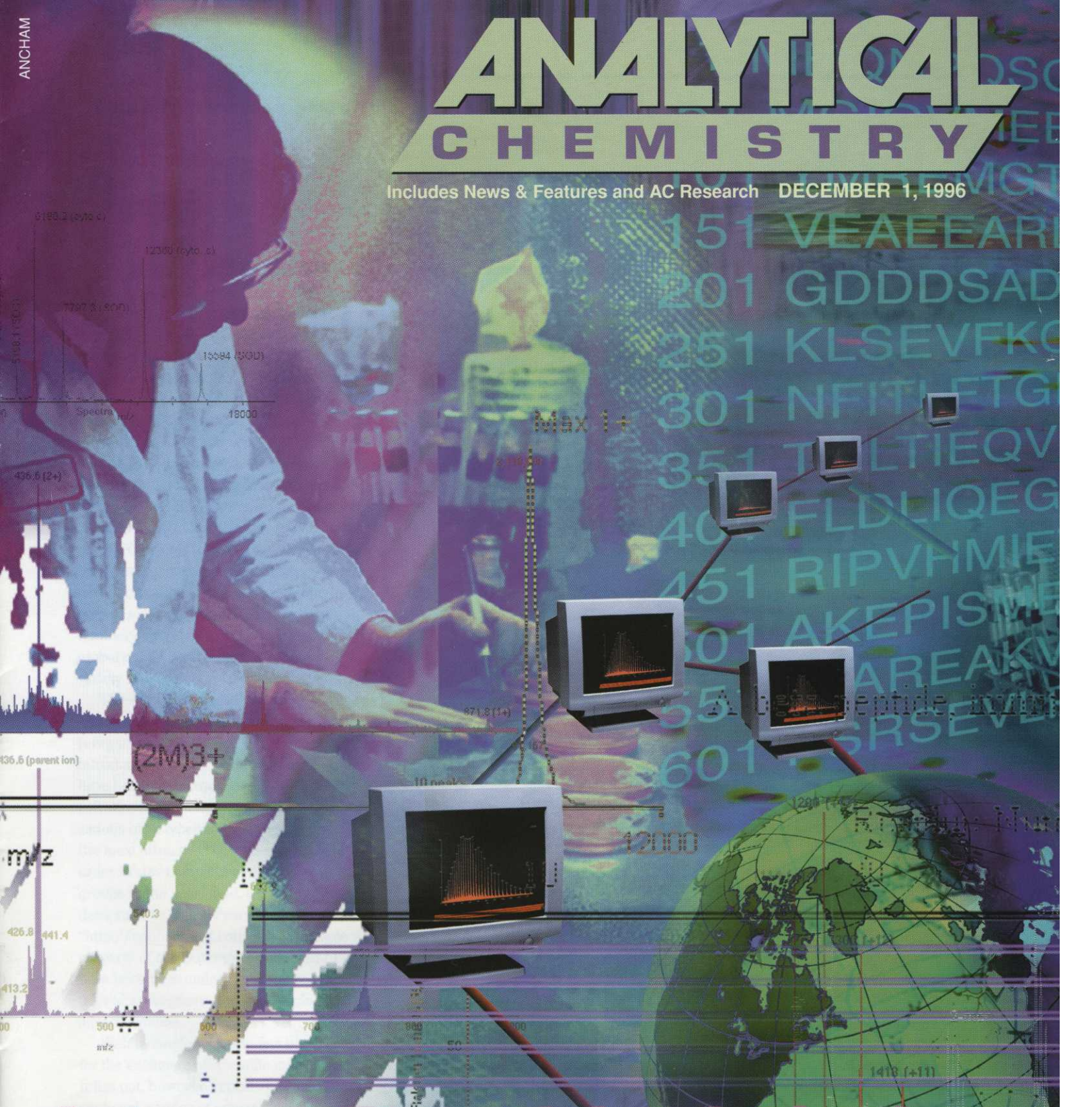
ANCHAM

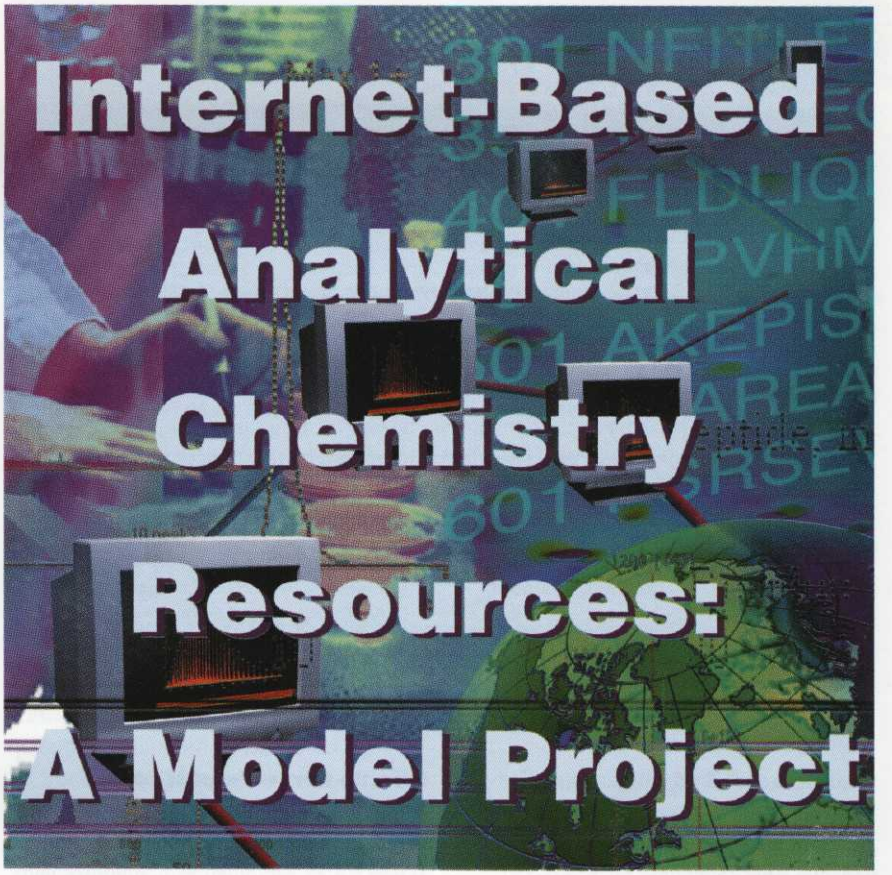
ANALYTICAL CHEMISTRY

Includes News & Features and AC Research DECEMBER 1, 1996



Internet-Based Analytical Chemistry Resources: A Model Project

Internet-Based Analytical Chemistry Resources: A Model Project



Web technology can be used to manipulate analytical data and facilitate the exchange of scientific information

Man has always sought to organize knowledge in his attempts to comprehend the complex world. We have seen many times during history how technological improvements prompted radical changes in the way information has been disseminated. One of the most dramatic examples was Gutenberg's invention some 500 hundred years ago that allowed the mass production of printed books. Today, with the popularity of the Internet, we may be seeing the beginning of a revolution equally influential. New tools have become available that promise to completely change the way information is disseminated. The Internet is already being used for a large number of projects involving, for example, global public libraries and teaching in a variety of fields (*1*).

The storage and dissemination of scientific information are also being radically reorganized. Within five years of their introduction, World Wide Web browsers have become the most commonly used software on personal computers. Organizations of all types are rushing to create the most attractive Web sites possible in order to stay competitive with other groups in the same field. Cryptic incantations such as "http://www.acs.org" or "http://nationaldebt.com", once the sole property of computer geeks and wizards, have become prominently displayed on television programs and in magazine advertisements.

Web technology was in fact invented for the exchange of scientific information. It has not, however, replaced more conventional sources of information such as journal articles and books any more than television has replaced radio. (Count how many radios you own compared with the number of televisions.) Computer screens are difficult to watch for long periods, and the physical requirements of using a computer make it incompatible with the way most people read long articles.

Web browsing has, however, become the first method of choice for many scientists and students when they want to find information on a subject. Currently, the quality of the information on the Internet is somewhat lower than that obtainable from conventional print resources, but the immediacy of discovering new information seems to outweigh issues of editorial and artistic nicety. Even so, the success of a global scientific "library" depends on the development of high-quality databases for the relevant scientific data, as well as on the development of means for convenient interconnection of these databases.

For all practical purposes, some types of scientific information exist only as Internet-based resources. Bioanalytical databases containing DNA and protein sequences and three-dimensional protein structures have become invaluable to molecular biology researchers and are normally accessed through Web interfaces (*2–7*). These databases are based on information from one of the largest analytical chemistry projects in history: the determination of the complete sequence of

David Fenyö
Wenzhu Zhang
Brian T. Chait
Rockefeller University
Ronald C. Beavis
New York University Medical Center

0003-2700/96/0368-721A/$12.00/0

Databases, CGI tools, and helper applications used in PROWL.

Databases	
MassBank	A collection of protein mass spectra
MatrixDepot	Data about known MALDI matrices
Protocols	Recipes and advice for carrying out protein chemistry experiments in a manner compatible with MS
Amino acid information	A collection of tables, figures, and models for understanding amino acid chemistry in proteins
Sequences	Up-to-date copies of SWISS-PROT, GENPEPT (the translation of GENBANK), PIR (Protein Identification Resource), and OWL (a nonredundant protein sequence database)
CGI tools	
ProteinInfo	Retrieves protein sequences on the basis of several query mechanisms, using a phylogenically organized set of sequence databases
ProFound	Searches known protein sequences for a pattern of masses obtained from protease digest experiments for protein identification
PepFrag	Searches known protein sequences for a pattern of MS/MS fragments from protease digest-derived peptides for protein identification
Display	An interactive tool for displaying mass spectra on an HTML page
Helper applications	
PAWS	For planning and analyzing the results of protein chemistry experiments on the basis of a proposed primary sequence that can be supplied in several standard formats
M/Z	Displays and analyzes mass spectra in a number of different formats, including the highly compressed format used to store raw data in MassBank

For the names and URLs of other helper applications that are suggested for use with PROWL, see the PROWL Software Page at "http://128.122.10.5/software/contents.htm".

genomes from selected organisms. The Web interface for these databases is simple: A user queries the database for information, and a file containing the requested data is downloaded by the user's computer.

In the process of constructing a bioinformatic database of protein mass spectra (MassBank), it became clear that the value of a Web-based resource can be greatly enhanced if it supplies a well-chosen set of software tools and basic information along with raw data. These additional features make the database an integrated resource for data retrieval and analysis rather than a simple data repository. Therefore, we created a more versatile resource called PROWL (http://chait-sgi.rockefeller.edu or http://mcphar04. med.nyu.edu). This resource incorporates a number of ideas about Internet resource design that can serve as an example of how current Web technology can be used for manipulating data derived from analytical chemistry. In order to explain the ideas underlying the design of PROWL, it is necessary to explain some of the basic technological concepts behind Web sites.

Client–server architecture

All Web-based transactions use a "client–server" model of computer interaction. When a user sits down at a computer and runs browser software, that computer becomes a "client". When the client browser wants a piece of information, it sends out over a network a request that details what the client wants. The request is in a highly structured, standardized form (called a "protocol") that depends on the type of information required and carries with it the address of the computer that has the required data (the "server") and some type of pointer that says where that information can be found on the server. The combination of protocol, server address, and pointer to the data is referred to as the universal resource locator (URL) of the information.

Protocols currently supported by browsers include the hypertext transfer protocol (HTTP), the file transfer protocol (FTP), and the sendmail transfer protocol (SMTP). A universal resource locator starts with the abbreviation for the protocol to be used, followed by a colon and two separator marks; the URL for a piece of hypertext has the general format "http://...". The name of the protocol is very important because it tells the server what sort of software to run to retrieve the required piece of information and what format to use when sending it back to the client computer.

The next element in a URL is the Internet address of the computer that will act as the server, in either words or numbers that are separated by periods. The address of the NYU server for PROWL can be written as either "mcphar04.med.nyu.edu" or "128.122.10.5". The address in words doesn't actually contain the information necessary to find a computer directly; it is used to find the numerical address in lookup tables that are scattered around the network in computers called domain name servers (DNS). Once the numerical address has been determined, it is then used to send the request to the appropriate server.

The use of numerical addresses for computers attached to a network in the form of four 8-bit numbers is a fundamental part of the Internet protocol (IP), which is the current standard for the interconnection of computer networks. Entering the numerical address in a URL will usually result in a quicker response from a server than using a word address because it does not require the initial step of consulting a set of lookup tables to translate the address.

The simplest (and most common) response that a server can make when receiving a query is to send back a requested file to the Internet address of the client computer, using the specified protocol. The server includes an additional piece of information along with the file, specifying what type of file it is sending. This additional information is called a multipurpose Internet mail extension (MIME) specification. The browser running on the client computer then interprets data sent from the server in a fashion that is appropriate for that particular MIME specification. For example, a browser will interpret text (MIME type "text/plain") differently from a picture (MIME type "image/gif") or a sound (MIME type "audio/x-wav").

This simple set of behaviors—sending a URL and receiving information with a

MIME type—has led to the success of the Internet style of information exchange, as opposed to older models. A client does not have to "log on" to a remote computer and run software explicitly; the standard protocols perform these operations in the background without user intervention. The server can send back a complicated mixture of different types of data (pictures, text, videos, etc.) and have it interpreted correctly. The server and the client do not need to be running the same type of operating system in order to interact, because neither computer attempts to directly control the other. Instead, they interact with each other in a rather abstract manner using formal expressions that do not refer to the manner of performing an action but only request that the action be carried out.

More complex client–server behavior

During the designing of MassBank, it became clear that the simple set of client–server behaviors outlined above is not flexible enough to produce a really useful resource. Protein mass spectra are usually quite complicated, so it was decided that rather than storing a simple table of masses, it would be necessary to store a representation of the original mass spectrum. The interpretation of these protein mass spectra can only be made within the context of the proposed covalent structure of the protein (i.e., the protein's amino acid sequence). In turn, the amino acid sequence can only be interpreted with reference to the properties of the individual amino acid residues, known or suspected post-translational modifications, and the methods used to prepare a sample for analysis.

Storing, viewing, and interpreting complex mass spectra and protein sequences require software tools that can examine and compare data from several sources. The resource designer's aim should be to condense all of the available information to a format that can be readily grasped by a user who is interested in a particular protein, while retaining the possibility of accessing detailed information for in-depth examination. The best format for the reduced information is a set of standard diagrams that can be composed "on-the-fly" by the client–server combination. The client–server architecture should allow the user to interact with these diagrams, so that features of interest to the user can be highlighted and explored by a sequence of point-and-click operations.

A range of technologies have been developed to make client–server interactions more versatile. The first and most commonly used of these techniques, called the common gateway interface (CGI), provides an interface to the server computer. The CGI allows a client to request a server to run a specified program that must be located on that server. The program is used to generate a stream of information directly back to the client by using the hypertext transfer protocol. The browser allows the client to send a set of parameters to the CGI, which will be supplied to the requested program and used to determine what output will come from the program. Therefore, the server can respond to a client by composing a piece of hypertext that did not originally exist but was composed to resolve the question asked by the client. Almost all WWW-based databases use CGI programs to do searches on the database and return the entries that correspond with the search parameters entered by the user.

Another well-developed method involves the server requesting that a client computer run a specific program to read the data that it is transmitting. The server accomplishes this task by specifying a MIME type for the data, in response to which the browser starts up another program on the client computer that interprets the information from the server. For the browser to start the correct program, the browser must be configured so that when it receives a particular MIME type (such as "image/gif"), it starts up a program capable of displaying GIF-type graphics files. The program that gets started on the client computer is called a "helper application". The browser actually receives all of the information from the server and stores it in a temporary file; the helper application is given the name of the temporary file when it starts, which gives it access to the information without it having to deal with receiving network information. Therefore, a helper application can be any software on the client machine that accepts a filename as a parameter when it is started. A somewhat more sophisticated type of helper application called a "plug-in" can be used to display graphics within a browser's window. Plug-ins use the same idea as helper applications: A program that already exists on the client computer is activated with a downloaded temporary file as one of its start-up parameters.

The most recently developed form of enhanced client–server interaction involves programs that exist on the server and are downloaded and run on the client computer in response to a request from the client's browser. The client computer never has a permanent copy of the program; a new copy is downloaded and run every time it is required. The program is not run directly on the client computer's operating system, and it has no direct access to the client computer's hard disks or memory. Instead, the browser behaves like an operating system, managing the downloaded program's use of the client machine's resources.

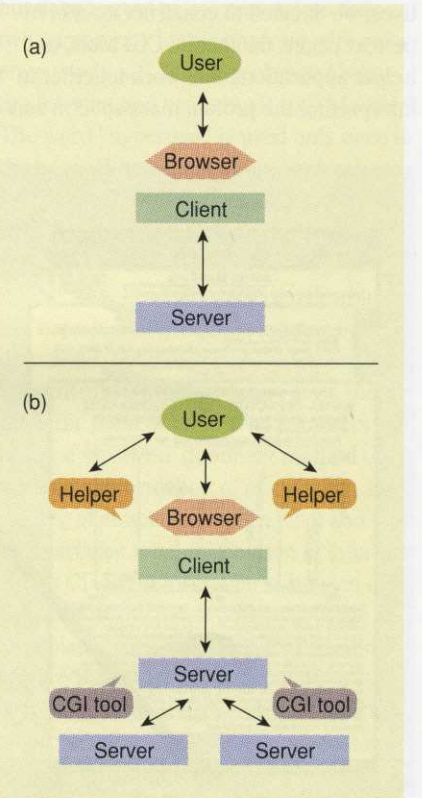


Figure 1. Comparison of (a) simple client–server architecture and (b) the more complex model chosen for PROWL.

The arrows indicate the main channels of information flow between elements of the resource. Some lesser channels of information flow, such as swapping information between helper applications, have been omitted for clarity.

This type of program is written to conform to a set of browser-based standards rather than a set of operating system standards. The two most popular of these browser-based standards are Java and Active-X. This style of client–server behavior is potentially useful but is limited by the speed of both the client computer and the computer's network connection. Any useful software requires the downloading of many kilobytes of executable code, which can be a slow process. Because the program runs using the browser rather than the operating system, it is necessarily slower than a normal program that has direct access to the operating system. Also, for security reasons, these programs cannot access the computer's local disks or store the results of a calculation or manipulation.

An enhanced client–server model

After considering the available alternatives, we decided to construct a set of hypertext pages, databases, CGI tools, and helper applications that work together in interpreting the protein mass spectra contained in MassBank. This suite of modules is what we call PROWL.

Figure 1a shows the simplest client–server model; Figure 1b describes the more versatile model used for PROWL. In PROWL, none of the server-based databases can be accessed directly by the client; all database queries are made through a set of CGI tools that allows users to browse through complex data sets (such as mass spectrum raw data files) as easily as possible. These tools link together mass spectra, protein sequences, and information about proteins and peptides in a straightforward manner. The only direct access to the server is to obtain hypertext pages of information (written using hypertext markup language, HTML) or virtual reality worlds (written using virtual reality modeling language, VRML).

At first glance, some of the CGI tools and helper applications might seem redundant. It is reasonable to ask why a helper application that plots and analyzes mass spectra should be included when a CGI tool exists for that purpose. The answer is that both types of software have limitations and strengths. A CGI tool that shows a simple plot of a mass spectrum embedded in a hypertext page allows a user to decide if the data in the spectrum are of any interest. However, it is currently impossible to produce a CGI program with the sort of advanced graphical user interface that most people expect from data analysis software. To achieve the speed of operation and visual elegance that users are now familiar with, it is necessary to download the data and run a fast, optimized program on the client machine.

Helper applications were chosen as client-based interpretation software rather than as plug-ins so that users could examine downloaded data without using a browser. Plug-ins are also limited in the types of window and menu structures that they can draw. Java applications are not practical for the demanding data manipulation necessary for analyzing protein mass spectra because of the complexity and size of application necessary. A minimally useful mass spectrum analysis program is approximately 0.25 MB, and a good program is nearly 0.5 MB. Current network data transfer rates make it very cumbersome to download such a large program every time a new spectrum is examined. The window and menu interfaces allowed by Java programs are currently very limited, leading to the same objection cited above for plug-ins.

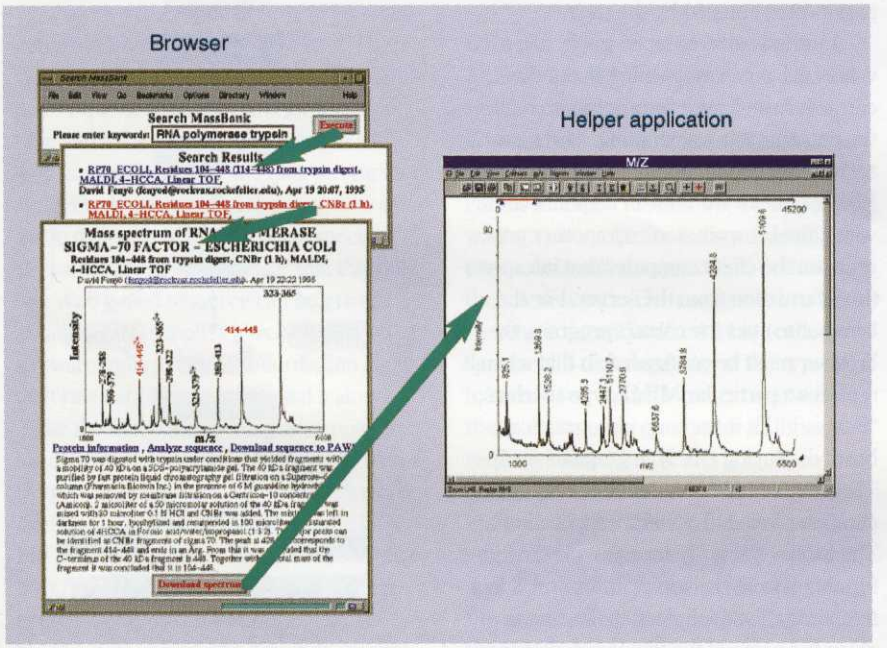


Figure 2. An example of how MassBank can be used to search for mass spectra.

The keywords "RNA polymerase trypsin" are used to search MassBank, and two mass spectra corresponding to a domain of *Escherichia coli* RNA polymerase σ^{70} factor are found. The mass spectrum of a CNBr digest of the protein domain is selected for further examination, and the mass spectrum appears, together with a description of the experimental conditions, in the browser window. Simple analysis of the mass spectrum can be performed with the Web browser. For more advanced analysis, the spectrum can be downloaded and analyzed on the user's computer with the helper application M/Z.

Client–server–helper examples

The full set of databases, CGI tools, and helper applications currently available in PROWL is given in the box on p. 723 A. The best way to illustrate the usefulness of the sort of structure shown in Figure 1b is to give some examples from PROWL. One good example is the method chosen for querying and examining entries in MassBank. Another example is the method for obtaining and analyzing protein sequences with respect to protein MS experiments through ProteinInfo. In a real situation, these two functions are closely linked and are usually performed together; for clarity, we will consider them separately.

MassBank—examining mass spectral data. MassBank is a collection of protein mass spectral data, including raw mass spectra, experimental conditions, comments, and information about the protein being analyzed. The proteins

can be derived from natural or recombinant sources (with the primary structure fully known, partially known, or unknown). MassBank accommodates mass spectra of pure proteins, mixtures, enzymatically or chemically degraded proteins, and mass spectrometrically fragmented peptides. An example of how MassBank can be searched to examine a deposited mass spectrum is shown in Figure 2. The keywords are entered and submitted to a CGI tool that searches MassBank and returns a list of hypertext links to the relevant mass spectra.

By following one of the hypertext links in the list, the corresponding mass spectrum can be examined with annotated information concerning the experimental conditions, the identity and history of the sample, and so forth. A CGI tool is provided for simple analysis of the mass spectrum, such as determining peak masses and expansion of spectrum regions of interest, using any standard Web browser. More advanced analysis can be performed by downloading the mass spectrum and analyzing it with the program M/Z, which functions as a helper application to the local Web browser. Hypertext links can also be followed to obtain information about the protein, analyze the amino acid sequence, and send an e-mail to the author to ask questions about experimental conditions.

ProteinInfo—making a hypothesis. ProteinInfo contains a collection of protein sequence databases that include the primary structure of known proteins and genes. These primary structures can be used to form hypotheses about the outcome of protein chemistry experiments, as monitored by MS. These hypotheses can then be compared with the results of the real experiment, pointing out places where the sequence in the database does not agree with the experiment. The reasons for this sort of disagreement vary depending on the protein being examined and include mutations, post-translational modifications, or chemical changes caused by environmental factors. This type of hypothesis testing is at the core of many types of protein MS experiments.

An example of a ProteinInfo search is shown in Figure 3. A database is selected, the kingdom that the species belongs to is specified, and the keywords are submitted.

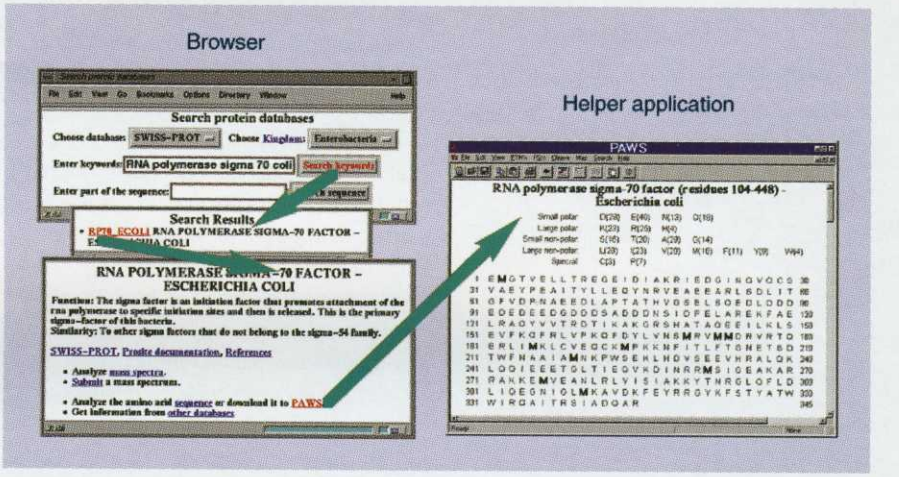


Figure 3. An example of how ProteinInfo can be used to obtain information about proteins.

All sequences from enterobacteria in the SWISS-PROT database are searched with the keywords "RNA polymerase sigma 70 coli". Only one protein is found—the *Escherichia coli* RNA polymerase σ^{70} cofactor. A hypertext link leads to a page containing information from the SWISS-PROT protein sequence database about RNA polymerase σ^{70} cofactor, and the amino acid sequence is downloaded for analysis with the helper application PAWS.

The ProteinInfo CGI tool searches the database and returns a list of hypertext links to matching proteins. By following one of the links, the user will receive a page containing information about the selected protein. This page will also contain hypertext links to mass spectra in MassBank, literature references, information from other protein databases, and a CGI tool for protein primary structure analysis and for downloading the amino acid sequence to PAWS, a helper application to the local Web browser. Using PAWS and the sequence analysis CGI tool, it is possible to calculate masses of peptides from enzymatic or chemical degradation of the protein and masses of ions produced in mass spectrometric fragmentation of peptides. It is also possible to obtain amino acid compositions and find signal peptides, as well as protein motifs (such as glycosylation and phosphorylation sites).

The future of Internet-based chemistry resources

The pace of change in information exchange technology is so rapid that it is difficult to formulate any clear vision of the future. Even experts who stay in contact with the most recent developments can be caught flat-footed. Nicholas Negroponte, head of the MIT Media Lab, does not explicitly refer to Internet technology in his book *Being Digital* (*8*), which was published in 1995. He does mention three times in the book a browser called Mosaic (the prototype for Netscape's Navigator) but devotes, out of 243 pages, less than two paragraphs to this program with references such as "...the current rage about the Internet and browsing it with Mosaic." The word "hypertext" is used only once in the book. When surveyed in September 1996, a local bookstore had 30 meters of shelf space devoted to Internet browsers and hypertext.

It is impossible to say in detail what will happen in the far future (one year from now), but we may speculate on developments in the near future from the scientific point of view. Web browsers will become (are?) the dominant method of retrieving information of many types, including abstracts, reference data, and short articles. The distribution of information in CD-ROM format for use by an individual on a single computer will cease and be replaced by access to Web sites that require passwords and subscription fees or the mounting to CD-ROM information on protected internal networks accessible by browsers, hidden behind "firewalls" from the rest of the Internet. Anything that is available by Internet will be used in preference to similar information that is available only in a library. As has always been the case, the use of information will be dictated by its accessibility.

The vision of a global information library accessible from any location is still

considerably in the future. Use of Web sites is currently dictated by the rate at which data can be passed between client and server machines. Relatively minor slowdowns in the exchange of information—for example, a one-minute wait for an HTML page to appear on a browser—result in most people abandoning that transaction and moving elsewhere. These slowdowns are caused by fundamental problems with the way the hardware carrying the information interacts with the transmission protocols responsible for internetwork communication. Many sites in Europe cannot be reached from North America during business hours (and vice versa). One method for alleviating this problem is to set up "mirror" sites at several well-chosen locations, but this solution leads to serious site administration difficulties.

Another problem involves information retrieval from a database. The current software for querying databases is often crude, and no standard question-and-answer formats exist. The less-than-subtle differences between conversational English questions and the simplistic Boolean logical questions that most interfaces understand make it unnecessarily difficult for even experienced users to find information. Some sites have documentation explaining their particular type of query format, but it is often framed in impenetrable, fragmentary style.

This problem can be extremely frustrating when dealing with protein sequence databases. The proliferation of very similar names and abbreviations for proteins makes it necessary to formulate a question capable of retrieving your protein rather than several thousand uninteresting entries that must be sorted through manually. Frequently, formulating that question becomes a time-consuming trial-and-error process, with the user posing question after question until something familiar appears in the result list. A simple graphical representation that would allow a user to navigate through a database by point-and-click steps rather than engaging in a typewritten syllogistic debate with an anything-but-heuristic interface would improve the current situation dramatically.

PROWL is currently being revised to address the problem of how to create a database query system that more accurately reflects how the data will be used rather than how it is stored. Rather than organizing the queries around trying to find a file containing a mass spectrum, the new queries will be structured so that the user can ask questions about proteins and the laboratory experiments performed on proteins, receiving in response the relevant mass spectral information. This level of sophistication can be achieved by organizing the internal structure of the databases around the concept of the physical objects being studied (in our case, proteins). Ultimately, queries will be translated by "intelligent" software so that a multitude of databases can be searched simultaneously for the most relevant information.

We gratefully acknowledge the financial support of the National Institutes of Health (Grant RR00862), the National Science Foundation (Grant 9630936), and the Skirball Institute of Biomedical Research at the New York University Medical Center. We would also like to thank Ole Jensen, Ole Vorm, Mathias Mann, Karl Clauser, Salvatore Sechi, Yingming Zhao, Steven Cohen, Rong Wang, Jun Qin, Urooj Mirza, James Carroll, Janet Brostowin, Jules Schear, Werner Ens, and Mihkael Velikanov for their input to the project.

References

(1) The World Wide Web (WWW) conference series; http://www.igd.fhg.de/www/www95/documentation/conference.html.
(2) Benson, D. A.; Boguski, M.; Lippman, D. J.; Ostell, J. *Nucleic Acid Res.* **1996**, *24*, 1.
(3) Rodriguez-Tomé, P.; Stoehr, P. J.; Cameron, G. N.; Flores, T. P. *Nucleic Acid Res.* **1996**, *24*, 6.
(4) Bairoch, A.; Boeckmann, B. *Nucleic Acid Res.* **1996,** *24*, 21.
(5) George, D. G.; Barker, W. C.; Mewes, H. W.; Pfeiffer, F.; Tsugita, A. *Nucleic Acid Res.* **1996**, *24*, 17.
(6) Pattabiraman, N.; Namboodiri, K.; Lowrey, A.; Gaber, B. P. *Protein Seq. Data Anal.* **1990**, *3*, 387.
(7) Bleasby, A. J.; Akrigg, D.; Attwood, T. A. *Nucleic Acid Res.* **1994**, *21*, 3574.
(8) Negroponte, N. *Being Digital*; Alfred A. Knopf: New York, 1995.

Ronald C. Beavis is Associate Professor of Pharmacology and Chemistry at New York University Medical Center. Brian T. Chait is Camille and Henry Dreyfus Professor at The Rockefeller University and Director of the National Resource for the Mass Spectrometric Analysis of Biological Macromolecules. Fenyö and Zhang are research associates in Chait's laboratory. Address correspondence to Chait at The Rockefeller University, 1230 York Ave., New York, NY 10021-6399.