BIOINFORMATICS

Working the Web With a Virtual Lab and Some Java

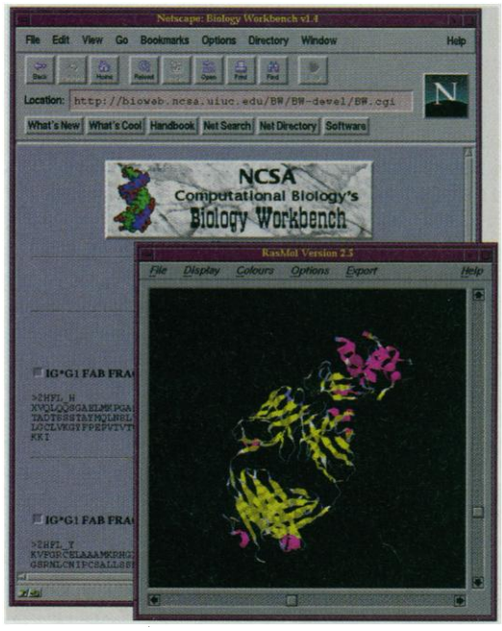
Curt Jamison knows the drill all too well. Say you have a new piece of DNA to analyze, says the University of Maryland geneticist. You're looking for sequences similar to those in known DNA and clues to the function of their proteins. There are plenty of databases and analytical software out there, but until this year, anyone trying to use these biocomputing tools faced what Jamison calls "a kind of Tower of Babel. There are a lot of tools available, but they all speak a different language." Jamison recalls having to translate sequence data from MS-DOS text to a different format to run BLAST, a sequence comparison program, on a Sun workstation. Then he had to convert all the sequences back into straight text to send them over the Internet to another computer running DNAstar, a program that aligns matching sequences to show similarities and differences. There had to be a better way, he thought: Biocomputing needed a common language, or at least some common ground.

Now Jamison and other biologists are finding plenty of both, from two separate developments. The common ground takes the form of the Biology Workbench, unveiled in June by the National Center for Supercomputing Applications (NCSA) at the University of Illinois. At one site on the World Wide Web, biologists can find a "gateway" that provides one point of access to a far-flung collection of protein, DNA, and bibliographic databases and tools for searching and swapping data and analyzing sequences. The Workbench itself runs on NCSA supercomputers, but any biologist with access to the Web can use them. "It's point-and-click biology," says Shankar Subramanian, who led the NCSA team (including Jamison) that built the bench.

The common language is Java. Java is a programming language that allows a user to retrieve little programs called "applets" from remote sites on the Web and run them on a local machine without worrying about software compatibility. While the Biology Workbench brings data and tools together at a common site, Java brings both back to the user, where biologists can create sophisticated displays from data retrieved from the source—homing in, for instance, on specific data subsets—without spending time going back and forth at each step.

Just what that means to scientists became clear in June, at the Meeting on the Interconnection of Molecular Biology Databases in St. Louis, where researchers saw prototypes of Java genome browsers that allowed them to literally zoom in on the fly, going from a section of a *Drosophila* chromosome down to the DNA sequences, or browse along comparative linkage maps that stretched a grass gene alongside a corn gene. Some of these browsers are versatile enough that researchers are adapting them to work on the human genome, or on yeast. "Java has come on the scene like gangbusters, and it blew away everything else," says bioinformatics specialist Stanley Letovsky, who works on the Genome Data Base (GDB) at Johns Hopkins University.

These feats haven't toppled the Tower of Babel completely. The Workbench doesn't have Java's interactive finesse, and Java itself has some problems. Some observers caution that its graphics capabilities are limited, and Java is the focus of both security concerns (see box on next page) and complaints that existing security features limit its usefulness, actually preventing researchers from saving their work on their own machines. Moreover, it's not a universal language, for applets can't go anywhere on the Web for data; they can only access the servers where they originated, which limits the kind of analysis they can do. Still, says Harvard University's William Gelbart, a principal investigator of the Flybase *Drosophila* genetics project, "we view Java as the wave of the future. It's democratic," because it lets anyone with a Java-equipped Web browser tap into the latest software. Or as Jamison puts it: "Power to the people."



Pop-up proteins. NCSA's Biology Workbench lets users search many sequence databases and see protein structures.

Putting the Web to work. Jamison (then at NCSA), Subramanian, and their colleagues already had that slogan in mind 2 years ago, even before Java, when they laid the groundwork for the Biology Workbench. They realized that by combining the universal access provided by the Web with something called dynamic federation, they could put a whole arsenal of biocomputing power into the hands of any Web-equipped biologist. NCSA's Bruce Schatz explains the concept: "One [Web site] takes your query—give me all proteins with this particular sequence, for instance—and sends it to 30 different sources in their own formats," he says. "It gets the data back, translates it into your language, and gives you the information. So you can send queries to many databases across the Web and change the results back into Web pages. It's completely transparent."

Subramanian, Jamison, and their NCSA group were already at work on such a federation of databases, called ENQuire, that searched both large and small genome databases. They realized that they could make their gateway still more useful by adding tools for analyzing sequences for structure and function, which would run on NCSA supercomputers. The result, in June, was the Biology Workbench.

In a typical search, says Subramanian, a user might type the name of the digestive protein "trypsin" in the Workbench query box, then select three databases (PIR, GenBank, and PDB) to search. A set of similar protein sequences identified by other researchers soon appears, and the user clicks the "import" button to bring them to the Workbench. Clicking the "Multiple Sequence Alignment" button aligns them. Another tool called MSAShade highlights all the matches and mismatches, showing which sequence is closest to the original. A protein structure prediction tool will even pop up a three-dimensional image of the molecule, which gives clues to its function. "So by running a machine with just a Web browser, a user has access to programs that pack all this power," Jamison says. The Workbench now provides access to about 30 databases and 100 analysis tools.

He's quick to add, however, that while everyone can get to the Workbench, it's not a spectacularly fast or flexible working environment. The interface is written in the Web's vernacular, hypertext markup lan-

Do Java Users Live Dangerously?

Java is giving researchers a dose of excitement by allowing them to visualize or analyze data with software commandeered from distant machines (see main text). It is also fueling visions of a bustling Internet economy. This operating system, which makes it possible to download foreign programs over the Internet and run them in a local computer, will allow users to invite electronic shopkeepers into their home machines to conduct transactions. But these very attributes are also giving security experts the jitters.

The ability to download and run foreign programs—applets, in Java-speak—could in theory expose a host machine to computer viruses and other digital mischief. Java's designers were well aware of this vulnerability, so they included an array of software guardians that screen each applet for admission and then keep it in what Java engineers call a sandbox, where it can't run amok. But several computer experts have learned to use bugs in Java to bypass its safeguards. Each new security breach has made the vision of a bustling Internet economy seem more distant and even raised concerns about other uses of Java. They have also rattled staff at JavaSoft. "I've named gray hairs on my head after [the attacks]," says Marianne Mueller, a staff engineer at the company.

Finding fault. Oxford's David Hopwood.

The latest Java bug was found by David Hopwood, a student at Oxford University. On 1 June, he announced on the Web that he had created an applet capable of undermining Java's security safeguards. "He found a bug—a subtle bug," agrees Mueller. Hopwood's subtle bug exposes the "security manager" at Java's front door to attack. The security manager acts as a gatekeeper. Whenever an applet tries to do something potentially mischievous, like renaming files or creating new directories on the hard drive, the security manager slams the door. But Hopwood wrote a program that, when downloaded by a Java host, kills the security manager and replaces it with an impostor—a phony who dozes at the gate, leaving it open for later invaders.

To make matters worse, three Princeton University researchers had already found a bug that enables a rogue applet to escape from the sandbox. "If you combine the two attacks," says Hopwood, "you can run any code"—even code that tells your computer to send bank records to Taiwan or to erase your hard drive. "It's a vicious attack," agrees Drew Dean, a member of the Princeton team.

"I'm feeling kind of bloodied," admits Mueller. JavaSoft and Netscape, which developed the World Wide Web browser on which Java runs, are both working to patch the holes. But others keep cropping up, which is in part a reflection of Java's power. "There's a tension between being secure and doing interesting things," says Mueller. "Often, we're between a rock and a hard place." Security experts also blame a hurried production schedule. "Overall, companies are racing too rapidly to add new features" to software including Java, says Edward Felten, head of the Java research effort at Princeton. "And new code means new bugs." But he notes that JavaSoft is working hard to identify any new vulnerabilities—even going as far as funding the Princeton group's efforts.

So far, the attacks seem to be limited to laboratory exercises. Although rumors of Java viruses are rife, Hopwood calls them "hype. Sensationalist hype." Even so, Hopwood, Dean, and Felten all disable Java and Javascript on their browsers when wandering through the Internet.

Although only the computer experts are nervous now, consumers might have reason to worry in the future. If Java becomes the backbone of a new virtual marketplace, applets will interact with shoppers on their home computers. This means that Java applets will have to be able to accept money—and hostile applets could easily eavesdrop on transactions and skim off some of the proceeds. In that case, "[a Java-based attack] would be a good way to steal money a little at a time from a lot of people," says Dean. "It's all fun and games until there's real money involved."

–Charles Seife

Charles Seife is a free-lance science writer in Scarsdale, New York.

guage, or HTML, and as languages go it's pretty static. "HTML basically works with forms, like documents. There are whole classes of things that you can't do with it," Jamison says. "You can highlight a chunk of sequence, for instance, but to do something with it, you have to fill in its parameters on a form. Then the server does the processing and ships it back to you." To focus on a smaller chunk of the sequence, you have to fill in more parameters and resubmit. "HTML is back and forth, back and forth, every time you want to add a term. Or maybe you want to rotate a protein view or loosen a folding bond. HTML can't do that for you."

Some genes with your Java? But the Java language can, to a certain extent. "What Java does is move the code over to the client," says Gregg Helt, a biologist in the University of California, Berkeley's, *Drosophila* Genome Project and author of the Java-based *Drosophila* Genome Browser prototype. "And that makes things more dynamic." Java applets are embedded in HTML pages, the standard form for displaying information on the Web, but they run on the user's machine and pull data off the server they came from to reshape it at the user's command. The user doesn't need to go back to the original server for each operation. So, says Helt, you get the advantages of the Web's universal access and Java's facile visualization and interaction.

Actually, Helt says, "I was opposed to Java at first" because its graphics seemed limited and he worried about bugs. "I set out to prove it didn't work. What I found was that, hey, it works pretty well." It was good enough, in fact, for him to put the genome browser on the Web earlier this year. It shows a 3-megabase region of the *Drosophila* genome as a physical map, with chromosome bands. Users can zoom in on a band to see the subbands and still finer landmarks known as contigs and P1s. "You can't do that with HTML," notes Helt. "It would have to go to the server and get another picture file."

Then, says Helt, you can unpack and analyze the information hidden in each of those features. "Click on the P1, and you get a window with an annotated map of that sequence: BLAST homologies, gene predictions, and known GenBank entries. You can get down to the DNA level, and for a P1 that's 80,000 base pairs. You get a DNA viewer in a pop-up window, and you can select features in the annotated map and the viewer will move to them." And once users have selected a gene, another window will let them see what cells in the early fruit fly embryo express it.

Helt has also included a primer-prediction tool for scientists who want to amplify a particular segment with the polymerase chain reaction. A window lets a user specify primer length, temperature, and other options; the tool then recommends primer sequences, and arrows appear on the map to show where they

would bind. "That's a way to find the best primers to pull out a particular region," Helt says.

Researchers who have browsed the browser are impressed. "One of the biggest wins is the platform independence" that comes with the Web and Java, says Nomi Harris of Lawrence Berkeley National Laboratory's Human Genome Informatics Group. "I also like the hierarchical way the display works, so you can see many levels of detail." Harris, in fact, is working to adapt the fly map browser to work on human genome data. Over at Stanford University, Michael Cherry, who works on yeast and *Arabidopsis* genome sequencing projects, is also working on an adaptation. "Greg has done an excellent job," he says.

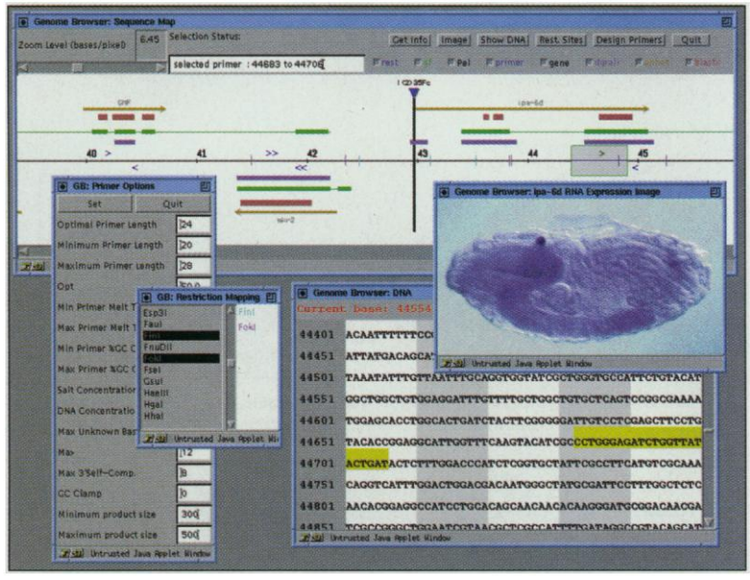
And Gelbart notes that the applet, running on the Web, solves a long-standing Flybase problem: keeping the various maps up to date. "The important thing in these projects is to get information out in a timely manner. Until now, if we wanted high-quality dynamic maps, the best we could do was the Encyclopedia of *Drosophila*, a CD-ROM that has a lot of this material. We'd come out with a new one every 6 months or so. But with Java on the Web, as we update maps and graphics, you update."

A brew of applets. But Java isn't limited to fly genetics. Applets are popping up like Starbucks coffee houses. In St. Louis, Jamison showed off a comparative map viewer for plant genomes. "We want to make information available about agriculturally important organisms," he says. The viewer takes data about plant genomes stored in a database at the U.S. National Agricultural Library and aligns linkage maps, which lets a user scrolling down the maps find homologies across species. Researchers at Stanford are working on SStruct, an applet that allows a user to choose an RNA sequence and see its secondary structure, which influences how it interacts with other molecules in the cell. GDB, at Johns Hopkins, which contains location information for markers linked to genetic diseases, is working on a "multimap" that will let users view the same region drawn from several different data sources, revealing gaps in marker coverage. ACEDB, a database first developed for the *Caenorhabditis elegans* sequencing project, is getting a Java front-end called Jade. And Subramanian says he plans to come out with a Java-ized Workbench.

Nor are all applets linked to specific databases. The University of Pennsylvania's Computational Biology and Informatics Laboratory has been trying to develop a library of adaptable, reusable software modules for standard biocomputing tasks—a.k.a. bioWidgets. "One big problem in the genome project is that we keep reinventing the wheel," says Penn's Christian Overton. "Groups that write software don't shrink-wrap their programs and provide support for other groups." Widgets, an idea that came from Penn's David Searls, could eliminate the wasted effort, says Overton. "They are small, clean graphical modules like a map viewer or a sequence viewer. ... Widgets have been implemented in several languages now. But the problem was: How to get them across the Internet?"

The solution seems to be Java, Overton says. With collaborators from Berkeley and several other places, the Penn group is starting a widget consortium, to create them and make them available as applets. "The response has just been amazing. Everybody wants in," Overton says. Jamison explains why: "That will save me a heck of a lot of work."

That savings may not come right away. Helt, Jamison, and their colleagues admit that unlike other computer languages such as Perl, Java doesn't come with many drawing routines that are useful for scientific graphics, so the programmers have to build the routines from scratch. And that takes time. A bigger problem is that for security reasons Web browsers don't let Java applets store data on the local computer, which means that all the vaunted Java interactivity goes to waste at the end of a session. Although you can zoom in on a region on a linkage map and jot some markers down in a notebook, "this is a fairly big hole," Jamison says. "A lot of what we do, we'd like to save." (Netscape and Sun Microsystems—who gave the world Java—have made noises about changing this soon.)



Java on the fly. This Java "applet" lets users go from a *Drosophila* physical map to DNA sequences to patterns of gene expression.

True transparency. Some researchers add that Java's fortes—visualizing data and some limited analysis—are not enough to turn the world of biology upside down. "I think everyone should look at Java seriously," says David Lipman, director of the National Center for Biotechnology Information at the U.S. National Institutes of Health. But he notes that applets may not give biology the same kind of boost that the field would get from, say, a comprehensive, searchable library of protein motifs—a tool for detecting distant evolutionary relationships that does not presently exist.

What could truly speed things along, just about everyone agrees, is true transparency: If every database, every analysis tool, on every platform, could interact, then the Web would be any browser's oyster. "There are literally hundreds of biological databases out there. You're not ever going to get them all in one place" or write applets that can interact with them all, says Overton. The trick will be to get them to interact with one another. The tool for that task, many feel, will be something called CORBA.

It stands for Common Object Request Broker Architecture, a moniker that means, in essence, that every database everywhere on the Web will have the same wrapping on the outside. "CORBA is a standard for packaging remote objects," says Tom Flores of the European Bioinformatics Institute (EBI) in Cambridge, U.K. "It's a kind of 'middleware' between the databases and the clients. With CORBA, I don't need to know anything about a particular implementation. You could change from flat files to a relational database, and I don't need to know. The program gives me a handle on the object."

EBI plans to put CORBA handles on several objects in the coming year. Flores says they have just won preliminary approval from the European Union for a grant to place CORBA wrapping around several large databases such as EMBL, SWISS-PROT, PIR, and GDB, and several new and specialized ones, such as P53 and TRANSFAC. And then, says Flores, "we'd really have some power. Think of the serious research you could do while sitting at home."

–Joshua Fischman