

Editorial

OPEN BIOINFORMATICS

Many wonderful algorithms are published each month in *Bioinformatics* and other computational biology journals. Most of these algorithms are made freely available to the rest of the bioinformatics community. However, there is a growing trend to make new software available only through a HTML interface on the World-Wide Web, or via a closed, proprietary binary distribution. This trend will ultimately inhibit the growth of our discipline. As a community, we need to dedicate ourselves to the free and open exchange of algorithms and software.

Our journals recognize the need for freely exchanged ideas: *Bioinformatics* requires that software be available for two years. However, the mechanism of making programs available to the rest of the community is not specified. The choice is rightly left up to the researcher. But some choices are better than others, and it is important to be sure that we make the choice that will best advance our community as well as our careers.

The most forthright approach to sharing software is through the use of Open Source. The Open Source initiative makes the original source code available to other developers via a license that allows for free redistribution and derived works while preserving the integrity of the author's original source code (see <http://www.opensource.org>). The largest repository of Open Source software is the SourceForge (<http://sourceforge.net>), which hosts nearly 200 bioinformatics software development projects, including GeneX (Mangalam *et al.*, 2001; <http://genex.sourceforge.net>), GO (The Gene Ontology Consortium, 2000; <http://www.geneontology.org>), and JMOL (<http://jmol.sourceforge.net>).

Open Source is not the only way to make source code available to other researchers. Many projects use the GNU artistic license to distribute their software. These project range from code libraries such as Bioperl (<http://bioperl.org>) to relatively complete sequence analysis packages like EMBOSS (Rice *et al.*, 2000). Other packages like Phred/Phrap and Consed (Ewing and Green, 2001; Ewing *et al.*, 1998; Gordon *et al.*, 1998) use license agreements peculiar to their own organization. But the end result is the same, placing our source code in the hands of other researchers.

While such free and open source distribution is the ideal, such distribution is often constrained by our parent organizations. As universities become more aware of the potential value of our IP, the full distribution of our code is often blocked by administrators who have visions of license agreements making millions for the university general fund. Thus we are often forced to come up with

a compromise between free distribution and secrecy.

WWW interfaces are by far the most common means of compromising between the need for IP protection and making an algorithm freely available. In the sequence analysis world, the Blast interface by NCBI is the most widely used template for user interface. However, unlike Blast, most programs behind the WWW interface are not freely available. For example, when we went looking for a program to predict transcription factor binding sites, we found many sites that would allow us to submit a sequence, but none that allowed us access to the original program. Similarly, there are sites that allow a user to search for SNPs, identify genes, and perform other types of sequence analysis via a WWW interface, but only with a single sequence at a time.

Unfortunately, a scaling problem often arises from hiding implementations behind WWW interfaces. While analysis via single sequence submission is fine for most biology labs focused upon a few specific genes, in the genome era there is also a need for high-throughput sequence analysis to go with our high throughput experimental protocols. For example, microarray gene expression analysis often identifies hundreds of interesting genes. Annotating those genes using a single sequence submission interface would be time consuming and frustrating to say the least.

Evolution of WWW protocols has given us a much better approach. Software applications that are not being distributed as source code can be implemented as a set of web services. Web services are programmatic methods of accessing a remote program or database, and serve as application interfaces for remote applications. The World Wide Web Consortium (W3C, <http://www.w3.org>) is the clearinghouse for technologies using WWW protocols to transmit and exchange data.

The primary technology standards for creating web services are XML and SOAP. XML is the acronym for eXtensible Markup Language, and in structure XML is a superset of HTML. The XML message is created according to the Shared Object Access Protocol (SOAP) which defines a framework for describing what is in the message and what datatypes and what requests and responses are supported by the server creating the XML message. Creation of XML and SOAP responses are nearly as straight-forward as creating HTML pages.

Several new bioinformatics projects are using web services to provide access to data. A prime example of how XML can be used to create web services is DAS, the Distributed Annotation System (Dowell *et al.*, 2001; <http://www.biodas.org>). DAS allows database providers to publish their data in a form that can be accessed via remote clients and programs. Dozens of DAS servers already exist, and several sophisticated clients like the

Ensembl DAS client (<http://www.ensembl.org/das>) allow easy access.

A more ambitious example of how web services can be used is the MOBY project (<http://www.biomoby.org>). Still in the early stages of development, the MOBY system consists of a central repository which tracks available services, and remote servers which provide those services. While the details are somewhat more complex than the DAS model, the bottom line is that like a DAS client, a MOBY client can access data and programs from remote servers.

Properly configured, web services will allow tools and databases to be accessed as if they were local, and users are assured of access to the latest versions without the concerns of performing local updates. Software clients of web services can be more complex and sophisticated than HTML-based interfaces, and results from algorithms made available via web services can be easily integrated into larger programs. Just as the WWW is a distributed information system, bioinformatics web services could be tied together into a distributed analysis environment.

Of course, web services are not the only mechanism for making programs and databases available for remote use. Before the ascendancy of the WWW there were many programs like Network Entrez (Rioux *et al.*, 1994) that connected to databases and programs using low-level network protocols. Indeed, many of the W3C protocols are layered upon the low-level protocols. The advantage of using web services is that the higher level protocols take less work to implement. And, since the technologies are similar, providing web services to our programs and databases will require an incremental increase in work over simply putting up a HTML-forms based interface.

As scientists, our ideas are our main currency for earning fame, fortune, or tenure, leading to an understandable desire to protect our valuable intellectual property.

But software security measures which don't allow for examination of original code or for reasonable mechanisms of validity testing are counter to the open communication needed to do science properly. Providing access to our software via web services is a reasonable compromise between secrecy and openness, and could have the added benefit of speeding bioinformatics systems design and creating greater synergy within the community.

REFERENCES

- Dowell,R.D., Jokerst,R.M., Day,A., Eddy,S.R. and Stein,L. (2001) The Distributed Annotation System. *BMC Bioinformatics*, **2**, 7.
- Ewing,B. and Green,P. (1998) Base-calling of automated sequencer traces using phred. II. Error probabilities. *Genome Res.*, **8**, 186–194.
- Ewing,B., Hillier,L., Wendl,M.C. and Green,P. (1998) Base-calling of automated sequencer traces using phred. I. Accuracy assessment. *Genome Res.*, **8**, 175–185.
- Gordon,D., Abajian,C. and Green,P. (1998) Consed: a graphical tool for sequence finishing. *Genome Res.*, **8**, 195–202.
- Mangalam,H., Stewart,J., Zhou,J., Schlauch,K., Waugh,M., Chen,G., Farmer,A.D., Colello,G. and Weller,J.W. (2001) GeneX: An Open Source gene expression database and integrated tool set. *IBM Systems Journal*, **40**, 552.
- Rice,P., Longden,I. and Bleasby,A. (2000) EMBOS: The European Molecular Biology Open Software Suite. *Trends Genet.*, **16**, 276.
- Rioux,P.A., Gilbert,W.A. and Littlejohn,T.G. (1994) A portable search engine and browser for the Entrez database. *J. Comp. Biol.*, **1**, 293.
- The Gene Ontology Consortium, (2000) Gene Ontology: tool for the unification of biology. *Nat. Genet.*, **25**, 25.

D. Curtis Jamison
School of Computational Sciences
George Mason University Manassas
VA 20110 703-993-8426
E-mail: cjamison@gmu.edu