1

BIOINFORMATICS AND THE INTERNET

Andreas D. Baxevanis

Genome Technology Branch
National Human Genome Research Institute
National Institutes of Health
Bethesda, Maryland

Bioinformatics represents a new, growing area of science that uses computational approaches to answer biological questions. Answering these questions requires that investigators take advantage of large, complex data sets (both public and private) in a rigorous fashion to reach valid, biological conclusions. The potential of such an approach is beginning to change the fundamental way in which basic science is done, helping to more efficiently guide experimental design in the laboratory.

With the explosion of sequence and structural information available to researchers, the field of bioinformatics is playing an increasingly large role in the study of fundamental biomedical problems. The challenge facing computational biologists will be to aid in gene discovery and in the design of molecular modeling, site-directed mutagenesis, and experiments of other types that can potentially reveal previously unknown relationships with respect to the structure and function of genes and proteins. This challenge becomes particularly daunting in light of the vast amount of data that has been produced by the Human Genome Project and other systematic sequencing efforts to date.

Before embarking on any practical discussion of computational methods in solving biological problems, it is necessary to lay the common groundwork that will enable users to both access and implement the algorithms and tools discussed in this book. We begin with a review of the Internet and its terminology, discussing major Internet protocol classes as well, without becoming overly engaged in the engineering

minutiae underlying these protocols. A more in-depth treatment on the inner workings of these protocols may be found in a number of well-written reference books intended for the lay audience (Rankin, 1996; Conner-Sax and Krol, 1999; Kennedy, 1999). This chapter will also discuss matters of connectivity, ranging from simple modem connections to digital subscriber lines (DSL). Finally, we will address one of the most common problems that has arisen with the proliferation of Web pages throughout the world—finding useful information on the World Wide Web.

INTERNET BASICS

Despite the impression that it is a single entity, the Internet is actually a network of networks, composed of interconnected local and regional networks in over 100 countries. Although work on remote communications began in the early 1960s, the true origins of the Internet lie with a research project on networking at the Advanced Research Projects Agency (ARPA) of the US Department of Defense in 1969 named ARPANET. The original ARPANET connected four nodes on the West Coast, with the immediate goal of being able to transmit information on defense-related research between laboratories. A number of different network projects subsequently surfaced, with the next landmark developments coming over 10 years later. In 1981, BITNET (''Because It's Time'') was introduced, providing point-to-point connections between universities for the transfer of electronic mail and files. In 1982, ARPA introduced the Transmission Control Protocol (TCP) and the Internet Protocol (IP); TCP/IP allowed different networks to be connected to and communicate with one another, creating the system in place today. A number of references chronicle the development of the Internet and communications protocols in detail (Quarterman, 1990; Froehlich and Kent, 1991; Conner-Sax and Krol, 1999). Most users, however, are content to leave the details of *how* the Internet works to their systems administrators; the relevant fact to most is that it *does* work.

Once the machines on a network have been connected to one another, there needs to be an unambiguous way to specify a single computer so that messages and files actually find their intended recipient. To accomplish this, all machines directly connected to the Internet have an *IP number*. IP addresses are unique, identifying one and only one machine. The IP address is made up of four numbers separated by periods; for example, the IP address for the main file server at the National Center for Biotechnology Information (NCBI) at the National Institutes of Health (NIH) is 130.14.25.1. The numbers themselves represent, from left to right, the domain (130.14 for NIH), the subnet (.25 for the National Library of Medicine at NIH), and the machine itself (.1). The use of IP numbers aids the computers in directing data; however, it is obviously very difficult for users to remember these strings, so IP addresses often have associated with them a *fully qualified domain name* (FQDN) that is dynamically translated in the background by *domain name servers*. Going back to the NCBI example, rather than use `130.14.25.1` to access the NCBI computer, a user could instead use `ncbi.nlm.nih.gov` and achieve the same result. Reading from left to right, notice that the IP address goes from least to most specific, whereas the FQDN equivalent goes from most specific to least. The name of any given computer can then be thought of as taking the general form *computer.domain*, with the top-level domain (the portion coming after the last period in the FQDN) falling into one of the broad categories shown in Table 1.1. Outside the

TABLE 1.1. Top-Level Doman Names

TOP-LEVEL DOMAIN NAMES	
.com	Commercial site
.edu	Educational site
.gov	Government site
.mil	Military site
.net	Gateway or network host
.org	Private (usually not-for-profit) organizations
EXAMPLES OF TOP-LEVEL DOMAIN NAMES USED OUTSIDE THE UNITED STATES	
.ca	Canadian site
.ac.uk	Academic site in the United Kingdom
.co.uk	Commercial site in the United Kingdom
GENERIC TOP-LEVEL DOMAINS PROPOSED BY IAHC	
.firm	Firms or businesses
.shop	Businesses offering goods to purchase (stores)
.web	Entities emphasizing activities relating to the World Wide Web
.arts	Cultural and entertainment organizations
.rec	Recreational organizations
.info	Information sources
.nom	Personal names (e.g., *yourlastname.nom*)

A complete listing of domain suffixes, including country codes, can be found at http://www.currents.net/resources/directory/noframes/nf.domains.html.

United States, the top-level domain names *may* be replaced with a two-letter code specifying the country in which the machine is located (e.g., .ca for Canada and .uk for the United Kingdom). In an effort to anticipate the needs of Internet users in the future, as well as to try to erase the arbitrary line between top-level domain names based on country, the now-dissolved International Ad Hoc Committee (IAHC) was charged with developing a new framework of generic top-level domains (gTLD). The new, recommended gTLDs were set forth in a document entitled *The Generic Top Level Domain Memorandum of Understanding* (gTLD-MOU); these gTLDs are overseen by a number of governing bodies and are also shown in Table 1.1.

The most concrete measure of the size of the Internet lies in actually counting the number of machines physically connected to it. The Internet Software Consortium (ISC) conducts an Internet Domain Survey twice each year to count these machines, otherwise known as *hosts*. In performing this survey, ISC considers not only how many hostnames have been assigned, but how many of those are actually in use; a hostname might be issued, but the requestor may be holding the name in abeyance for future use. To test for this, a representative sample of host machines are sent a probe (a ''ping''), with a signal being sent back to the originating machine if the host was indeed found. The rate of growth of the number of hosts has been phenomenal; from a paltry 213 hosts in August 1981, the Internet now has more than 60 million ''live'' hosts. The doubling time for the number of hosts is on the order of 18 months. At this time, most of this growth has come from the commercial sector, capitalizing on the growing popularity of multimedia platforms for advertising and communications such as the World Wide Web.

CONNECTING TO THE INTERNET

Of course, before being able to use all the resources that the Internet has to offer, one needs to actually make a physical connection between one's own computer and ''the information superhighway.'' For purposes of this discussion, the elements of this connection have been separated into two discrete parts: the actual, physical connection (meaning the ''wire'' running from one's computer to the Internet backbone) and the service provider, who handles issues of routing and content once connected. Keep in mind that, in practice, these are not necessarily treated as two separate parts—for instance, one's service provider may also be the same company that will run cables or fibers right into one's home or office.

Copper Wires, Coaxial Cables, and Fiber Optics

Traditionally, users attempting to connect to the Internet away from the office had one and only one option—a modem, which uses the existing copper twisted-pair cables carrying telephone signals to transmit data. Data transfer rates using modems are relatively slow, allowing for data transmission in the range of 28.8 to 56 kilobits per second (kbps). The problem with using conventional copper wire to transmit data lies not in the copper wire itself but in the switches that are found along the way that route information to their intended destinations. These switches were designed for the efficient and effective transfer of voice data but were never intended to handle the high-speed transmission of data. Although most people still use modems from their home, a number of new technologies are already in place and will become more and more prevalent for accessing the Internet away from hardwired Ethernet networks. The maximum speeds at which each of the services that are discussed below can operate are shown in Figure 1.1.

The first of these ''new solutions'' is the integrated services digital network or ISDN. The advent of ISDN was originally heralded as the way to bring the Internet into the home in a speed-efficient manner; however, it required that special wiring be brought into the home. It also required that users be within a fixed distance from a central office, on the order of 20,000 feet or less. The cost of running this special, dedicated wiring, along with a per-minute pricing structure, effectively placed ISDN out of reach for most individuals. Although ISDN is still available in many areas, this type of service is quickly being supplanted by more cost-effective alternatives.

In looking at alternatives that did not require new wiring, cable television providers began to look at ways in which the coaxial cable already running into a substantial number of households could be used to also transmit data. Cable companies are able to use bandwidth that is not being used to transmit television signals (effectively, unused channels) to push data into the home at very high speeds, up to 4.0 megabits per second (Mbps). The actual computer is connected to this network through a cable modem, which uses an Ethernet connection to the computer and a coaxial cable to the wall. Homes in a given area all share a single cable, in a wiring scheme very similar to how individual computers are connected via the Ethernet in an office or laboratory setting. Although this branching arrangement can serve to connect a large number of locations, there is one major disadvantage: as more and more homes connect through their cable modems, service effectively slows down as more signals attempt to pass through any given node. One way of circumventing

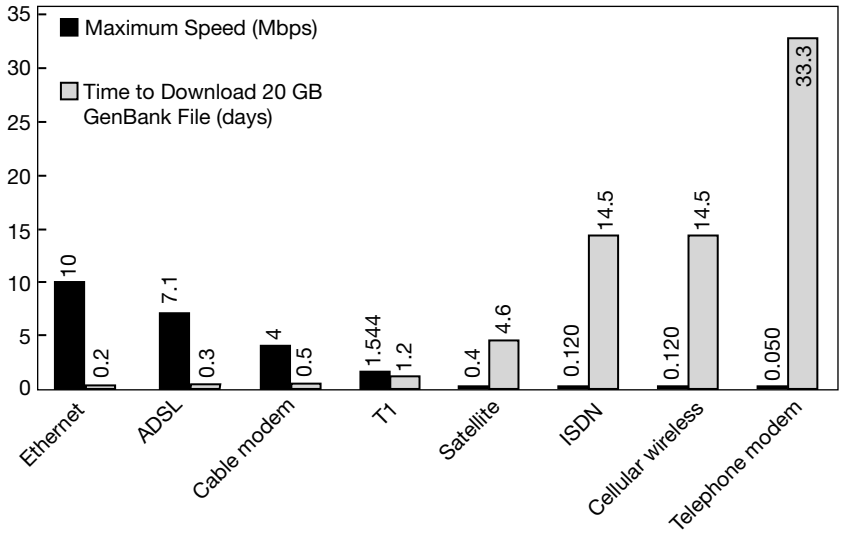


Figure 1.1. Performance of various types of Internet connections, by maximum throughput. The numbers indicated in the graph refer to peak performance; often times, the actual performance of any given method may be on the order of one-half slower, depending on configurations and system conditions.

this problem is the installation of more switching equipment and reducing the size of a given ''neighborhood.''

Because the local telephone companies were the primary ISDN providers, they quickly turned their attention to ways that the existing, conventional copper wire already in the home could be used to transmit data at high speed. The solution here is the digital subscriber line or DSL. By using new, dedicated switches that are designed for rapid data transfer, DSL providers can circumvent the old voice switches that slowed down transfer speeds. Depending on the user's distance from the central office and whether a particular neighborhood has been wired for DSL service, speeds are on the order of 0.8 to 7.1 Mbps. The data transfers do not interfere with voice signals, and users can use the telephone while connected to the Internet; the signals are ''split'' by a special modem that passes the data signals to the computer and a microfilter that passes voice signals to the handset. There is a special type of DSL called *asynchronous* DSL or ADSL. This is the variety of DSL service that is becoming more and more prevalent. Most home users download much more information than they send out; therefore, systems are engineered to provide super-fast transmission in the ''in'' direction, with transmissions in the ''out'' direction being 5–10 times slower. Using this approach maximizes the amount of bandwidth that can be used without necessitating new wiring. One of the advantages of ADSL over cable is that ADSL subscribers effectively have a direct line to the central office, meaning that they do not have to compete with their neighbors for bandwidth. This, of course, comes at a price; at the time of this writing, ADSL connectivity options were on the order of twice as expensive as cable Internet, but this will vary from region to region.

Some of the newer technologies involve wireless connections to the Internet. These include using one's own cell phone or a special cell phone service (such as

Ricochet) to upload and download information. These cellular providers can provide speeds on the order of 28.8–128 kbps, depending on the density of cellular towers in the service area. Fixed-point wireless services can be substantially faster because the cellular phone does not have to ''find'' the closest tower at any given time. Along these same lines, satellite providers are also coming on-line. These providers allow for data download directly to a satellite dish with a southern exposure, with uploads occuring through traditional telephone lines. Along the satellite option has the potential to be among the fastest of the options discussed, current operating speeds are only on the order of 400 kbps.

Content Providers vs. ISPs

Once an appropriately fast and price-effective connectivity solution is found, users will then need to actually connect to some sort of service that will enable them to traverse the Internet space. The two major categories in this respect are *online services* and *Internet service providers* (ISPs). Online services, such as America Online (AOL) and CompuServe, offer a large number of interactive digital services, including information retrieval, electronic mail (E-mail; see below), bulletin boards, and ''chat rooms,'' where users who are online at the same time can converse about any number of subjects. Although the online services now provide access to the World Wide Web, most of the specialized features and services available through these systems reside in a proprietary, closed network. Once a connection has been made between the user's computer and the online service, one can access the special features, or content, of these systems without ever leaving the online system's host computer. Specialized content can range from access to online travel reservation systems to encyclopedias that are constantly being updated—items that are not available to nonsubscribers to the particular online service.

Internet service providers take the opposite tack. Instead of focusing on providing content, the ISPs provide the tools necessary for users to send and receive E-mail, upload and download files, and navigate around the World Wide Web, finding information at remote locations. The major advantage of ISPs is connection speed; often the smaller providers offer faster connection speeds than can be had from the online services. Most ISPs charge a monthly fee for unlimited use.

The line between online services and ISPs has already begun to blur. For instance, AOL's now monthly flat-fee pricing structure in the United States allows users to obtain all the proprietary content found on AOL as well as all the Internet tools available through ISPs, often at the same cost as a simple ISP connection. The extensive AOL network puts access to AOL as close as a local phone call in most of the United States, providing access to E-mail no matter where the user is located, a feature small, local ISPs cannot match. Not to be outdone, many of the major national ISP providers now also provide content through the concept of *portals*. Portals are Web pages that can be customized to the needs of the individual user and that serve as a jumping-off point to other sources of news or entertainment on the Net. In addition, many national firms such as Mindspring are able to match AOL's ease of connectivity on the road, and both ISPs and online providers are becoming more and more generous in providing users the capacity to publish their own Web pages. Developments such as this, coupled with the move of local telephone and cable companies into providing Internet access through new, faster fiber optic net-

works, foretell major changes in how people will access the Net in the future, changes that should favor the end user in both price and performance.

ELECTRONIC MAIL

Most people are introduced to the Internet through the use of electronic mail or *E-mail*. The use of E-mail has become practically indispensable in many settings because of its convenience as a medium for sending, receiving, and replying to messages. Its advantages are many:

- It is much quicker than the postal service or ''snail mail.''
- Messages tend to be much clearer and more to the point than is the case for typical telephone or face-to-face conversations.
- Recipients have more flexibility in deciding whether a response needs to be sent immediately, relatively soon, or at all, giving individuals more control over workflow.
- It provides a convenient method by which messages can be filed or stored.
- There is little or no cost involved in sending an E-mail message.

These and other advantages have pushed E-mail to the forefront of interpersonal communication in both industry and the academic community; however, users should be aware of several major disadvantages. First is the issue of security. As mail travels toward its recipient, it may pass through a number of remote nodes, at any one of which the message may be intercepted and read by someone with high-level access, such as a systems administrator. Second is the issue of privacy. In industrial settings, E-mail is often considered to be an asset of the company for use in official communication only and, as such, is subject to monitoring by supervisors. The opposite is often true in academic, quasi-academic, or research settings; for example, the National Institutes of Health's policy encourages personal use of E-mail within the bounds of certain published guidelines. The key words here are ''published guidelines''; no matter what the setting, users of E-mail systems should always find out their organization's policy regarding appropriate use and confidentiality so that they may use the tool properly and effectively. An excellent, basic guide to the effective use of E-mail (Rankin, 1996) is recommended.

Sending E-Mail. E-mail addresses take the general form *user@computer.domain*, where *user* is the name of the individual user and *computer.domain* specifies the actual computer that the E-mail account is located on. Like a postal letter, an E-mail message is comprised of an *envelope* or *header*, showing the E-mail addresses of sender and recipient, a line indicating the subject of the E-mail, and information about how the E-mail message actually traveled from the sender to the recipient. The header is followed by the actual message, or *body*, analogous to what would go inside a postal envelope. Figure 1.2 illustrates all the components of an E-mail message.

E-mail programs vary widely, depending on both the platform and the needs of the users. Most often, the characteristics of the local area network (LAN) dictate what types of mail programs can be used, and the decision is often left to systems

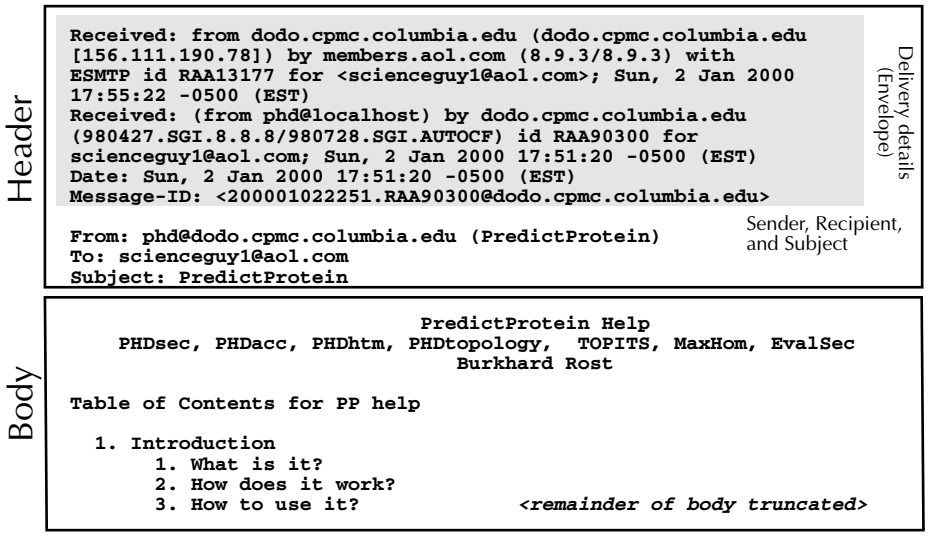


Figure 1.2. Anatomy of an E-mail message, with relevant components indicated. This message is an automated reply to a request for help file for the PredictProtein E-mail server.

administrators rather than individual users. Among the most widely used E-mail packages with a graphical user interface are Eudora for the Macintosh and both Netscape Messenger and Microsoft Exchange for the Mac, Windows, and UNIX platforms. Text-based E-mail programs, which are accessed by logging in to a UNIX-based account, include Elm and Pine.

Bulk E-Mail. As with postal mail, there has been an upsurge in ''spam'' or ''junk E-mail,'' where companies compile bulk lists of E-mail addresses for use in commercial promotions. Because most of these lists are compiled from online registration forms and similar sources, the best defense for remaining off these bulk E-mail lists is to be selective as to whom E-mail addresses are provided. Most newsgroups keep their mailing lists confidential; if in doubt and if this is a concern, one should ask.

E-Mail Servers. Most often, E-mail is thought of a way to simply send messages, whether it be to one recipient or many. It is also possible to use E-mail as a mechanism for making predictions or retrieving records from biological databases. Users can send E-mail messages in a format defining the action to be performed to remote computers known as *servers*; the servers will then perform the desired operation and E-mail back the results. Although this method is not interactive (in that the user cannot adjust parameters or have control over the execution of the method in real time), it does place the responsibility for hardware maintenance and software upgrades on the individuals maintaining the server, allowing users to concentrate on their results instead of on programming. The use of a number of E-mail servers is discussed in greater detail in context in later chapters. For most of these servers, sending the message `help` to the server E-mail address will result in a detailed set of instructions for using that server being returned, including ways in which queries need to be formatted.

Aliases and Newsgroups. In the example in Figure 1.2, the E-mail message is being sent to a single recipient. One of the strengths of E-mail is that a single piece of E-mail can be sent to a large number of people. The primary mechanism for doing this is through *aliases*; a user can define a group of people within their mail program and give the group a special name or alias. Instead of using individual E-mail addresses for all of the people in the group, the user can just send the E-mail to the alias name, and the mail program will handle broadcasting the message to each person in that group. Setting up alias names is a tremendous time-saver even for small groups; it also ensures that all members of a given group actually receive all E-mail messages intended for the group.

The second mechanism for broadcasting messages is through *newsgroups*. This model works slightly differently in that the list of E-mail addresses is compiled and maintained on a remote computer through subscriptions, much like magazine subscriptions. To participate in a newsgroup discussions, one first would have to subscribe to the newsgroup of interest. Depending on the newsgroup, this is done either by sending an E-mail to the host server or by visiting the host's Web site and using a form to subscribe. For example, the BIOSCI newsgroups are among the most highly trafficked, offering a forum for discussion or the exchange of ideas in a wide variety of biological subject areas. Information on how to subscribe to one of the constituent BIOSCI newsgroups is posted on the BIOSCI Web site. To actually participate in the discussion, one would simply send an E-mail to the address corresponding to the group that you wish to reach. For example, to post messages to the computational biology newsgroup, mail would simply be addressed to `comp-bio@net.bio.net`, and, once that mail is sent, *everyone* subscribing to that newsgroup would receive (and have the opportunity to respond to) that message. The ease of reaching a large audience in such a simple fashion is both a blessing and a curse, so many newsgroups require that postings be reviewed by a moderator before they get disseminated to the individual subscribers to assure that the contents of the message are actually of interest to the readers.

It is also possible to participate in newsgroups without having each and every piece of E-mail flood into one's private mailbox. Instead, interested participants can use news-reading software, such as NewsWatcher for the Macintosh, which provides access to the individual messages making up a discussion. The major advantage is that the user can pick and choose which messages to read by scanning the subject lines; the remainder can be discarded by a single operation. NewsWatcher is an example of what is known as a *client-server application*; the client software (here, NewsWatcher) runs on a client computer (a Macintosh), which in turn interacts with a machine at a remote location (the server). Client-server architecture is interactive in nature, with a direct connection being made between the client and server machines.

Once NewsWatcher is started, the user is presented with a list of newsgroups available to them (Fig. 1.3). This list will vary, depending on the user's location, as system administrators have the discretion to allow or to block certain groups at a given site. From the rear-most window in the figure, the user double-clicks on the newsgroup of interest (here, *bionet.genome.arabidopsis*), which spawns the window shown in the center. At the top of the center window is the current unread message count, and any message within the list can be read by double-clicking on that particular line. This, in turn, spawns the last window (in the foreground), which shows the actual message. If a user decides not to read any of the messages, or is done

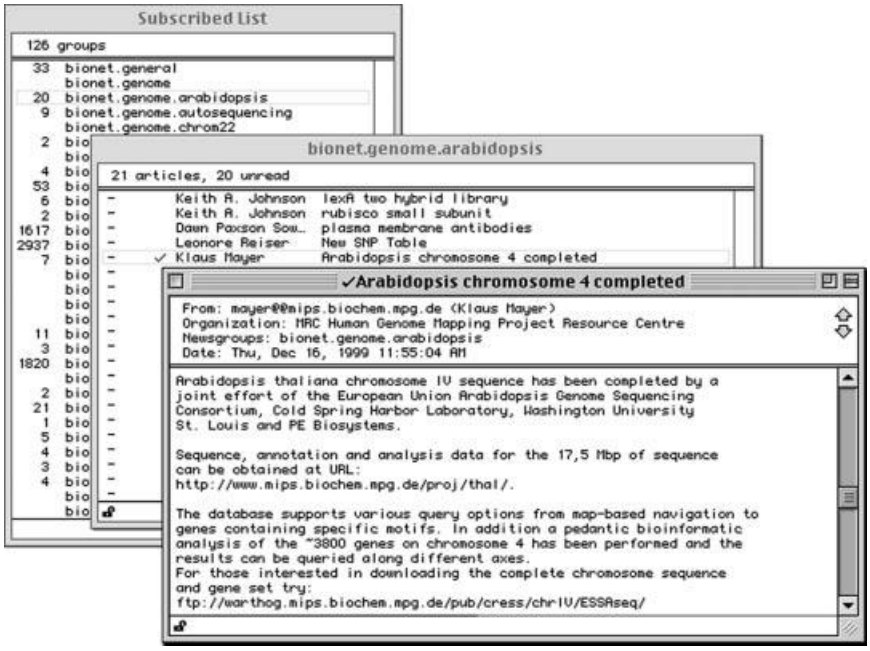


Figure 1.3. Using NewsWatcher to read postings to newsgroups. The list of newsgroups that the user has subscribed to is shown in the Subscribed List window (*left*). The list of new postings for the highlighted newsgroup (*bionet.genome.arabidopsis*) is shown in the center window. The window in the foreground shows the contents of the posting selected from the center window.

reading individual messages, the balance of the messages within the newsgroup (center) window can be deleted by first choosing Select All from the File menu and then selecting Mark Read from the News menu. Once the newsgroup window is closed, the unread message count is reset to zero. Every time NewsWatcher is restarted, it will automatically poll the news server for new messages that have been created since the last session. As with most of the tools that will be discussed in this chapter, news-reading capability is built into Web browsers such as Netscape Navigator and Microsoft Internet Explorer.

FILE TRANSFER PROTOCOL

Despite the many advantages afforded by E-mail in transmitting messages, many users have no doubt experienced frustration in trying to transmit files, or *attachments*, along with an E-mail message. The mere fact that a file can be attached to an E-mail message and sent does not mean that the recipient will be able to detach, decode, and actually use the attached file. Although more cross-platform E-mail packages such as Microsoft Exchange are being developed, the use of different E-mail packages by people at different locations means that sending files via E-mail is not an effective, foolproof method, at least in the short term. One solution to this

problem is through the use of a *file transfer protocol* or FTP. The workings of FTP are quite simple: a connection is made between a user's computer (the *client*) and a remote server, and that connection remains in place for the duration of the FTP session. File transfers are very fast, at rates on the order of 5–10 kilobytes per second, with speeds varying with the time of day, the distance between the client and server machines, and the overall traffic on the network.

In the ordinary case, making an FTP connection and transferring files requires that a user have an account on the remote server. However, there are many files and programs that are made freely available, and access to those files does not require having an account on each and every machine where these programs are stored. Instead, connections are made using a system called *anonymous FTP*. Under this system, the user connects to the remote machine and, instead of entering a username/password pair, types `anonymous` as the username and enters their E-mail address in place of a password. Providing one's E-mail address allows the server's system administrators to compile access statistics that may, in turn, be of use to those actually providing the public files or programs. An example of an anonymous FTP session using UNIX is shown in Figure 1.4.

Although FTP actually occurs within the UNIX environment, Macintosh and PC users can use programs that rely on graphical user interfaces (GUI, pronounced

```
$ ftp ftp.bio.indiana.edu
Connected to magpie.bio.indiana.edu.
220 iubio.bio.indiana.edu FTP server ready.
Name: anonymous
331 Guest login ok, send your complete e-mail address as password.
Password: ********
230-                    Welcome to IUBio archive!
230-
230-  This is a user-supported archive for biology software and data.
230-
230-       See the file Archive.Doc for details of this archive.
230-
230-          See IUBio Bio-Mirror archive of large data sets at
230-     ftp to iubio.bio.indiana.edu, user: iubio, password: iubio
230-   This includes GenBank, EMBL and DDBJ and other biosequence data.
230-
230-     Report problems, uploads and other matters via e-mail to
230-                     archive@bio.indiana.edu.
230-
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> cd /molbio/align/clustal
250 CWD command successful.
ftp> get clustalw1.75.unix.tar.Z
local: clustalw1.75.unix.tar.Z remote: clustalw1.75.unix.tar.Z
200 PORT command successful.
150 Opening BINARY mode data connection for clustalw1.75.unix.tar.Z (230379 bytes).
226 Transfer complete.
230379 bytes received in 0.45 seconds (500.75 Kbytes/s)
ftp> quit
221-You have transferred 230379 bytes in 1 files.
221-Total traffic for this session was 231859 bytes in 1 transfers.
221-Thank you for using the FTP service on iubio.bio.indiana.edu.
221 Goodbye.
```

Figure 1.4. Using UNIX FTP to download a file. An anonymous FTP session is established with the molecular biology FTP server at the University of Indiana to download the CLUSTAL W alignment program. The user inputs are shown in boldface.

"gooey") to navigate through the UNIX directories on the FTP server. Users need not have any knowledge of UNIX commands to download files; instead, they select from pop-up menus and point and click their way through the UNIX file structure. The most popular FTP program on the Macintosh platform for FTP sessions is Fetch. A sample Fetch window is shown in Figure 1.5 to illustrate the difference between using a GUI-based FTP program and the equivalent UNIX FTP in Figure 1.4. In the figure, notice that the Automatic radio button (near the bottom of the second window under the Get File button) is selected, meaning that Fetch will determine the appropriate type of file transfer to perform. This may be manually overridden by selecting either Text or Binary, depending on the nature of the file being transferred. As a rule, text files should be transferred as Text, programs or executables as Binary, and graphic format files such as PICT and TIFF files as Raw Data.

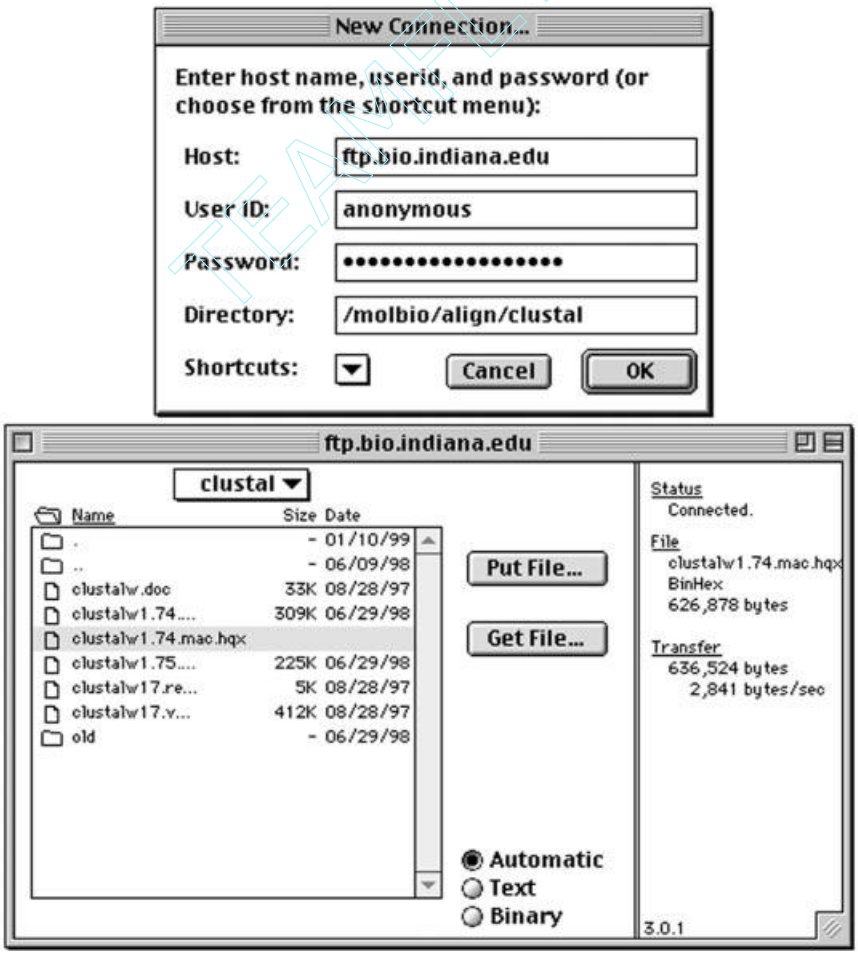


Figure 1.5. Using Fetch to download a file. An anonymous FTP session is established with the molecular biology FTP server at the University of Indiana (*top*) to download the CLUSTAL W alignment program (*bottom*). Notice the difference between this GUI-based program and the UNIX equivalent illustrated in Figure 1.4.

THE WORLD WIDE WEB

Although FTP is of tremendous use in the transfer of files from one computer to another, it does suffer from some limitations. When working with FTP, once a user enters a particular directory, they can only see the names of the directories or files. To actually view what is within the files, it is necessary to physically download the files onto one's own computer. This inherent drawback led to the development of a number of *distributed document delivery systems* (DDDS), interactive client-server applications that allowed information to be viewed without having to perform a download. The first generation of DDDS development led to programs like Gopher, which allowed plain text to be viewed directly through a client-server application. From this evolved the most widely known and widely used DDDS, namely, the World Wide Web. The Web is an outgrowth of research performed at the European Nuclear Research Council (CERN) in 1989 that was aimed at sharing research data between several locations. That work led to a medium through which text, images, sounds, and videos could be delivered to users on demand, anywhere in the world.

Navigation on the World Wide Web

Navigation on the Web does not require advance knowledge of the location of the information being sought. Instead, users can navigate by clicking on specific text, buttons, or pictures. These clickable items are collectively known as *hyperlinks*. Once one of these hyperlinks is clicked, the user is taken to another Web location, which could be at the same site or halfway around the world. Each document displayed on the Web is called a *Web page*, and all of the related Web pages on a particular server are collectively called a *Web site*. Navigation strictly through the use of hyperlinks has been nicknamed ''Web surfing.''

Users can take a more direct approach to finding information by entering a specific address. One of the strengths of the Web is that the programs used to view Web pages (appropriately termed *browsers*) can be used to visit FTP and Gopher sites as well, somewhat obviating the need for separate Gopher or FTP applications. As such, a unified naming convention was introduced to indicate to the browser program both the location of the remote site and, more importantly, the type of information at that remote location so that the browser could properly display the data. This standard-form address is known as a *uniform resource locator*, or URL, and takes the general form *protocol://computer.domain*, where *protocol* specifies the type of site and *computer.domain* specifies the location (Table 1.2). The *http* used for the protocol in World Wide Web URLs stands for *hypertext transfer protocol*, the method used in transferring Web files from the host computer to the client.

TABLE 1.2. Uniform Resource Locator (URL) Format for Each Type of Transfer Protocol

General form	*protocol://computer.domain*
FTP site	*ftp://ftp.ncbi.nlm.nih.gov*
Gopher site	*gopher://gopher.iubio.indiana.edu*
Web site	*http://www.nhgri.nih.gov*

Browsers

Browsers, which are used to look at Web pages, are client-server applications that connect to a remote site, download the requested information at that site, and display the information on a user's monitor, then disconnecting from the remote host. The information retrieved from the remote host is in a platform-independent format named *hypertext markup language* (HTML). HTML code is strictly text-based, and any associated graphics or sounds for that document exist as separate files in a common format. For example, images may be stored and transferred in GIF format, a proprietary format developed by CompuServe for the quick and efficient transfer of graphics; other formats, such as JPEG and BMP, may also be used. Because of this, a browser can display any Web page on any type of computer, whether it be a Macintosh, IBM compatible, or UNIX machine. The text is usually displayed first, with the remaining elements being placed on the page as they are downloaded. With minor exception, a given Web page will look the same when the same browser is used on any of the above platforms. The two major players in the area of browser software are Netscape, with their Communicator product, and Microsoft, with Internet Explorer. As with many other areas where multiple software products are available, the choice between Netscape and Internet Explorer comes down to one of personal preference. Whereas the computer literati will debate the fine points of difference between these two packages, for the average user, both packages perform equally well and offer the same types of features, adequately addressing the Web-browser needs of most users.

It is worth mentioning that, although the Web is by definition a visually-based medium, it is also possible to travel through Web space and view documents without the associated graphics. For users limited to line-by-line terminals, a browser called Lynx is available. Developed at the University of Kansas, Lynx allows users to use their keyboard arrow keys to highlight and select hyperlinks, using their return key the same way that Netscape and Internet Explorer users would click their mouse.

Internet vs. Intranet

The Web is normally thought of as a way to communicate with people at a distance, but the same infrastructure can be used to connect people within an organization. Such *intranets* provide an easily accessible repository of relevant information, capitalizing on the simplicity of the Web interface. They also provide another channel for broadcast or confidential communication within the organization. Having an intranet is of particular value when members of an organization are physically separated, whether in different buildings or different cities. Intranets are protected: that is, people who are not on the organization's network are prohibited from accessing the internal Web pages; additional protections through the use of passwords are also common.

Finding Information on the World Wide Web

Most people find information on the Web the old-fashioned way: by word of mouth, either using lists such as those preceding the References in the chapters of this book or by simply following hyperlinks put in place by Web authors. Continuously clicking from page to page can be a highly ineffective way of finding information, though,

especially when the information sought is of a very focused nature. One way of finding interesting and relevant Web sites is to consult *virtual libraries*, which are curated lists of Web resources arranged by subject. Virtual libraries of special interest to biologists include the WWW Virtual Library, maintained by Keith Robison at Harvard, and the EBI BioCatalog, based at the European Bioinformatics Institute. The URLs for these sites can be found in the list at the end of this chapter.

It is also possible to directly search the Web by using *search engines*. A search engine is simply a specialized program that can perform full-text or keyword searches on databases that catalog Web content. The result of a search is a hyperlinked list of Web sites fitting the search criteria from which the user can visit any or all of the found sites. However, the search engines use slightly different methods in compiling their databases. One variation is the attempt to capture most or all of the text of every Web page that the search engine is able to find and catalog (''Web crawling''). Another technique is to catalog only the title of each Web page rather than its entire text. A third is to consider words that must appear next to each other or only relatively close to one another. Because of these differences in search-engine algorithms, the results returned by issuing the same query to a number of different search engines can produce wildly different results (Table 1.3). The other important feature of Table 1.3 is that most of the numbers are exceedingly large, reflecting the overall size of the World Wide Web. Unless a particular search engine ranks its results by relevance (e.g., by scoring words in a title higher than words in the body of the Web page), the results obtained may not be particularly useful. Also keep in mind that, depending on the indexing scheme that the search engine is using, the found pages may actually no longer exist, leading the user to the dreaded ''404 Not Found'' error.

Compounding this problem is the issue of *coverage*—the number of Web pages that any given search engine is actually able to survey and analyze. A comprehensive study by Lawrence and Giles (1998) indicates that the coverage provided by any of the search engines studied is both small and highly variable. For example, the HotBot engine produced 57.5% coverage of what was estimated to be the size of the ''indexable Web,'' whereas Lycos had only 4.41% coverage, a full order of magnitude less than HotBot. The most important conclusion from this study was that the extent of coverage increased as the number of search engines was increased and the results from those individual searches were combined. Combining the results obtained from the six search engines examined in this study produced coverage approaching 100%.

To address this point, a new class of search engines called *meta-search engines* have been developed. These programs will take the user's query and poll anywhere from 5–10 of the ''traditional'' search engines. The meta-search engine will then

TABLE 1.3. Number of Hits Returned for Four Defined Search Queries on Some of the More Popular Search and Meta-Search Engines

	Search Engine				Meta-Search Engine		
Search Term	HotBot	Excite	Infoseek	Lycos	Google	MetaCrawler	SavvySearch
Genetic mapping	478	1,040	4,326	9,395	7,043	62	58
Human genome	13,213	34,760	15,980	19,536	19,797	42	54
Positional cloning	279	735	1,143	666	3,987	40	52
Prostate cancer	14,044	53,940	24,376	33,538	23,100	0	57

collect the results, filter out duplicates, and return a single, annotated list to the user. One big advantage is that the meta-search engines take relevance statistics into account, returning much smaller lists of results. Although the hit list is substantially smaller, it is much more likely to contain sites that directly address the original query. Because the programs must poll a number of different search engines, searches conducted this way obviously take longer to perform, but the higher degree of confidence in the compiled results for a given query outweighs the extra few minutes (and sometimes only seconds) of search time. Reliable and easy-to-use meta-search engines include MetaCrawler and Savvy Search.

INTERNET RESOURCES FOR TOPICS PRESENTED IN CHAPTER 1

DOMAIN NAMES	
gTLD-MOU	*http://www.gtld-mou.org*
Internet Software Consortium	*http://www.isc.org*
ELECTRONIC MAIL AND NEWSGROUPS	
BIOSCI Newsgroups	*http://www.bio.net/docs/biosci.FAQ.html*
Eudora	*http://www.eudora.com*
Microsoft Exchange	*http://www.microsoft.com/exchange/*
NewsWatcher	*ftp://ftp.acns.nwu.edu/pub/newswatcher/*
FILE TRANSFER PROTOCOL	
Fetch 3.0/Mac	*http://www.dartmouth.edu/pages/softdev/fetch.html*
LeechFTP/PC	*http://stud.fh-heilbronn.de/fdebis/leechftp/*
INTERNET ACCESS	
America Online	*http://www.aol.com*
AT&T	*http://www.att.com/worldnet*
Bell Atlantic	*http://www.verizon.net*
Bell Canada	*http://www.bell.ca*
CompuServe	*http://www.compuserve.com*
Ricochet	*http://www.ricochet.net*
Telus	*http://www.telus.net*
Worldcom	*http://www. worldcom.com*
VIRTUAL LIBRARIES	
EBI BioCatalog	*http://www.ebi.ac.uk/biocat/biocat.html*
Amos' WWW Links Page	*http://www.expasy.ch/alinks.html*
NAR Database Collection	*http://www.nar.oupjournals.org*
WWW Virtual Library	*http://mcb.harvard.edu/BioLinks.html*
WORLD WIDE WEB BROWSERS	
Internet Explorer	*http://explorer.msn.com/home.htm*
Lynx	*ftp://ftp2.cc.ukans.edu/pub/lynx*
Netscape Navigator	*http://home.netscape.com*
WORLD WIDE WEB SEARCH ENGINES	
AltaVista	*http://www.altavista.com*
Excite	*http://www.excite.com*
Google	*http://www.google.com*

HotBot	*http://hotbot.lycos.com*
Infoseek	*http://infoseek.go.com*
Lycos	*http://www.lycos.com*
Northern Light	*http://www.northernlight.com*

WORLD WIDE WEB META-SEARCH ENGINES

MetaCrawler	*http://www.metacrawler.com*
Savvy Search	*http://www.savvysearch.com*

REFERENCES

Conner-Sax, K., and Krol, E. (1999). *The Whole Internet: The Next Generation* (Sebastopol, CA: O'Reilly and Associates).

Froehlich, F., and Kent, A. (1991). ARPANET, the Defense Data Network, and Internet. In *Encyclopedia of Communications* (New York: Marcel Dekker).

Kennedy, A. J. (1999). *The Internet: Rough Guide 2000* (London: Rough Guides).

Lawrence, S., and Giles, C. L. (1998). Searching the World Wide Web. *Science* 280, 98–100.

Quarterman, J. (1990). *The Matrix: Computer Networks and Conferencing Systems Worldwide* (Bedford, MA: Digital Press).

Rankin, B. (1996). *Dr. Bob's Painless Guide to the Internet and Amazing Things You Can Do With E-mail* (San Francisco: No Starch Press).

2

THE NCBI DATA MODEL

James M. Ostell

National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Bethesda, Maryland

Sarah J. Wheelan

Department of Molecular Biology and Genetics
The Johns Hopkins School of Medicine
Baltimore, Maryland

Jonathan A. Kans

National Center for Biotechnology Information
National Library of Medicine
National Institutes of Health
Bethesda, Maryland

INTRODUCTION

Why Use a Data Model?

Most biologists are familiar with the use of animal models to study human diseases. Although a disease that occurs in humans may not be found in exactly the same form in animals, often an animal disease shares enough attributes with a human counterpart to allow data gathered on the animal disease to be used to make inferences about the process in humans. Mathematical models describing the forces involved in musculoskeletal motions can be built by imagining that muscles are combinations of springs and hydraulic pistons and bones are lever arms, and, often times,

such models allow meaningful predictions to be made and tested about the obviously much more complex biological system under consideration. The more closely and elegantly a model follows a real phenomenon, the more useful it is in predicting or understanding the natural phenomenon it is intended to mimic.

In this same vein, some 12 years ago, the National Center for Biotechnology Information (NCBI) introduced a new model for sequence-related information. This new and more powerful model made possible the rapid development of software and the integration of databases that underlie the popular Entrez retrieval system and on which the GenBank database is now built (cf. Chapter 7 for more information on Entrez). The advantages of the model (e.g., the ability to move effortlessly from the published literature to DNA sequences to the proteins they encode, to chromosome maps of the genes, and to the three-dimensional structures of the proteins) have been apparent for years to biologists using Entrez, but very few biologists understand the foundation on which this model is built. As genome information becomes richer and more complex, more of the real, underlying data model is appearing in common representations such as GenBank files. Without going into great detail, this chapter attempts to present a practical guide to the principles of the NCBI data model and its importance to biologists at the bench.

Some Examples of the Model

The GenBank flatfile is a ''DNA-centered'' report, meaning that a region of DNA coding for a protein is represented by a ''CDS feature,'' or ''coding region,'' on the DNA. A *qualifier* (`/translation="MLLYY"`) describes a sequence of amino acids produced by translating the CDS. A limited set of additional *features* of the DNA, such as `mat_peptide`, are occasionally used in GenBank flatfiles to describe cleavage products of the (possibly unnamed) protein that is described by a `/translation`, but clearly this is not a satisfactory solution. Conversely, most protein sequence databases present a ''protein-centered'' view in which the connection to the encoding gene may be completely lost or may be only indirectly referenced by an accession number. Often times, these connections do not provide the exact codon-to-amino acid correspondences that are important in performing mutation analysis.

The NCBI data model deals directly with the two sequences involved: a DNA sequence and a protein sequence. The translation process is represented as a link between the two sequences rather than an annotation on one with respect to the other. Protein-related annotations, such as peptide cleavage products, are represented as features annotated directly on the protein sequence. In this way, it becomes very natural to analyze the protein sequences derived from translations of CDS features by BLAST or any other sequence search tool without losing the precise linkage back to the gene. A collection of a DNA sequence and its translation products is called a *Nuc-prot set*, and this is how such data is represented by NCBI. The GenBank flatfile format that many readers are already accustomed to is simply a particular style of report, one that is more ''human-readable'' and that ultimately flattens the connected collection of sequences back into the familiar one-sequence, DNA-centered view. The navigation provided by tools such as Entrez much more directly reflects the underlying structure of such data. The protein sequences derived from GenBank translations that are returned by BLAST searches are, in fact, the protein sequences from the Nuc-prot sets described above.

The standard GenBank format can also hide the multiple-sequence nature of some DNA sequences. For example, three genomic exons of a particular gene are sequenced, and partial flanking, noncoding regions around the exons may also be available, but the full-length sequences of these intronic sequences may not yet be available. Because the exons are not in their complete genomic context, there would be three GenBank flatfiles in this case, one for each exon. There is no explicit representation of the complete set of sequences over that genomic region; these three exons come in genomic order and are separated by a certain length of unsequenced DNA. In GenBank format there would be a Segment line of the form `SEGMENT 1 of 3` in the first record, `SEGMENT 2 of 3` in the second, and `SEGMENT 3 of 3` in the third, but this only tells the user that the lines are part of some undefined, ordered series (Fig. 2.1*A*). Out of the whole GenBank release, one locates the correct Segment records to place together by an algorithm involving the `LOCUS` name. All segments that go together use the same first combination of letters, ending with the numbers appropriate to the segment, e.g., `HSDDT1`, `HSDDT2`, and `HSDDT3`. Obviously, this complicated arrangement can result in problems when `LOCUS` names include numbers that inadvertently interfere with such series. In addition, there is no one sequence record that describes the whole assembled series, and there is no way to describe the distance between the individual pieces. There is no segmenting convention in the EMBL sequence database at all, so records derived from that source or distributed in that format lack even this imperfect information.

The NCBI data model defines a sequence type that directly represents such a segmented series, called a ''segmented sequence.'' Rather than containing the letters A, G, C, and T, the segmented sequence contains instructions on how it can be built from other sequences. Considering again the example above, the segmented sequence would contain the instructions ''take all of `HSDDT1`, then a gap of unknown length, then all of `HSDDT2`, then a gap of unknown length, then all of `HSDDT3`.'' The segmented sequence itself can have a name (e.g., `HSDDT`), an accession number, features, citations, and comments, like any other GenBank record. Data of this type are commonly stored in a so-called ''Seg-set'' containing the sequences `HSDDT`, `HSDDT1`, `HSDDT2`, `HSDDT3` and all of their connections and features. When the GenBank release is made, as in the case of Nuc-prot sets, the Seg-sets are broken up into multiple records, and the segmented sequence itself is not visible. However, GenBank, EMBL, and DDBJ have recently agreed on a way to represent these constructed assemblies, and they will be placed in a new CON division, with CON standing for ''contig'' (Fig. 2.1*B*). In the Entrez graphical view of segmented sequences, the segmented sequence is shown as a line connecting all of its component sequences (Fig. 2.1*C*).

An NCBI segmented sequence does not require that there be gaps between the individual pieces. In fact the pieces can overlap, unlike the case of a segmented series in GenBank format. This makes the segmented sequence ideal for representing large sequences such as bacterial genomes, which may be many megabases in length. This is what currently is done within the Entrez Genomes division for bacterial genomes, as well as other complete chromosomes such as yeast. The NCBI Software Toolkit (Ostell, 1996) contains functions that can gather the data that a segmented sequence refers to ''on the fly,'' including constituent sequence and features, and this information can automatically be remapped from the coordinates of a small, individual record to that of a complete chromosome. This makes it possible to provide graphical views, GenBank flatfile views, or FASTA views or to perform analyses on

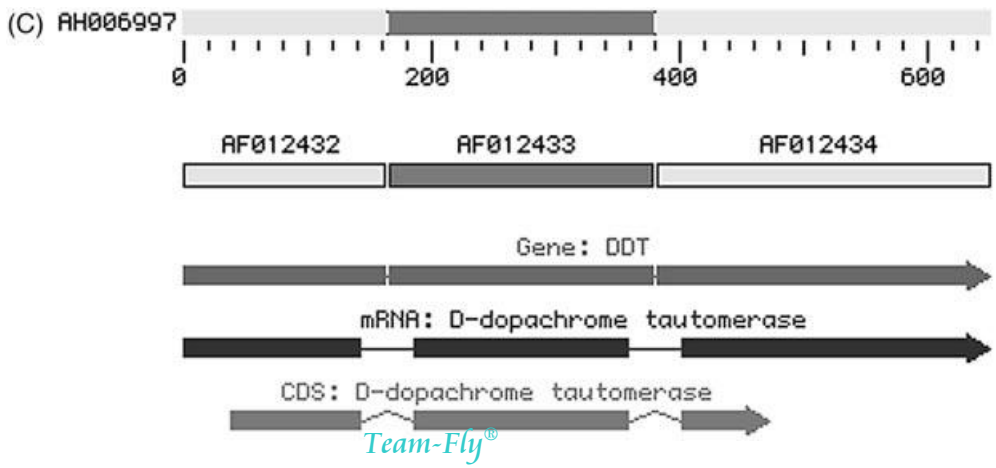
(A)

```
LOCUS       HSDDT1        166 bp    DNA             PRI       01-FEB-2000
DEFINITION  Homo sapiens D-dopachrome tautomerase (DDT) gene, exon 1.
ACCESSION   AF012432
VERSION     AF012432.1  GI:2352911
KEYWORDS    .
SEGMENT     1 of 3
....
LOCUS       HSDDT2        216 bp    DNA             PRI       01-FEB-2000
DEFINITION  Homo sapiens D-dopachrome tautomerase (DDT) gene, exon 2.
ACCESSION   AF012433
VERSION     AF012433.1  GI:2352912
KEYWORDS    .
SEGMENT     2 of 3
....
LOCUS       HSDDT3        271 bp    DNA             PRI       01-FEB-2000
DEFINITION  Homo sapiens D-dopachrome tautomerase (DDT) gene, exon 3 and
            complete cds.
ACCESSION   AF012434
VERSION     AF012434.1  GI:2352913
KEYWORDS    .
SEGMENT     3 of 3
....
```

(B)

```
LOCUS       HSDDT         653 bp    DNA             CON       01-FEB-2000
DEFINITION  Homo sapiens D-dopachrome tautomerase (DDT) gene, complete cds.
ACCESSION   AH006997
VERSION     AH006997.2  GI:6849043
KEYWORDS    .
SOURCE      human.
  ORGANISM  Homo sapiens
            Eukaryota; Metazoa; Chordata; Craniata; Vertebrata; Mammalia;
            Eutheria; Primates; Catarrhini; Hominidae; Homo.
REFERENCE   1  (bases 1 to 653)
  AUTHORS   Esumi,N., Budarf,M., Ciccarelli,L., Sellinger,B., Kozak,C.A.
            and Wistow,G.
  TITLE     Conserved gene structure and genomic linkage for D-dopachrome
            tautomerase (DDT) and MIF
  JOURNAL   Mamm. Genome 9 (9), 753-757 (1998)
  MEDLINE   98384542
   PUBMED   9716662
REFERENCE   2  (bases 1 to 653)
  AUTHORS   Esumi,N. and Wistow,G.
  TITLE     Direct Submission
  JOURNAL   Submitted (07-JUL-1997) Molecular Structure and Function, NEI,
            Building 6, Rm. 331, NIH, Bethesda, MD 20892, USA
COMMENT     On Feb 1, 2000 this sequence version replaced gi:2352914.
FEATURES             Location/Qualifiers
     source          1..653
                     /organism="Homo sapiens"
                     /db_xref="taxon:9606"
                     /chromosome="22"
CONTIG      join(AF012432.1:1..166,gap(),AF012433.1:1..216,gap(),
            AF012434.1:1..271)
//
```

(C)



whole chromosomes quite easily, even though data exist only in small, individual pieces. This ability to readily assemble a set of related sequences on demand for any region of a very large chromosome has already proven to be valuable for bacterial genomes. Assembly on demand will become more and more important as larger and larger regions are sequenced, perhaps by many different groups, and the notion that an investigator will be working on one huge sequence record becomes completely impractical.

What Does ASN.1 Have to Do With It?

The NCBI data model is often referred to as, and confused with, the ''NCBI ASN.1'' or ''ASN.1 Data Model.'' *Abstract Syntax Notation 1* (ASN.1) is an International Standards Organization (ISO) standard for describing structured data that reliably encodes data in a way that permits computers and software systems of all types to reliably exchange both the structure and the content of the entries. Saying that a data model is written in ASN.1 is like saying a computer program is written in C or FORTRAN. The statement identifies the *language*; it does not say what the program *does*. The familiar GenBank flatfile was really designed for humans to read, from a DNA-centered viewpoint. ASN.1 is designed for a *computer* to read and is amenable to describing complicated data relationships in a very specific way. NCBI describes and processes data using the ASN.1 format. Based on that single, common format, a number of human-readable formats and tools are produced, such as Entrez, GenBank, and the BLAST databases. Without the existence of a common format such as this, the neighboring and hard-link relationships that Entrez depends on would not be possible. This chapter deals with the structure and content of the NCBI data model and its implications for biomedical databases and tools. Detailed discussions about the choice of ASN.1 for this task and its overall form can be found elsewhere (Ostell, 1995).

What to Define?

We have alluded to how the NCBI data model defines sequences in a way that supports a richer and more explicit description of the experimental data than can be

Figure 2.1. (*A*) Selected parts of GenBank-formatted records in a segmented sequence. GenBank format historically indicates merely that records are part of some ordered series; it offers no information on what the other components are or how they are connected. To see the complete view of these records, see *http://www.ncbi.nlm.nih.gov/htbin-post/Entrez/query?uid=6849043&form=6&db=n&Dopt=g*. (*B*) Representation of segmented sequences in the new CON (contig) division. A new extension of GenBank format allows the details of the construction of segmented records to be presented. The `CONTIG` line can include individual accessions, gaps of known length, and gaps of unknown length. The individual components can still be displayed in the traditional form, although no features or sequences are present in this format. (*C*) Graphical representation of a segmented sequence. This view displays features mapped to the coordinates of the segmented sequence. The segments include all exonic and untranslated regions plus 20 base pairs of sequence at the ends of each intron. The segment gaps cover the remaining intronic sequence.

obtained with the GenBank format. The details of the model are important, and will be expanded on in the ensuing discussion. At this point, we need to pause and briefly describe the reasoning and general principles behind the model as a whole.

There are two main reasons for putting data on a computer: retrieval and discovery. Retrieval is basically being able to get back out what was put in. Amassing sequence information without providing a way to retrieve it makes the sequence information, in essence, useless. Although this is important, it is even *more* valuable to be able to get back from the system *more* knowledge than was put in to begin with—that is, to be able to use the information to make biological discoveries. Scientists can make these kinds of discoveries by discerning connections between two pieces of information that were not known when the pieces were entered separately into the database or by performing computations on the data that offer new insight into the records. In the NCBI data model, the emphasis is on facilitating discovery; that means the data must be defined in a way that is amenable to both linkage and computation.

A second, general consideration for the model is stability. NCBI is a US Government agency, not a group supported year-to-year by competitive grants. Thus, the NCBI staff takes a very long-term view of its role in supporting bioinformatics efforts. NCBI provides large-scale information systems that will support scientific inquiry well into the future. As anyone who is involved in biomedical research knows, many major conceptual and technical revolutions can happen when dealing with such a long time span. Somehow, NCBI must address these changing views and needs with software and data that may have been created years (or decades) earlier. For that reason, basic observations have been chosen as the central data elements, with interpretations and nomenclature (elements more subject to change) being placed outside the basic, core representation of the data.

Taking all factors into account, NCBI uses four core data elements: bibliographic citations, DNA sequences, protein sequences, and three-dimensional structures. In addition, two projects (taxonomy and genome maps) are more interpretive but nonetheless are so important as organizing and linking resources that NCBI has built a considerable base in these areas as well.

PUBs: PUBLICATIONS OR PERISH

Publication is at the core of every scientific endeavor. It is the common process whereby scientific information is reviewed, evaluated, distributed, and entered into the permanent record of scientific progress. Publications serve as vital links between factual databases of different structures or content domains (e.g., a record in a sequence database and a record in a genetic database may cite the same article). They serve as valuable entry points into factual databases (''I have read an article about this, now I want to see the primary data'').

Publications also act as essential annotation of function and context to records in factual databases. One reason for this is that factual databases have a structure that is essential for efficient use of the database but may not have the representational capacity to set forward the full biological, experimental, or historical context of a particular record. In contrast, the published paper is limited only by language and contains much fuller and more detailed explanatory information than will ever be in a record in a factual database. Perhaps more importantly, authors are evaluated by

their scientific peers based on the content of their published papers, not by the content of the associated database records. Despite the best of intentions, scientists move on and database records become static, even though the knowledge about them has expanded, and there is very little incentive for busy scientists to learn a database system and keep records based on their own laboratory studies up to date.

Generally, the form and content of citations have not been thought about carefully by those designing factual databases, and the quality, form, and content of citations can vary widely from one database to the next. Awareness of the importance of having a link to the published literature and the realization that bibliographic citations are much less volatile than scientific knowledge led to a decision that a careful and complete job of defining citations was a worthwhile endeavor. Some components of the publication specification described below may be of particular interest to scientists or users of the NCBI databases, but a full discussion of all the issues leading to the decisions governing the specifications themselves would require another chapter in itself.

Authors

Author names are represented in many formats by various databases: last name only, last name and initials, last name-comma-initials, last name and first name, all authors with initials and the last with a full first name, with or without honorifics (Ph.D.) or suffixes (Jr., III), to name only a few. Some bibliographic databases (such as MEDLINE) might represent only a fixed number of authors. Although this inconsistency is merely ugly to a human reader, it poses *severe* problems for database systems incorporating names from many sources and providing functions as simple as looking up citations by author last name, such as Entrez does. For this reason, the specification provides two alternative forms of author name representation: one a simple string and the other a structured form with fields for last name, first name, and so on. When data are submitted directly to NCBI or in cases when there is a consistent format of author names from a particular source (such as MEDLINE), the structured form is used. When the form cannot be deciphered, the author name remains as a string. This limits its use for retrieval but at least allows data to be viewed when the record is retrieved by other means.

Even the structured form of author names must support diversity, since some sources give only initials whereas others provide a first and middle name. This is mentioned to specifically emphasize two points. First, the NCBI data model is designed both to direct our view of the data into a more useful form and to accommodate the available existing data. (This pair of functions can be confusing to people reading the specification and seeing alternative forms of the same data defined.) Second, software developers must be aware of this range of representations and accommodate whatever form had to be used when a particular source was being converted. In general, NCBI tries to get as much of the data into a uniform, structured form as possible but carries the rest in a less optimal way rather than losing it altogether.

Author affiliations (i.e., authors' institutional addresses) are even more complicated. As with author names, there is the problem of supporting both structured forms and unparsed strings. However, even sources with reasonably consistent author name conventions often produce affiliation information that cannot be parsed from text into a structured format. In addition, there may be an affiliation associated with the whole

author list, or there may be different affiliations associated with each author. The NCBI data model allows for both scenarios. At the time of this writing only the first form is supported in either MEDLINE or GenBank, both types may appear in published articles.

Articles

The most commonly cited bibliographic entity in biological science is an article in a journal; therefore, the citation formats of most biological databases are defined with that type in mind. However, ''articles'' can also appear in books, manuscripts, theses, and now in electronic journals as well. The data model defines the fields necessary to cite a book, a journal, or a manuscript. An article citation occupies one field; other fields display additional information necessary to uniquely identify the article in the book, journal, or manuscript—the author(s) of the article (as opposed to the author or editor of the book), the title of the article, page numbers, and so on.

There is an important distinction between the fields necessary to uniquely identify a published article from a citation and those necessary to describe the same article meaningfully to a database user. The NCBI Citation Matching Service takes fields from a citation and attempts to locate the article to which they refer. In this process, a successful match would involve only correctly matching the journal title, the year, the first page of the article, and the last name of an author of the article. Other information (e.g., article title, volume, issue, full pages, author list) is useful to look at but very often is either not available or outright incorrect. Once again, the data model must allow the minimum information set to come in as a citation, be matched against MEDLINE, and then be replaced by a citation having the full set of desired fields obtained from MEDLINE to produce accurate, useful data for consumption by the scientific public.

Patents

With the advent of patented sequences it became necessary to cite a patent as a bibliographic entity instead of an article. The data model supports a very complete patent citation, a format developed in cooperation with the US Patent Office. In practice, however, patented sequences tend to have limited value to the scientific public. Because a patent is a *legal* document, not a scientific one, its purpose is to present and support the claims of the patent, *not* to fully describe the biology of the sequence itself. It is often prepared in a lawyer's office, not by the scientist who did the research. The sequences presented in the patent may function only to illustrate some discreet aspect of the patent, rather than being the focus of the document. Organism information, location of biological features, and so on may not appear at all if they are not germane to the patent. Thus far, the vast majority of sequences appearing in patents also appear in a more useful form (to scientists) in the public databases.

In NCBI's view, the main purpose of listing patented sequences in GenBank is to be able to retrieve sequences by similarity searches that may serve to locate patents related to a given sequence. To make a legal determination in the case, however, one would still have to examine the full text of the patent. To evaluate the biology of the sequence, one generally must locate information other than that contained in the patent. Thus, the critical linkage is between the sequence and its patent number.

Additional fields in the patent citation itself may be of some interest, such as the title of the patent and the names of the inventors.

Citing Electronic Data Submission

A relatively new class of citations comprises the act of data submission to a database, such as GenBank. This is an act of publication, similar but not identical to the publication of an article in a journal. In some cases, data submission precedes article publication by a considerable period of time, or a publication regarding a particular sequence may never appear in press. Because of this, there is a separate citation designed for deposited sequence data. The submission citation, because it is indeed an act of publication, may have an author list, showing the names of scientists who worked on the record. This may or may not be the same as the author list on a subsequently published paper also cited in the same record. In most cases, the scientist who submitted the data to the database is also an author on the submission citation. (In the case of large sequencing centers, this may not always be the case.) Finally, NCBI has begun the practice of citing the update of a record with a submission citation as well. A comment can be included with the update, briefly describing the changes made in the record. All the submission citations can be retained in the record, providing a history of the record over time.

MEDLINE and PubMed Identifiers

Once an article citation has been matched to MEDLINE, the simplest and most reliable key to point to the article is the MEDLINE unique identifier (MUID). This is simply an integer number. NCBI provides many services that use MUID to retrieve the citation and abstract from MEDLINE, to link together data citing the same article, or to provide Web hyperlinks.

Recently, in concert with MEDLINE and a large number of publishers, NCBI has introduced *PubMed*. PubMed contains *all* of MEDLINE, as well as citations provided directly by the publishers. As such, PubMed contains more recent articles than MEDLINE, as well as articles that may never appear in MEDLINE because of their subject matter. This development led NCBI to introduce a new article identifier, called a PubMed identifier (PMID). Articles appearing in MEDLINE will have *both* a PMID and an MUID. Articles appearing only in PubMed will have only a PMID. PMID serves the same purpose as MUID in providing a simple, reliable link to the citation, a means of linking records together, and a means of setting up hyperlinks.

Publishers have also started to send information on ahead-of-print articles to PubMed, so this information may now appear before the printed journal. A new project, *PubMed Central*, is meant to allow electronic publication to occur in lieu of or ahead of publication in a traditional, printed journal. PubMed Central records contain the full text of the article, not just the abstract, and include all figures and references.

The NCBI data model stores most citations as a collection called a Pub-equiv, a set of equivalent citations that includes a reliable identifier (PMID or MUID) and the citation itself. The presence of the citation form allows a useful display without an extra retrieval from the database, whereas the identifier provides a reliable key for linking or indexing the same citation in the record.

SEQ-IDs: WHAT'S IN A NAME?

The NCBI data model defines a whole class of objects called Sequence Identifiers (Seq-id). There has to be a whole class of such objects because NCBI integrates sequence data from many sources that name sequence records in different ways and where, of course, the individual names have different meanings. In one simple case, PIR, SWISS-PROT, and the nucleotide sequence databases all use a string called an ''accession number,'' all having a similar format. Just saying ''A10234'' is not enough to uniquely identify a sequence record from the collection of all these databases. One must distinguish ''A10234'' in SWISS-PROT from ''A10234'' in PIR. (The DDBJ/EMBL/GenBank nucleotide databases share a common set of accession numbers; therefore, ''A12345'' in EMBL is the same as ''A12345'' in GenBank or DDBJ.) To further complicate matters, although the sequence databases define their records as containing a single sequence, PDB records contain a single *structure*, which may contain more than one sequence. Because of this, a PDB Seq-id contains a molecule name and a chain ID to identify a single unique sequence. The subsections that follow describe the form and use of a few commonly used types of Seq-ids.

Locus Name

The *locus* appears on the `LOCUS` line in GenBank and DDBJ records and in the `ID` line in EMBL records. These originally were the only identifier of a discrete GenBank record. Like a genetic locus name, it was intended to act both as a unique identifier for the record and as a mnemonic for the function and source organism of the sequence. Because the `LOCUS` line is in a fixed format, the locus name is restricted to ten or fewer numbers and uppercase letters. For many years in GenBank, the first three letters of the name were an organism code and the remaining letters a code for the gene (e.g., `HUMHBB` was used for ''human β-globin region''). However, as with genetic locus names, locus names were changed when the function of a region was discovered to be different from what was originally thought. This instability in locus names is obviously a problem for an identifier for retrieval. In addition, as the number of sequences and organisms represented in GenBank increased geometrically over the years, it became impossible to invent and update such mnemonic names in an efficient and timely manner. At this point, the locus name is dying out as a useful name in GenBank, although it continues to appear prominently on the first line of the flatfile to avoid breaking the established format.

Accession Number

Because of the difficulties in using the locus/ID name as the unique identifier for a nucleotide sequence record, the International Nucleotide Sequence Database Collaborators (DDBJ/EMBL/GenBank) introduced the accession number. It intentionally carries no biological meaning, to ensure that it will remain (relatively) stable. It originally consisted of one uppercase letter followed by five digits. New accessions consist of two uppercase letters followed by six digits. The first letters were allocated to the individual collaborating databases so that accession numbers would be unique across the Collaboration (e.g., an entry beginning with a ''U'' was from GenBank).

The accession number was an improvement over the locus/ID name, but, with use, problems and deficiencies became apparent. For example, although the accession

is stable over time, many users noticed that the sequence retrieved by a particular accession was not always the same. This is because the accession identifies the *whole database record*. If the sequence in a record was updated (say by the insertion of 1000 bp at the beginning), the accession number did not change, as it was an updated version of the same record. If one had analyzed the original sequence and recorded that at position 100 of accession U00001 there was a putative protein-binding site, after the update a completely different sequence would be found at position 100!

The accession number appears on the ACCESSION line of the GenBank record. The first accession on the line, called the ''primary'' accession, is the key for retrieving this record. Most records have only this type of accession number. However, other accessions may follow the primary accession on the ACCESSION line. These ''secondary'' accessions are intended to give some notion of the history of the record. For example, if U00001 and U00002 were merged into a single updated record, then U00001 would be the primary accession on the new record and U00002 would appear as a secondary accession. In standard practice, the U00002 record would be removed from GenBank, since the older record had become obsolete, and the secondary accessions would allow users to retrieve whatever records superseded the old one. It should also be noted that, historically, secondary accession numbers do not always mean the same thing; therefore, users should exercise care in their interpretations. (Policies at individual databases differed, and even shifted over time in a given database.) The use of secondary accession numbers also caused problems in that there was still not enough information to determine exactly what happened and why. Nonetheless, the accession number remains the most controlled and reliable way to point to a record in DDBJ/EMBL/GenBank.

gi Number

In 1992, NCBI began assigning GenInfo Identifiers (gi) to all sequences processed into Entrez, including nucleotide sequences from DDBJ/EMBL/GenBank, the protein sequences from the translated CDS features, protein sequences from SWISS-PROT, PIR, PRF, PDB, patents, and others. The gi is assigned in addition to the accession number provided by the source database. Although the form and meaning of the accession Seq-id varied depending on the source, the meaning and form of the gi is the same for all sequences regardless of the source.

The gi is simply an integer number, sometimes referred to as a *GI number*. It is an identifier *for a particular sequence only*. Suppose a sequence enters GenBank and is given an accession number U00001. When the sequence is processed internally at NCBI, it enters a database called ID. ID determines that it has not seen U00001 before and assigns it a gi number—for example, 54. Later, the submitter might update the record by changing the citation, so U00001 enters ID again. ID, recognizing the record, retrieves the first U00001 and compares its sequence with the new one. If the two are completely identical, ID reassigns gi 54 to the record. If the sequence differs in any way, even by a single base pair, it is given a new gi number, say 88. However, the new sequence retains accession number U00001 because of the semantics of the source database. At this time, ID marks the old record (gi 54) with the date it was replaced and adds a ''history'' indicating that it was replaced by gi 88. ID also adds a history to gi 88 indicating that it replaced gi 54.

The gi number serves three major purposes:

- It provides a single identifier across sequences from many sources.
- It provides an identifier that specifies an exact sequence. Anyone who analyzes gi 54 and stores the analysis can be sure that it will be valid as long as U00001 has gi 54 attached to it.
- It is stable and retrievable. NCBI keeps the last version of every gi number. Because the history is included in the record, anyone who discovers that gi 54 is no longer part of the GenBank release can still retrieve it from ID through NCBI and examine the history to see that it was replaced by gi 88. Upon aligning gi 54 to gi 88 to determine their relationship, a researcher may decide to remap the former analysis to gi 88 or perhaps to reanalyze the data. This can be done at any time, not just at GenBank release time, because gi 54 will always be available from ID.

For these reasons, all internal processing of sequences at NCBI, from computing Entrez sequence neighbors to determining when new sequence should be processed or producing the BLAST databases, is based on gi numbers.

Accession.Version Combined Identifier

Recently, the members of the International Nucleotide Sequence Database Collaboration (GenBank, EMBL, and DDBJ) introduced a ''better'' sequence identifier, one that combines an accession (which identifies a particular sequence record) with a version number (which tracks changes to the sequence itself). It is expected that this kind of Seq-id will become the preferred method of citing sequences.

Users will still be able to retrieve a record based on the accession number alone, without having to specify a particular version. In that case, the latest version of the record will be obtained by default, which is the current behavior for queries using Entrez and other retrieval programs.

Scientists who are analyzing sequences in the database (e.g., aligning all alcohol dehydrogenase sequences from a particular taxonomic group) and wish to have their conclusions remain valid over time will want to reference sequences by accession and the given version number. Subsequent modification of one of the sequences by its owner (e.g., 5′ extension during a study of the gene's regulation) will result in the version number being incremented appropriately. The analysis that cited accession and version remains valid because a query using both the accession and version will return the desired record.

Combining accession and version makes it clear to the casual user that a sequence has changed since an analysis was done. Also, determining how many times a sequence has changed becomes trivial with a version number. The accession.version number appears on the `VERSION` line of the GenBank flatfile. For sequence retrieval, the accession.version is simply mapped to the appropriate gi number, which remains the underlying tracking identifier at NCBI.

Accession Numbers on Protein Sequences

The International Sequence Database Collaborators also started assigning accession.version numbers to *protein* sequences within the records. Previously, it was difficult to reliably cite the translated product of a given coding region feature, except

by its gi number. This limited the usefulness of translated products found in BLAST results, for example. These sequences will now have the same status as protein sequences submitted directly to the protein databases, and they have the benefit of direct linkage to the nucleotide sequence in which they are encoded, showing up as a CDS feature's `/protein_id` qualifier in the flatfile view. Protein accessions in these records consist of three uppercase letters followed by five digits and an integer indicating the version.

Reference Seq-id

The NCBI RefSeq project provides a curated, nonredundant set of reference sequence standards for naturally occurring biological molecules, ranging from chromosomes to transcripts to proteins. RefSeq identifiers are in accession.version form but are prefixed with `NC_` (chromosomes), `NM_` (mRNAs), `NP_` (proteins), or `NT_` (constructed genomic contigs). The `NG_` prefix will be used for genomic regions or gene clusters (e.g., immunoglobulin region) in the future. RefSeq records are a stable reference point for functional annotation, point mutation analysis, gene expression studies, and polymorphism discovery.

General Seq-id

The General Seq-id is meant to be used by genome centers and other groups as a way of identifying their sequences. Some of these sequences may never appear in public databases, and others may be preliminary data that eventually will be submitted. For example, records of human chromosomes in the Entrez Genomes division contain multiple physical and genetic maps, in addition to sequence components. The physical maps are generated by various groups, and they use General Seq-ids to identify the proper group.

Local Seq-id

The Local sequence identifier is most prominently used in the data submission tool Sequin (see Chapter 4). Each sequence will eventually get an `accession.version` identifier and a gi number, but only when the completed submission has been processed by one of the public databases. During the submission process, Sequin assigns a local identifier to each sequence. Because many of the software tools made by NCBI require a sequence identifier, having a local Seq-id allows the use of these tools without having to first submit data to a public database.

BIOSEQs: SEQUENCES

The Bioseq, or biological sequence, is a central element in the NCBI data model. It comprises a single, continuous molecule of either nucleic acid or protein, thereby defining a linear, integer coordinate system for the sequence. A Bioseq must have at least one sequence identifier (Seq-id). It has information on the physical type of molecule (DNA, RNA, or protein). It may also have annotations, such as biological features referring to specific locations on specific Bioseqs, as well as descriptors.

Descriptors provide additional information, such as the organism from which the molecule was obtained. Information in the descriptors describe the entire Bioseq.

However, the Bioseq isn't necessarily a fully sequenced molecule. It may be a segmented sequence in which, for example, the exons have been sequenced but not all of the intronic sequences have been determined. It could also be a genetic or physical map, where only a few landmarks have been positioned.

Sequences are the Same

All Bioseqs have an integer coordinate system, with an integer length value, even if the actual sequence has not been completely determined. Thus, for physical maps, or for exons in highly spliced genes, the spacing between markers or exons may be known only from a band on a gel. Although the coordinates of a fully sequenced chromosome are known exactly, those in a genetic or physical map are a best guess, with the possibility of significant error from the ''real'' coordinates.

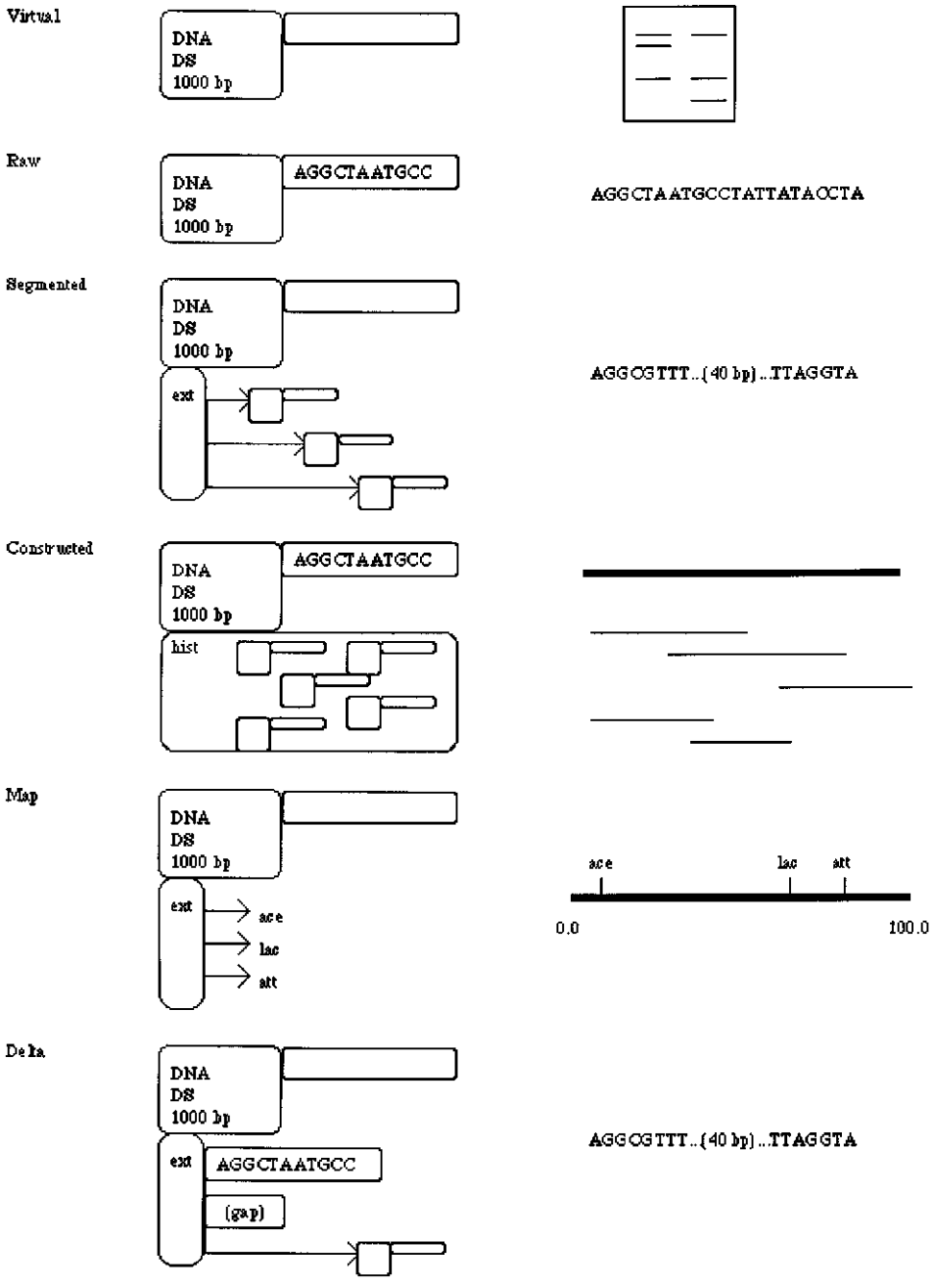
Nevertheless, any Bioseq can be annotated with the same kinds of information. For example, a gene feature can be placed on a region of sequenced DNA or at a discrete location on a physical map. The map and the sequence can then be aligned on the basis of their common gene features. This greatly simplifies the task of writing software that can display these seemingly disparate kinds of data.

Sequences are Different

Despite the benefits derived from having a common coordinate system, the different Bioseq classes do differ in the way they are represented. The most common classes (Fig. 2.2) are described briefly below.

Virtual Bioseq. In the virtual Bioseq, the molecule type is known, and its length and topology (e.g., linear, circular) may also be known, but the actual sequence is not known. A virtual Bioseq can represent an intron in a genomic molecule in which only the exon sequences have been determined. The length of the putative sequence may be known only by the size of a band on an agarose gel.

Figure 2.2. Classes of Bioseqs. All Bioseqs represent a single, continuous molecule of nucleic acid or protein, although the complete sequence may not be known. In a virtual Bioseq, the type of molecule is known, but the sequence is not known, and the precise length may not be known (e.g., from the size of a band on an electrophoresis gel). A raw Bioseq contains a single contiguous string of bases or residues. A segmented Bioseq points to its components, which are other raw or virtual Bioseqs (e.g., sequenced exons and undetermined introns). A constructed sequence takes its original components and subsumes them, resulting in a Bioseq that contains the string of bases or residues and a "history" of how it was built. A map Bioseq places genes or physical markers, rather than sequence, on its coordinates. A delta Bioseq can represent a segmented sequence but without the requirement of assigning identifiers to each component (including gaps of known length), although separate raw sequences can still be referenced as components. The delta sequence is used for unfinished high-throughput genome sequences (HTGS) from genome centers and for genomic contigs.


Virtual
DNA
DS
1000 bp
Raw
AGGCTAATGCC
AGGCTAATGCCTATTATACCTA
Segmented
ext
AGGCGTTT..(40 bp)..TTAGGTA
Constructed
hist
Map
ace
lac
att
0.0
100.0
Delta
(gap)
AGGCGTTT..(40 bp)..TTAGGTA

Raw Bioseq. This is what most people would think of as a sequence, a single contiguous string of bases or residues, in which the actual sequence is known. The length is obviously known in this case, matching the number of bases or residues in the sequence.

Segmented Bioseq. A segmented Bioseq does not contain raw sequences but instead contains the identifiers of other Bioseqs from which it is made. This type of Bioseq can be used to represent a genomic sequence in which only the exons are known. The ''parts'' in the segmented Bioseq would be the individual, raw Bioseqs representing the exons and the virtual Bioseqs representing the introns.

Delta Bioseq. Delta Bioseqs are used to represent the unfinished high-throughput genome sequences (HTGS) derived at the various genome sequencing centers. Using delta Bioseqs instead of segmented Bioseqs means that only one Seq-id is needed for the entire sequence, even though subregions of the Bioseq are not known at the sequence level. Implicitly, then, even at the early stages of their presence in the databases, delta Bioseqs maintain the same accession number.

Map Bioseq. Used to represent genetic and physical maps, a map Bioseq is similar to a virtual Bioseq in that it has a molecule type, perhaps a topology, and a length that may be a very rough estimate of the molecule's actual length. This information merely supplies the coordinate system, a property of every Bioseq. Given this coordinate system for a genetic map, we estimate the positions of genes on it based on genetic evidence. The table of the resulting gene features is the essential data of the map Bioseq, just as bases or residues constitute the raw Bioseq's data.

BIOSEQ-SETs: COLLECTIONS OF SEQUENCES

A biological sequence is often most appropriately stored in the context of other, related sequences. For example, a nucleotide sequence and the sequences of the protein products it encodes naturally belong in a set. The NCBI data model provides the Bioseq-set for this purpose.

A Bioseq-set can have a list of *descriptors*. When packaged on a Bioseq, a descriptor applies to all of that Bioseq. When packaged on a Bioseq-set, the descriptor applies to every Bioseq in the set. This arrangement is convenient for attaching publications and biological source information, which are expected on all sequences but frequently are identical within sets of sequences. For example, both the DNA and protein sequences are obviously from the same organism, so this descriptor information can be applied to the set. The same logic may apply to a publication.

The most common Bioseq-sets are described in the sections that follow.

Nucleotide/Protein Sets

The Nuc-prot set, containing a nucleotide and one or more protein products, is the type of set most frequently produced by a Sequin data submission. The component Bioseqs are connected by coding sequence region (CDS) features that describe how translation from nucleotide to protein sequence is to proceed. In a traditional nucleotide or protein sequence database, these records might have cross-references to each

other to indicate this relationship. The Nuc-prot set makes this explicit by packaging them together. It also allows descriptive information that applies to all sequences (e.g., the organism or publication citation) to be entered once (see *Seq-descr: Describing the Sequence*, below).

Population and Phylogenetic Studies

A major class of sequence submissions represent the results of population or phylogenetic studies. Such research involves sequencing the same gene from a number of individuals in the same species (population study) or in different species (phylogenetic study). An alignment of the individual sequences may also be submitted (see *Seq-align: Alignments*, below). If the gene encodes a protein, the components of the Population or Phylogenetic Bioseq-set may themselves be Nuc-prot sets.

Other Bioseq-sets

A Seg set contains a segmented Bioseq and a Parts Bioseq-set, which in turn contains the raw Bioseqs that are referenced by the segmented Bioseq. This may constitute the nucleotide component of a Nuc-prot set.

An Equiv Bioseq-set is used in the Entrez Genomes division to hold multiple equivalent Bioseqs. For example, human chromosomes have one or more genetic maps, physical maps derived by different methods and a segmented Bioseq on which ''islands'' of sequenced regions are placed. An alignment between the various Bioseqs is made based on references to any available common markers.

SEQ-ANNOT: ANNOTATING THE SEQUENCE

A Seq-annot is a self-contained package of sequence annotations or information that refers to specific locations on specific Bioseqs. It may contain a feature table, a set of sequence alignments, or a set of graphs of attributes along the sequence.

Multiple Seq-annots can be placed on a Bioseq or on a Bioseq-set. Each Seq-annot can have specific attribution. For example, PowerBLAST (Zhang and Madden, 1997) produces a Seq-annot containing sequence alignments, and each Seq-annot is named based on the BLAST program used (e.g., BLASTN, BLASTX, etc.). The individual blocks of alignments are visible in the Entrez and Sequin viewers.

Because the components of a Seq-annot have specific references to locations on Bioseqs, the Seq-annot can stand alone or be exchanged with other scientists, and it need not reside in a sequence record. The scope of descriptors, on the other hand, does depend on where they are packaged. Thus, information *about* Bioseqs can be created, exchanged, and compared independently of the Bioseq itself. This is an important attribute of the Seq-annot and of the NCBI data model.

Seq-feat: Features

A sequence feature (Seq-feat) is a block of structured data explicitly attached to a region of a Bioseq through one or two sequence locations (Seq-locs). The Seq-feat itself can carry information common to all features. For example, there are flags to indicate whether a feature is partial (i.e., goes beyond the end of the sequence of

the Bioseq), whether there is a biological exception (e.g., RNA editing that explains why a codon on the genomic sequence does not translate to the expected amino acid), and whether the feature was experimentally determined (e.g., an mRNA was isolated from a proposed coding region).

A feature must always have a location. This is the Seq-loc that states where on the sequence the feature resides. A coding region's location usually starts at the ATG and ends at the terminator codon. The location can have more than one interval if it is on a genomic sequence and mRNA splicing occurs. In cases of alternative splicing, separate coding region features are created, with one multi-interval Seq-loc for each isolated molecular species.

Optionally, a feature may have a product. For a coding region, the product Seq-loc points to the resulting protein sequence. This is the link that allows the data model to separately maintain the nucleotide and protein sequences, with annotation on each sequence appropriate to that molecule. An mRNA feature on a genomic sequence could have as its product an mRNA Bioseq whose sequence reflects the results of posttranscriptional RNA editing. Features also have information unique to the kind of feature. For example, the CDS feature has fields for the genetic code and reading frame, whereas the tRNA feature has information on the amino acid transferred.

This design completely modularizes the components required by each feature type. If a particular feature type calls for a new field, no other field is affected. A new feature type, even a very complex one, can be added without changing the existing features. This means that software used to display feature locations on a sequence need consider only the location field common to all features.

Although the DDBJ/EMBL/GenBank feature table allows numerous kinds of features to be included (see Chapter 3), the NCBI data model treats some features as ''more equal'' than others. Specifically, certain features directly model the central dogma of molecular biology and are most likely to be used in making connections between records and in discovering new information by computation. These features are discussed next.

Genes. A gene is a feature in its own right. In the past, it was merely a qualifier on other features. The Gene feature indicates the location of a gene, a heritable region of nucleic acid sequence that confers a measurable phenotype. That phenotype may be achieved by many components of the gene being studied, including, but not limited to, coding regions, promoters, enhancers, and terminators. The Gene feature is meant to approximately cover the region of nucleic acid considered by workers in the field to be the gene. This admittedly fuzzy concept has an appealing simplicity, and it fits in well with higher-level views of genes such as genetic maps. It has practical utility in the era of large genomic sequencing when a biologist may wish to see just the ''xyz gene'' and not a whole chromosome. The Gene feature may also contain cross-references to genetic databases, where more detailed information on the gene may be found.

RNAs. An RNA feature can describe both coding intermediates (e.g., mRNAs) and structural RNAs (e.g., tRNAs, rRNAs). The locations of an mRNA and the corresponding coding region (CDS) completely determine the locations of 5′ and 3′ untranslated regions (UTRs), exons, and introns.

Coding Regions. A Coding Region (CDS) feature in the NCBI data model can be thought of as ''instructions to translate'' a nucleic acid into its protein product, via a genetic code (Fig. 2.3). A coding region serves as a link between the nucleotide and protein. It is important to note that several situations can provide exceptions to the classical colinearity of gene and protein. Translational stuttering (ribosomal slippage), for example, merely results in the presence of overlapping intervals in the feature's location Seq-loc.

The genetic code is assumed to be universal unless explicitly given in the Coding Region feature. When the genetic code is not followed at specific positions in the sequence—for example, when alternative initiation codons are used in the first position, when suppressor tRNAs bypass a terminator, or when selenocysteine is added —the Coding Region feature allows these anomalies to be indicated.

Proteins. A Protein feature names (or at least describes) a protein or proteolytic product of a protein. A single protein Bioseq may have many Protein features on it. It may have one over its full length describing a pro-peptide, the primary product of translation. (The name in this feature is used for the `/product` qualifier in the CDS feature that produces the protein.) It may have a shorter protein feature describing the mature peptide or, in the case of viral polyproteins, several mature peptide features. Signal peptides that guide a protein through a membrane may also be indicated.

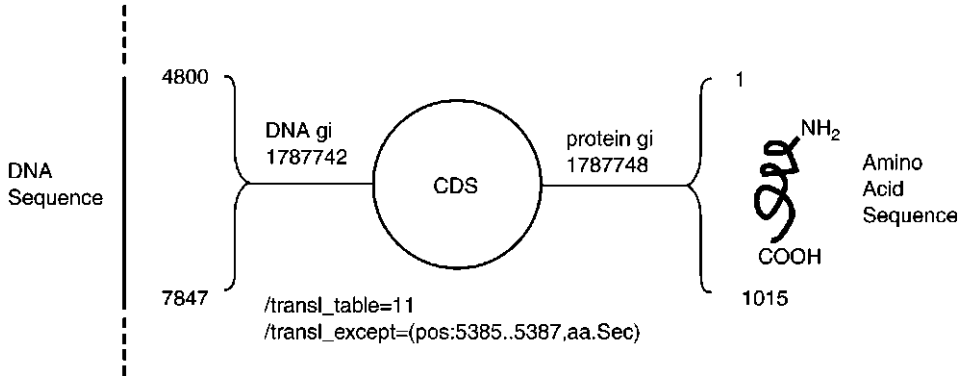


Figure 2.3. The Coding Region (CDS) feature links specific regions on a nucleotide sequence with its encoded protein product. All features in the NCBI data model have a "location" field, which is usually one or more intervals on a sequence. (Multiple intervals on a CDS feature would correspond to individual exons.) Features may optionally have a "product" field, which for a CDS feature is the entirety of the resulting protein sequence. The CDS feature also contains a field for the genetic code. This appears in the GenBank flat file as a `/transl_table` qualifier. In this example, the Bacterial genetic code (code 11) is indicated. A CDS may also have translation exceptions indicating that a particular residue is not what is expected, given the codon and the genetic code. In this example, residue 196 in the protein is selenocysteine, indicated by the `/transl_except` qualifier. NCBI software includes functions for converting between codon locations and residue locations, using the CDS as its guide. This capability is used to support the historical conventions of GenBank format, allowing a signal peptide, annotated on the protein sequence, to appear in the GenBank flat file with a location on the nucleotide sequence.

Others. Several other features are less commonly used. A Region feature provides a simple way to name a region of a chromosome (e.g., ''major histocompatibility complex'') or a domain on a polypeptide. A Bond feature annotates a bond between two residues in a protein (e.g., disulfide). A Site feature annotates a known site (e.g., active, binding, glycosylation, methylation, phosphorylation).

Finally, numerous features exist in the table of legal features, covering many aspects of biology. However, they are less likely than the above-mentioned features to be used for making connections between records or for making discoveries based on computation.

Seq-align: Alignments

Sequence alignments simply describe the relationships between biological sequences by designating portions of sequences that correspond to each other. This correspondence can reflect evolutionary conservation, structural similarity, functional similarity, or a random event. An alignment can be generated algorithmically by software (e.g., BLAST produces a Seq-annot containing one or more Seq-aligns) or directly by a scientist (e.g., one who is submitting an aligned population study using a favorite alignment tool and a submission program like Sequin; cf. Chapter 4). The Seq-align is designed to capture the final result of the process, not the process itself. Aligned regions can be given scores according to the probability that the alignment is a chance occurrence.

Regardless of how or why an alignment is generated or what its biological significance may be, the data model records, in a condensed format, which regions of which sequences are said to correspond. The fundamental unit of an alignment is a segment, which is defined as an unbroken region of the alignment. In these segments, each sequence is present either without gaps or is not present at all (completely gapped). The alignment below has four segments, delineated by vertical lines:

```
MRLTLLC-------EGEEGSELPLCASCGQRIELKYKPECYPDVKNSLHV
MRLTLLCCTWREERMGEEGSELPVCASCGQRLELKYKPECFPDVKNSIHA
MRLTCLCRTWREERMGEEGSEIPVCASCGQRIELKYKPE-----------
|      |       |                          |          |
```

Note that mismatches do not break a segment; only a gap opening or closing event will force the creation of a new segment.

By structuring the alignment in this fashion, it can be saved in condensed form. The data representation records the start position in sequence coordinates for each sequence in a segment and the length of the segment. If a sequence is gapped in a segment, its start position is -1. Note that this representation is independent of the actual sequence; that is, nucleotide and protein alignments are represented the same way, and only the score of an alignment gives a clue as to how many matches and mismatches are present in the data.

The Sequence Is Not the Alignment

Note that the gaps in the alignment are not actually represented in the Bioseqs as dashes. A fundamental property of the genetic code is that it is ''commaless'' (Crick et al., 1961). That is, there is no ''punctuation'' to distinguish one codon from the

next or to keep translation in the right frame. The gene is a contiguous string of nucleotides. We remind the reader that sequences themselves are also ''gapless.'' Gaps are shown only in the alignment report, generated from the alignment data; they are used only for comparison.

Classes of Alignments

Alignments can exist by themselves or in sets and can therefore represent quite complicated relationships between sequences. A single alignment can only represent a continuous and linear correspondence, but a set of alignments can denote a continuous, discontinuous, linear, or nonlinear relationship among sequences. Alignments can also be local, meaning that only portions of the sequences are included in the alignment, or they can be global, so that the alignment completely spans all the sequences involved.

A continuous alignment does not have regions that are unaligned; that is, for each sequence in the alignment, each residue between the lowest-numbered and highest-numbered residues of the alignment is also contained in the alignment. More simply put, there are no pieces missing. Because such alignments are necessarily linear, they can be displayed with one sequence on each line, with gaps representing deletions or insertions. To show the differences from a ''master'' sequence, one of the sequences can be displayed with no gaps and no insertions; the remaining sequences can have gaps or inserted segments (often displayed above or below the rest of the sequence), as needed. If pairwise, the alignment can be displayed in a square matrix as a squiggly line traversing the two sequences.

A discontinuous alignment contains regions that are unaligned. For example, the alignment below is a set of two local alignments between two protein sequences. The regions in between are simply not aligned at all:

```
12 MA-TLICCTWREGRMG 26 45 KPECFPDVKNSIHV 58
15 MRLTLLCCTWREERMG 30 35 KPECFPDAKNSLHV 48
```

This alignment could be between two proteins that have two matching (but not identical) structural domains linked by a divergent segment. There is simply no alignment for the regions that are not shown above. A discontinuous alignment can be linear, like the one in the current example, so that the sequences can still be shown one to a line without breaking the residue order. More complicated discontinuous alignments may have overlapping segments, alignments on opposite strands (for nucleotides), or repeated segments, so that they cannot be displayed in linear order. These nonlinear alignments are the norm and can be displayed in square matrices (if pairwise), in lists of aligned regions, or by complex shading schemes.

Data Representations of Alignments

A continuous alignment can be represented as a single list of coordinates, as described above. Depending on whether the alignment spans all of the sequences, it can be designated global or local.

Discontinuous alignments must be represented as sets of alignments, each of which is a single list of coordinates. The regions between discontinuous alignments are not represented at all in the data, and, to display these regions, the missing pieces

must be calculated. If the alignment as a whole is linear, the missing pieces can be fairly simply calculated from the boundaries of the aligned regions. A discontinuous alignment is usually local, although if it consists of several overlapping pieces it may in fact represent a global correspondence between the sequences.

Seq-graph: Graphs

Graphs are the third kind of annotation that can go into Seq-annots. A Seq-graph defines some continuous set of values over a defined interval on a Bioseq. It can be used to show properties like G + C content, surface potential, hydrophobicity, or base accuracy over the length of the sequence.

SEQ-DESCR: DESCRIBING THE SEQUENCE

A Seq-descr is meant to describe a Bioseq (or Bioseq-set) and place it in its biological and/or bibliographic context. Seq-descrs apply to the whole Bioseq or to the whole of each Bioseq in the Bioseq-set to which the Seq-descr is attached.

Descriptors were introduced in the NCBI data model to reduce redundant information in records. For example, the protein products of a nucleotide sequence should always be from the same biological source (organism, tissue) as the nucleotide itself. And the publication that describes the sequencing of the DNA in many cases also discusses the translated proteins. By placement of these items as descriptors at the Nuc-prot set level, only one copy of each item is needed to properly describe all the sequences.

BioSource: The Biological Source

The BioSource includes information on the source organism (scientific name and common name), its lineage in the NCBI integrated taxonomy, and its nuclear and (if appropriate) mitochondrial genetic code. It also includes information on the location of the sequence in the cell (e.g., nuclear genome or mitochondrion) and additional modifiers (e.g., strain, clone, isolate, chromosomal map location).

A sequence record for a gene and its protein product will typically have a single BioSource descriptor at the Nuc-prot set level. A population or phylogenetic study, however, will have BioSource descriptors for each component. (The components can be nucleotide Bioseqs or they can themselves be Nuc-prot sets.) The BioSources in a population study will have the same organism name and usually will be distinguished from each other by modifier information, such as strain or clone name.

MolInfo: Molecule Information

The MolInfo descriptor indicates the type of molecule [e.g., genomic, mRNA (usually isolated as cDNA), rRNA, tRNA, or peptide], the technique with which it was sequenced (e.g., standard, EST, conceptual translation with partial peptide sequencing for confirmation), and the completeness of the sequence [e.g., complete, missing the left (5′ or amino) end, missing both ends]. Each nucleotide and each protein should get its own MolInfo descriptor. Normally, then, this descriptor will not appear at-

tached at the Nuc-prot set level. (It may go on a Seg set, since all parts of a segmented Bioseq should be of the same type.)

USING THE MODEL

There are a number of consequences of using the NCBI data model for building databases and generating reports. Some of these are discussed in the remainder of this section.

GenBank Format

GenBank presents a ''DNA-centered'' view of a sequence record. (GenPept presents the equivalent ''protein-centered'' view.) To maintain compatibility with these historical views, some mappings are performed between features on different sequences or between overlapping features on the same sequence.

In GenBank format, the protein product of a coding region feature is displayed as a `/translation` qualifier, not as a sequence that can have its own features. The largest protein feature on the product Bioseq is used as the `/product` qualifier. Some of the features that are actually annotated on the protein Bioseq in the NCBI data model, such as mature peptide or signal peptide, are mapped onto the DNA coordinate system (through the CDS intervals) in GenBank format.

The Gene feature names a region on a sequence, typically covering anything known to affect that gene's phenotype. Other features contained in this region will pick up a `/gene` qualifier from the Gene feature. Thus, there is no need to separately annotate the `/gene` qualifier on the other features.

FASTA Format

FASTA format contains a definition line and sequence characters and may be used as input to a variety of analysis programs (see Chapter 3). The definition line starts with a right angle bracket (>) and is usually followed by the sequence identifiers in a parsable form, as in this example:

```
>gi|2352912|gb|AF012433.1|HSDDT2
```

The remainder of the definition line, which is usually a title for the sequence, can be generated by software from features and other information in a Nuc-prot set.

For a segmented Bioseq, each raw Bioseq part can be presented separately, with a dash separating the segments. (The regular BLAST search service uses this method for producing search databases, so that the resulting ''hits'' will map to individual GenBank records.) The segmented Bioseq can also be treated as a single sequence, in which case the raw components will be catenated. (This form is used for generating the BLAST neighbors in Entrez; see Chapter 7.)

BLAST

The Basic Local Alignment Search Tool (BLAST; Altschul et al., 1990) is a popular method of ascertaining sequence similarity. The BLAST program takes a query se-

quence supplied by the user and searches it against the entire database of sequences maintained at NCBI. The output for each ''hit'' is a Seq-align, and these are combined into a Seq-annot. (Details on performing BLAST searches can be found in Chapter 8.)

The resulting Seq-annot can be used to generate the traditional BLAST printed report, but it is much more useful when viewed with software tools such as Entrez and Sequin. The viewer in these programs is now designed to display alignment information in useful forms. For example, the Graphical view shows only insertions and deletions relative to the query sequence, whereas the Alignment view fetches the individual sequences and displays mismatches between bases or residues in aligned regions. The Sequence view shows the alignment details at the level of individual bases or residues. This ability to zoom in from an overview to fine details makes it much easier to see the relationships between sequences than with a single report.

Finally, the Seq-annot, or any of its Seq-aligns, can be passed to other tools (such as banded or gapped alignment programs) for refinement. The results may then be sent back into a display program.

Entrez

The Entrez sequence retrieval program (Schuler et al., 1996; cf. Chapter 7) was designed to take advantage of connections that are captured by the NCBI data model. For example, the publication in a sequence record may contain a MEDLINE UID or PubMed ID. These are direct links to the PubMed article, which Entrez can retrieve. In addition, the product Seq-loc of a Coding Region feature points to the protein product Bioseq, which Entrez can also retrieve. The links in the data model allow retrieval of linked records at the touch of a button. The Genomes division in Entrez takes further advantage of the data model by providing ''on the fly'' display of certain regions of large genomes, as is the case when one hits the ProtTable button in Web Entrez.

Sequin

Sequin is a submission tool that takes raw sequence data and other biological information and assembles a record (usually a Bioseq-set) for submission to one of the DDBJ/EMBL/GenBank databases (Chapter 4). It makes full use of the NCBI data model and takes advantage of redundant information to validate entries. For example, because the user supplies both the nucleotide and protein sequences, Sequin can determine the coding region location (one or more intervals on the nucleotide that, through the genetic code, produce the protein product). It compares the translation of the coding region to the supplied protein and reports any discrepancy. It also makes sure that each Bioseq has BioSource information applied to it. This requirement can be satisfied for a nucleotide and its protein products by placing a single BioSource descriptor on the Nuc-prot set.

Sequin's viewers are all interactive, in that double-clicking on an existing item (shown as a GenBank flatfile paragraph or a line in the graphical display of features on a sequence) will launch an editor for that item (e.g., feature, descriptor, or sequence data).

LocusLink

LocusLink is an NCBI project to link information applicable to specific genetic loci from several disparate databases. Information maintained by LocusLink includes official nomenclature, aliases, sequence accessions (particularly RefSeq accessions), phenotypes, Enzyme Commission numbers, map information, and Mendelian Inheritance in Man numbers. Each locus is assigned a unique identification number, which additional databases can then reference. LocusLink is described in greater detail in Chapter 7.

CONCLUSIONS

The NCBI data model is a natural mapping of how biologists think of sequence relationships and how they annotate these sequences. The data that results can be saved, passed to other analysis programs, modified, and then displayed, all without having to go through multiple format conversions. The model definition concentrates on fundamental data elements that can be measured in a laboratory, such as the sequence of an isolated molecule. As new biological concepts are defined and understood, the specification for data can be easily expanded without the need to change existing data. Software tools are stable over time, and only incremental changes are needed for a program to take advantage of new data fields. Separating the specification into domains (e.g., citations, sequences, structures, maps) reduces the complexity of the data model. Providing neighbors and links between individual records increases the richness of the data and enhances the likelihood of making discoveries from the databases.

REFERENCES

Altschul, S. F., Gish, W., Miller, W., Meyers, E. W., and Lipman, D. J. (1990). Basic Local Alignment Search Tool. *J. Mol. Biol.* 215, 403–410.

Crick, F. H. C., Barnett, L., Brenner, S., and Watts-Tobin, R. J. (1961). General nature of the genetic code for proteins. *Nature* 192, 1227–1232.

Ostell, J. M. (1995). Integrated access to heterogeneous biomedical data from NCBI. *IEEE Eng. Med. Biol.* 14, 730–736.

Ostell, J. M. (1996). The NCBI software tools. In *Nucleic Acid and Protein Analysis: A Practical Approach*, M. Bishop and C. Rawlings, Eds. (IRL Press, Oxford), p. 31–43.

Schuler, G. D., Epstein, J. A., Ohkawa, H., and Kans, J. A. (1996). Entrez: Molecular biology database and retrieval system. *Methods Enzymol*. 266, 141–162.

Zhang, J., and Madden, T. L. (1997). Power BLAST: A new network BLAST application for interactive or automated sequence analysis and annotation. *Genome Res.* 7, 649–656.