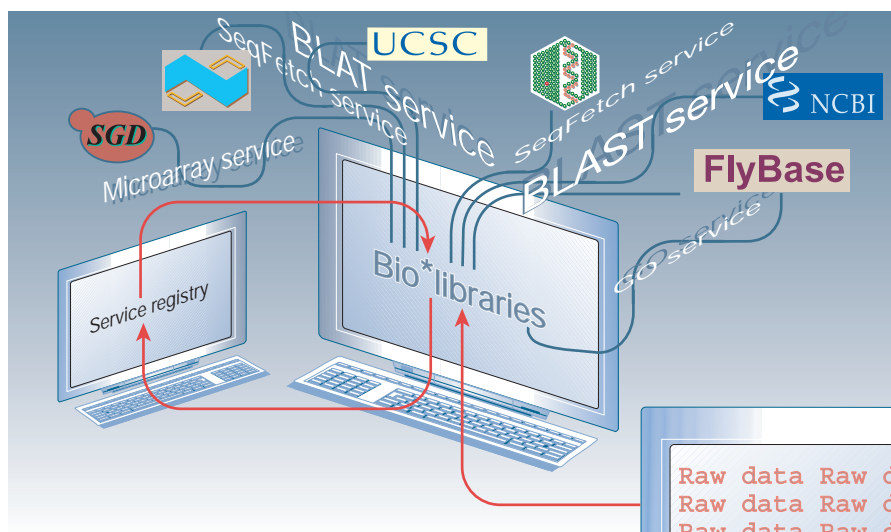


Figure 1 Moving towards a bioinformatics nation. Because each data provider (such as Flybase and UCSC) publishes data in an idiosyncratic form, the Bio* software package (Bio* libraries) was created to massage data into a standard internal format. Unfortunately, Bio* needs to be fixed each time a provider changes its formats. A web-services world would build on the successes of the Bio* projects by defining standard interfaces to various types of computations and data formats. The Bio* libraries can be written to recognize these interfaces, allowing them to interoperate easily with all data providers. A service registry would let data providers enter an electronic 'address book', allowing the Bio* libraries to locate and interact with new data sources automatically.

This inconvenience for the bench biologist is disastrous for the bioinformaticist, who typically needs to aggregate data from many online sources to create a data set for further analysis. When these data reside on different servers using different data formats and access methods, the first step is to write a set of software 'scripts' to fetch them, reformat them and place the extract into a local database. This is not straightforward, because most online biological databases were designed to be accessed by humans, not by machines. Bioinformaticists often find themselves writing scripts to parse the HTML source to extract the data while ignoring graphics links and explanatory text, a process called screen scraping.

Screen scraping is despised for various reasons. First and foremost, it is brittle. Database managers are always tinkering with the user interface, adding a graphic here, moving a button there, to improve the user experience. Each small change in a popular web page breaks dozens of screen-scraping scripts, causing anguished cries and hair-tearing among the bioinformaticists who depended on those scripts for their research and the research of the wet labs they support. Second, it is unreliable. There is no published documentation of what a data source's web pages are supposed to contain, so bioinformaticists must guess from a few

examples. Finally, there is massive duplication of effort. Almost every bioinformaticist has written a parser for the National Center for Biotechnology Information (NCBI) BLAST service at least once, sometimes many times. Because they are one-offs, these scripts are generally undocumented and not widely distributed. Most of them only work for a short time because BLAST changes every few months.

Bio* projects reduce the pain

The bioinformatics community has responded to the challenges posed by this 'city-state' situation with the Bio* projects, a series of freely available open-source projects (www.open-bio.org), in which nearly a hundred software engineers have developed re-usable code libraries in the Perl, Java, Python and Ruby programming languages (known as Bioperl, BioJava, Biopython and Bioruby, respectively). These libraries automate common bioinformatics tasks, such as manipulating DNA and protein sequences, and provide methods for importing and exporting data between data sources and among file formats. To fetch a piece of data from a database, the bioinformaticist uses the Bio* libraries to do the fetch, put the information in a standard format, and return the reformatted data to her script. This prevents duplication of effort. No one will ever again

be forced to write a BLAST parser.

But the Bio* libraries can't solve the problem of the brittleness of online data sources. As soon as one of these web pages changes, the Bio* library breaks and has to be patched up as quickly as possible. This works only because the Bio* libraries contain a series of adapter modules for each of the online databases, lovingly maintained by a group of dedicated (and very busy) programmers. The Bio* libraries also cannot immediately solve the problems of the data providers themselves. Whenever two providers need to exchange data, for example to share sequence annotation data, they must agree on an 'exchange of hostages' treaty in which they negotiate the terms and format of the exchange. Needless to say, this type of negotiation is awkward and time-consuming.

Strength through unity

To achieve seamless interoperability among online databases, data providers must change their ways. If they all had identical ways of representing biological data, standard user interfaces and standard methods for scripts to access the information, the problem of gathering and integrating information from diverse data sources would largely evaporate. But such conformity would destroy the creative aspect of online databases — and, indeed, no data provider would willingly surrender to it (but see Box 1 for a proposed code of conduct).

A more acceptable solution relies on the emerging technology of web services. A web service is a published interface to a type of data or computation. It uses commonly accepted data formats and access methods, and a directory service that allows scripts to find them. The web-services system shown in Fig. 1 allows several data sources to provide the same type of service so, for example, the NCBI, the University of California Santa Cruz (UCSC) and the European Bioinformatics Institute (EBI) could all provide a sequence-similarity service. A script written to use one service would work with them all, even though the three sites could implement the service in radically different ways: the NCBI might use BLAST, UCSC might use BLAT, and the EBI the SSAHA search engine.

For bench researchers, the web-services world might not look very different from the current one. Online databases would still exist, each with its own distinctive character and user interface. But bioinformaticists would now be able to troll the online databases to aggregate data simply and reliably. Furthermore, software engineers could create standard user interfaces that would work with any number of online data sources. This opens the door to genome 'portals' for those researchers who prefer to access multiple data sources from a single familiar environment.

A challenge for web services is to deter-

Box 1: Data provider's code of conduct

The downside of the web-services world is that it won't be achieved overnight. It will be some years before web services are stable and complete enough to replace current sites. Meanwhile, I propose the following bioinformatics data provider's code of conduct, to maximize the usefulness and re-usability of a data source.

1. A web page is an interface

Although web pages were designed for access by people, they can be accessed by scripts. Guide this trend, don't fight it. Online databases should provide the rules for linking to pages, including the hours and frequency available for scripts to download information.

2. An interface is a contract

Once a web page is in use by screen scrapers, it becomes a contract between the data provider and the data consumer. Changing the interface violates the contract. Document each interface and warn if it is unstable. If an interface needs to change, provide users with plenty of advance warning. When a widely used interface is changed, give it a new URL and maintain the legacy interface for a while.

3. Choice is good

Make your information available in several formats for bioinformaticists with different needs and abilities. HTML is the least desirable format to publish data for use with bioinformatics scripts. Tab-delimited text is easily handled by scripts and is suitable for many types of simple data. XML is harder to parse but can convey much more complex information. A SOAP/XML interface may not be accessible to novice bioinformaticists, but will be greatly appreciated by the more advanced ones.

4. Allow batch downloads

Make the whole data set available for batch download, breaking it up into logical, bite-sized pieces if necessary. Many developers have found themselves writing scripts to download entire databases one web page at a time as the only route to a full data set, which is wildly inefficient.

5. Use existing file formats

Avoid reinventing wheels: use existing file formats when possible. For example, there are already enough formats to describe features on a sequence (GenBank, EMBL, GFF, BSML, Agave, GAME, DAS), so it is doubtful that the world needs another. There are also good formats to describe genome assemblies, microarray experiments and results, three-dimensional structures and bibliographic references.

6. Design sensible formats

If an existing format doesn't support the information that a data source wants to publish, use common sense in designing new formats. Use tab-delimited text if it is sufficient. If the data are hierarchical, XML is a natural choice. It is better to start simple than to create a complex format to cover all contingencies.

7. Allow ad hoc queries

Support a true query language. Researchers often search for hidden relationships among biological data, but browsing through a set of hyperlinked pages may not be the best way. Many online databases have created web-based forms that generate summary reports based on simple filters. Better still would be to make copies of your databases available for direct access using the database's native query language. Although this is a significant investment, it is well worth the effort.

mine what data sources implement which services. The solution is a service registry. To advertise its services, a data source registers its services with a designated directory server on the Internet. Then, when a script needs access to a particular service, it consults the registry to find out what data sources provide the service. If the same service is provided by more than one data source, the script consults the user for help, or uses built-in rules to choose the data source, then returns the results.

Although this proposal may seem a far cry from what happens now, the technology exists to make it a reality. The World Wide Web Consortium, with industry heavyweights such as IBM and Microsoft, is promoting an alphabet soup of standards: SOAP/XML, WSDL, UDDI and XSDL. These are already being used by some data providers: examples include those sites that publish genomic annotation data using the distributed annotation system format (www.biodas.org) and the EBI's SOAP-based bibliographic query service (industry.ebi.ac.uk/openBQS). Also being developed is a highly promising integration platform called Omni-

gene (omnigene.sourceforge.net); Integr8, an ambitious integration project run by the EBI (www.ebi.ac.uk); and a pharmaceutical industry-backed project called I3C (www.i3c.org). Finally, the caBIO project (ncicb.nci.nih.gov/NCICB/core/caBIO) at the US National Cancer Institute (NCI) is a comprehensive open-source project that aims to make the NCI's cancer databases available via a web services architecture.

The risk, of course, is that like the mediaeval Italian city-states, each of these projects will endorse its own idea of standardization, and a chaotic world of incompatible bioinformatics data standards will be replaced by a chaotic world of incompatible web-service standards. We can look forward to a bit of a struggle before one set of standards achieves pre-eminence, but I have no doubt that unity will be reached eventually. ■

Lincoln Stein is at the Cold Spring Harbor Laboratory, Cold Spring Harbor, New York 11724, USA. This Commentary is adapted from a keynote speech given by the author at the 2002 O'Reilly Open Bioinformatics Conference in Tucson, Arizona, USA.