for example, a small packet gets stuck behind a large one while the large packet waits for enough bandwidth to become available. But an ATM-based network can predict, based on traffic, how long a transmission will take.

Moreover, because ATM sets up a virtual circuit for each transmission, it allows a user to request a specific quality of service in advance. A user running a multimedia application, for example, would request—and presumably pay for—service with very little cell delay because it's a real-time application, while e-mail users would be content with cheaper, slower service.

With software changes, ATM can run over existing Internet cables and routers, says Mark Laubach, who chairs an Internet Engineering Task Force working group on ATM, "but it doesn't work well unless done in hardware," meaning expensive new cables and other equipment. Still, ATM networks are up and running now: for example, the Bay Area Gigabit Testbed, an experimental high-speed ATM network connecting 15 sites in Northern California. It is used for a variety of collaborative scientific experiments including remote studies with optical and electron microscopes. MCI's Internet backbone (a major Internet pathway linking smaller users, the way an interstate highway connects feeder roads) also uses ATM.

Some lucky scientists stymied by the congestion on the Internet don't have to bother with caching, RSVP, or ATM. They can move off the existing network altogether. One "private roadway" already available to scientists is the NSF-sponsored vBNS (very high-speed Backbone Network Service), which connects five NSF supercomputing centers at 155 Mbs on an ATM network and provides bandwidth for cutting-edge network applications and research. It is not meant to be used for day-to-day operations such as e-mail and ftp, but that restriction may be difficult to maintain because the NSF is tying more universities and other sites into the vBNS.

That's the paradox of the Internet—and the reason that congestion is likely to plague scientists for the foreseeable future. Scientists move to high-speed networks, eventually everyone else jumps on board, and then the scientists have to move up another notch. "A few of us are out on the edge doing these things on very fast machines, and then 10 years later everyone else is doing it," says Paul Bash, a research scientist at Argonne National Laboratory. The Internet began as an experiment in computer networking, then became a popular phenomenon. Now it's groaning under the demand, and researchers are trying to make it safe for science again.

–Ellen Germain

Ellen Germain is a science writer in Arlington, Virginia.

COMPUTATIONAL MOLECULAR BIOLOGY

Software Matchmakers Help Make Sense of Sequences

Gene sequencers are spinning out data at a mind-boggling rate. They have already sequenced the complete genomes of several bacteria and brewer's yeast, they will have completed the genome of the roundworm *Caenorhabditis elegans* in a couple of years, and they intend to wrap up the human genome by 2005. A string of the four letters A, G, T, and C, designating the four nucleotides that make up DNA, is unreeling from sequencing labs at an ever-increasing pace, now nearly a million a day. For the human genome alone, the sequence will total 3 billion nucleotides.

All this would be little more than so much genetic ticker tape without some way to decipher its real meaning, which is largely hidden in the genes—the stretches of DNA, amounting to barely 3% of the human genome, coding for the proteins that are the workhorse molecules of life. The first step is to recognize the genes from their distinctive sequences of nucleotides. The next is to infer the function of the proteins they code for—and the key to doing that is to find related genes and proteins whose functions are already known.

As molecular evolutionist Russell Doolittle of the University of California, San Diego, explains, "The structures of all these proteins and the genes that code for them are all related through a big evolutionary expansion—some small number run through biochemical Xeroxes and used over and over in different settings." The challenge of learning the function of a newly generated sequence is the kind of challenge that computer scientists in other fields have been wrestling with for decades: spotting obvious, or less than obvious, similarities in different strings of data.

Welcome to the world of computational molecular biology. Over the past few years, biologists-turned–computer scientists and computer scientists–turned-biologists have begun churning out algorithms to find genes and other significant features in DNA sequences and to compare and contrast DNA, RNA, and protein sequences. The explosion has been triggered not only by supply—the information spewing from the genome projects—but also by demand from biologists hooked up to the World Wide Web, says David States, a computational biologist at the Institute for Biomedical Computing (IBC) at Washington University in St. Louis. "Most biologists in academic settings now have access to the Internet and Web browsers," says States, and that allows them to send their sequences to on-line analytical tools—or even borrow the tools and wield them on their own workstations (see p. 591).

This past June, States and his colleagues at the IBC hosted the Fourth International Conference on Intelligent Systems in Molecular Biology (ISMB) in St. Louis to survey the explosion. The computational tools under discussion ranged from simple programs that search for similarities between known and unknown sequences to ambitious efforts to find complete genes in DNA sequences and relate the proteins they produce to known protein structures. Many of the new tools rely on techniques developed by researchers in machine learning and artificial intelligence, and the hottest subject of the conference, known as hidden Markov models, springs directly from statistics and linguistics.

Sustaining all these efforts is a sense of mission, says Doolittle. The ISMB researchers "are missionaries and proselytizers, and they have this great esprit de corps." With the tools now under development, biologists "should be able to relate proteins whose relationships weren't detectable and do faster searching of genomes and comparisons of genomes," says Doolittle—"all sorts of things that weren't possible before."

Make me a match. One reason sequence comparisons are so powerful is evolution's conservative style. While the 20 amino acid alphabet of proteins could in theory spell out a nearly infinite number of proteins, actual proteins are variations on a limited number of themes. Human beings alone have perhaps 100,000 proteins, but we and other organisms "are dipping into a pool of relatively slowly evolving proteins we all share," says David Lipman, head of the National Center for Biotechnology Information (NCBI). All together, the number of different protein families is "maybe less than 1000." The result is that comparing an unknown gene to known ones has a reasonable chance of coming up with a match—providing the computer algorithm can recognize subtle similarities.

The first problem is to find the genes, which in higher organisms, known as eukaryotes, come interspersed with pieces of noncoding DNA called introns. One approach is to look for the telltale patterns of DNA that mark the boundaries between the coding and noncoding regions. Researchers have come up with various pattern-recognition

techniques for that purpose, says David Haussler, a computer scientist at the University of California, Santa Cruz. Among them are neural networks—computer algorithms that "learn," refining their ability to recognize a pattern as they are exposed to more examples of it—including the gene-finder most widely used for eukaryotes, the GRAIL program developed by Ed Uberbacher and Richard Mural of Oak Ridge National Laboratory in Tennessee. But along with exploiting clues in the unknown sequence itself, researchers can also determine whether it codes for a protein—and glean hints to that protein's function—by comparing it with known genes.

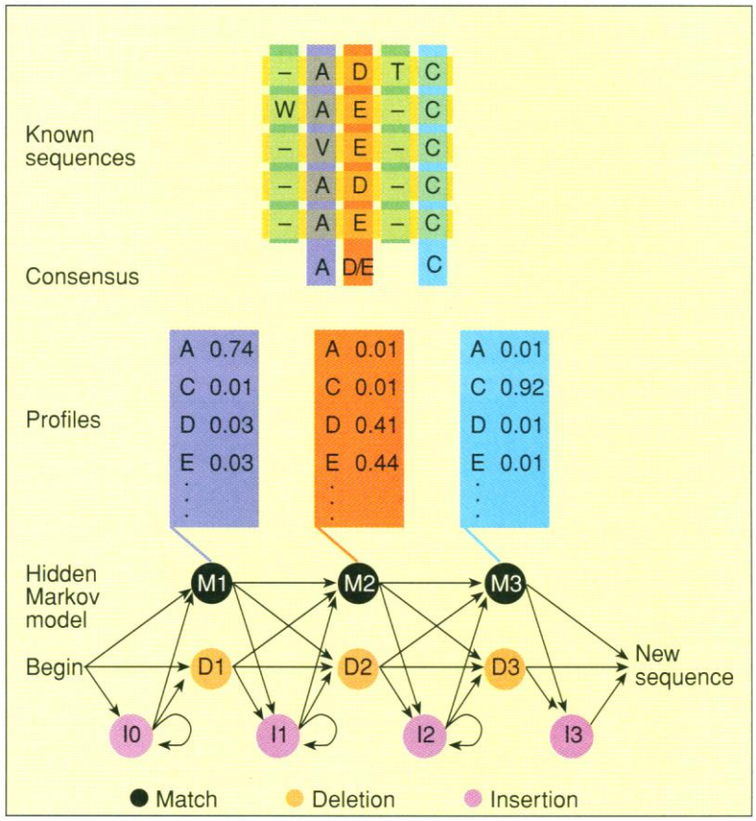
For the past few years, the two workhorse programs for that kind of comparison have been BLAST, written by researchers at the NCBI, and FASTA, written by computational biologist Bill Pearson at the University of Virginia. Both take an unknown sequence—DNA, RNA, or protein—and compare it to known sequences, looking for the best possible match. The programs then calculate the match's statistical power, which provides "a basis for saying that the relationship between two sequences may have some biological meaning," says geneticist Warren Gish, an author of BLAST, now at Washington University. "If something is known about the biological function of the database sequence, then we might infer our query sequence has the same or similar function, or the same or similar structure."

Both BLAST and FASTA are variations on an algorithm written in the 1980s by Mike Waterman at the University of Southern California in Los Angeles and Temple Smith of Boston University, but they use shortcuts that reduce computing time. BLAST, for instance, starts by scanning known sequences for short stretches of nucleotides or amino acids that are similar but not necessarily identical. The program then uses a scoring matrix for each match, awarding a positive, negative, or zero score, depending on how good a match it is. If the match is sufficiently close, then the program uses the sequence as a seed to proceed in both directions, comparing longer alignments "to see just how big an alignment score one can get," says Gish.

The most recent version of BLAST, which Gish discussed at the ISMB meeting, does a better job than its predecessors of taking into account small insertions or deletions of amino acids. As Stanford University computational biologist Michael Levitt explains, "It often happens that two sequences that are very similar to one another differ by just a few amino acids inserted or deleted relative to one another." These insertions and deletions can make sequence comparison difficult by knocking related sequences out of "register." Because of them, the old version of BLAST would often miss a significant match. But by adding up the scores of multiple high-scoring segments on the same sequence, then subtracting a penalty for any insertions or deletions, the new algorithm should now catch the similarity.

Programs like BLAST and FASTA can find matches for 40% to 50% of all new protein or gene sequences, says Lipman. Beyond this comes the twilight zone of sequence similarities, in which the potential homologies are less obvious because the evolutionary relations are more distant. One of the newest attempts to push into the twilight zone relies on the sophisticated statistics of hidden Markov models (HMMs).

These algorithms have their roots in the statistics work of the Russian mathematician A. Markov, who died in 1922. HMMs were first put to work in the mid-1960s in speech recognition programs, which address a problem similar to the one facing computational biologists: analyzing an unfamiliar string of data—a string of sounds in this case—to work out how similar it is to a known string. Haussler was the first to suggest that these software algorithms could be put to work on genome database searching problems, in a technical report he published in 1992 with his colleagues Anders Krogh, Saira Mian, and Kimmen Sjölander. The report quickly circulated through the community, says Sean Eddy of Washington University, "and while it was clearly not yet ready for prime time, it was also clear it had an awful lot of potential."



SOURCE: EDDY ET AL., J. COMPUT. BIOL.

A likely story. To build a hidden Markov model of a family of proteins, a computer aligns and analyzes known sequences from the family *(top)*. For each position along the sequence, it calculates a profile giving the likelihood that particular amino acids will be found there. The profiles feed into the model, which generates new sequences likely to belong to the same family. For each sequence position, the model incorporates an amino acid matching the profile, skips that position to simulate a deletion, or inserts a random amino acid.

Getting the essence. Instead of starting with an unknown sequence and looking for a match, HMM algorithms go the other way: They analyze a range of known sequences from a single family of proteins or genes, looking for the essential features of that family—a step generically known as creating a profile. The result is a model of what new members of the same family should look like—a hidden Markov model. For example, says Haussler, an HMM for the globin protein family, which includes hemoglobin, would try to capture the features that make globin proteins unmistakable: "The globin starts with a variable number of amino acids that occur before the first helix, called the A helix, which consists of 16 amino acids. The 16 positions in the A helix have propensities to be certain amino acids, and you can go through them and describe these propensities. Then after the first helix, there's a loop region consisting of a variable number of amino acids; then you start the B helix, etc. At some point you get to a position where an amino acid actually binds the heme iron, and that position is quite conserved among different globins. It has

to be a histidine."

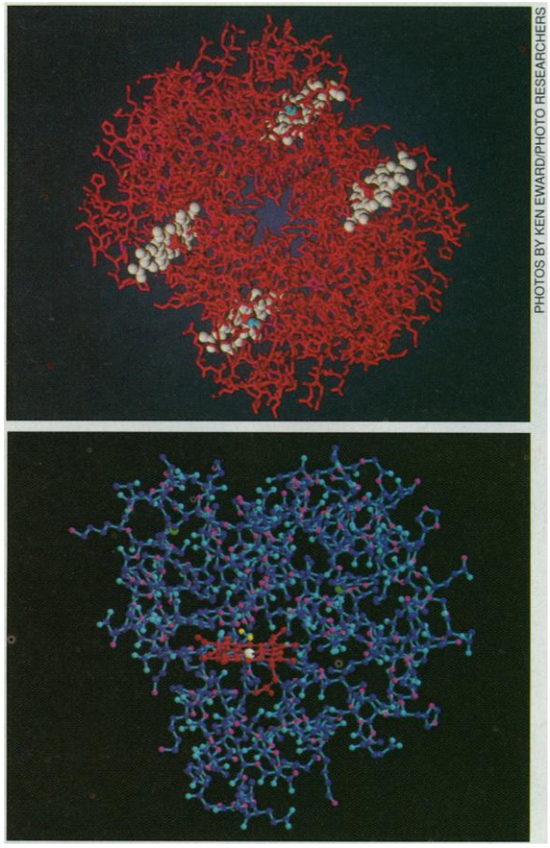
In the course of its learning process, the HMM takes examples of known globin sequences and a priori knowledge about the variability typically found in amino acid sequences, says Haussler. It then churns out a probability distribution for each globin residue position along the way, taking into account insertions and deletions that might change the register of one globin protein compared to another.

"It's not a black-and-white pattern recognition method," says Philippe Bucher, of the Swiss Institute for Experimental Cancer Research outside Lausanne. "It doesn't say this is allowed, this is not. It says that in the fifth position, there is a high probability that this amino acid is found and a very low probability that another amino acid is found, etc." Once HMMs for enough gene or protein families have been constructed, says Eddy, "we can take a newly predicted sequence, hand it to that software, and have it say it is very likely to belong to some specific protein family, say, or maybe it's a new family entirely."

HMMs were one of the hottest items at the June meeting, says computational biologist Chris Sander of the European Molecular Biology Laboratory in Heidelberg. But he adds that no one knows how useful they will turn out to be because they have yet to be widely used. ("What makes HMMs so popular," says Lipman, not entirely seriously, "is that the name is so tantalizing. Something is hidden and we're finding it and we have a Russian name to do it.") Eddy says, however, that HMM software has performed well at Washington University's Genome Sequencing Center, where he and his colleagues have used it for day-to-day analysis of sequences generated by a *C. elegans* sequencing project, and it seems to find matches for 5% more new sequences than does BLAST or FASTA. Adds Haussler, who has been testing HMMs for their ability to find distant matches, "It's not a panacea, but it should get you that little extra push."

HMMs, however, will never answer biologists' ultimate question, which is what a new gene's protein actually does. Sequence similarity to a known gene or protein doesn't give the full answer, because genes and proteins with completely different sequences sometimes perform similar functions. The key to a protein's function is its three-dimensional (3D) structure, and proteins with very different sequences occasionally fold up into similar shapes. So some biologists have tried skipping the process of matching sequences to known sequences and instead tried to match the new sequences directly to a structure.

Following a thread. The front-runners so far in this endeavor have been a class of algorithms, known appropriately as threading algorithms, that take an unknown sequence and try to thread it through a known structure to see how well it might fit. What makes threading algorithms promising, says Sander, is that instead of trying to predict the 3D structure of a protein based only on its sequence—a goal that computational biologists acknowledge lies far in the future—they proceed by making comparisons. Given a new sequence, he says, "they ask does it fit one of the several hundred known structures, yes or no? And if it does fit, what is the precise, best arrangement of [amino acid] sequences in that 3D structure? It's a clever way of simplifying the problem."



PHOTOS BY KEN EWARD/PHOTO RESEARCHERS

Family resemblance. Structures of hemoglobin *(top)* and myoglobin, two oxygen-binding proteins that have common evolutionary roots.

To answer those questions, Sander explains, threading algorithms look at how the unknown sequence and the known structure match up with respect to properties that affect a protein's folding, such as whether it is hydrophilic or hydrophobic at particular points. "When you thread the sequence through the structure," says Sander, "for each arrangement, you ask does a hydrophobic residue of the sequence end up in hydrophobic position of the structure, yes or no? If it does you give it a one; if it doesn't you give it a zero. Now you add up those numbers for all positions in the protein. And that gives you a number for one arrangement of the sequence in the structure. Then you push [the sequence] through further, and for the new arrangement you ask what is the number for this function and so on. And then you do it for every other known 3D protein sequence, and you compare what you have at the end" to find the most likely structure of the protein.

Threading algorithms originated in work by a number of investigators in the late 1980s. After an initial burst of popularity, they are now in what Sander describes as the "consolidation phase," which means, he says, that "there are improvements being reported consistently but with less excitement than the original round." Not only are the algorithms themselves being refined, says Lipman, but "as more structures become known, these threading methods will become that much more effective."

One recent illustration of their power, says Lipman, came last year when researchers discovered the obesity gene, so called because of its effect on mice when it is mutated (*Science*, 2 December 1994, p. 1477). Although researchers were unable to find a sequence match for the protein encoded by the gene, known as leptin, threading techniques showed that the new sequence was likely to have a structure resembling a well-studied class of proteins known as helical cytokines. A year later, when the receptor for the protein was sequenced, it turned out to be a cytokine receptor, confirming the threading prediction.

Lipman thinks that the prediction could have given the researchers a head start in the search for the receptor. "You could have leveraged the information," he says. "If you had looked in the sequence databases for examples of cytokine receptors that are not identical to ones we already know about, you would have been able to pull out a handful, and in that handful was the sequence for the leptin receptor a whole year before it was published. So these kinds of predictions could be extraordinarily useful."

Whatever the ultimate value of any particular technique, biocomputing experts say that their arsenal of comparative methods will become more powerful as the databases of known genes expand. "We have these islands of knowledge, and we can exploit each one to carry us to the next island of knowledge," says Lipman. "It's going to be easier and easier to do this sort of thing." Biologists will be well on their way to turning the data unreeling from the genome labs into real knowledge.

–Gary Taubes