

Profiles

By Katharine Miller

CIN Computer Science Courage

Reflections on the rewards of plunging into biomedicine





To a computer scientist, the fields of biology and medicine can seem like the vast Pacific Ocean, says **Leonidas Guibas, PhD**, professor of computer science at Stanford University. “You go to the edge and stare out for thousands of miles. How do you know where to go in? It’s scary.”

And Guibas is talking about how it feels today. That vast sea must have seemed quite daunting thirty years ago when the field of computational biology barely existed. But a few pioneers from computer science saw an opportunity to bring their skills and intuitions to bear in a new arena—an arena that could impact human health while also advancing the field of computer science. So they dove in: They learned the language of biomedicine; adjusted to a different peer-review and publishing system; and successfully developed a new field.

Today, many universities offer not only graduate degrees in computational biology, but undergraduate majors as well. Yet the field of biomedicine still presents tremendous opportunities to the pure computer scientist who knows little about the area.

The people profiled here provide a sampling of those opportunities. Some were pioneers thirty years ago; others are relative newcomers. Some now dedicate their careers to biomedicine while others still maintain a computer science focus. Their skills span a variety of computational techniques including computational algorithms, imaging, knowledge representation, robotics, machine learning, and computer vision. And they are applying their skills to biomedicine’s vast sea: genomic sequencing, molecular biology, phenotyping, drug design, epidemiology, neuroscience and more.

For researchers contemplating following in these scientists’ footsteps, it’s clear that each person must find his or her own path. Yet the stories and advice of these role models should prove reassuring: “The challenges in this space are never-ending and there’s always a need for smart people to look at the data and figure out how to extract the most information from them,” says **Daphne Koller, PhD**, professor of computer science at Stanford University.

Ron Shamir: A Gradual Transition

During the early years of his academic career, Ron Shamir had no idea that he would ever develop an interest in biology. He never took biology in school; his PhD thesis in operations research at the University of California, Berkeley in 1984 was completely theoretical; and his research after grad school focused on graph algorithms and optimization, with no biological applications. But around 1990, after presenting some research on temporal reasoning—an area of artificial intelligence—at Rutgers, an audience member (the late Gene Lawler from Berkeley) commented that it was a beautiful model for the physical mapping of DNA. The same approach—which determines whether the time periods for a group of overlapping events can be arranged to satisfy a set of constraints—could be used to study overlapping clones along the chromosome.

The comment sparked Shamir's curiosity and he started reading biology

texts. His wife, a biologist, helped him with the basics, and the newly launched human genome project fanned his fascination. "It was just serendipity," he says. He counts himself lucky to have been involved when things really took off. "It was evident that there would be a need for a good deal of computing. Otherwise the human genome project wouldn't fly."

In the late 1980s, 100 percent of Shamir's work involved optimization and graph algorithms. By the mid-1990s, 50 percent of that had been replaced by biologically motivated problems, and more recently, the vast majority of his work became driven by biology. Making the shift was pretty risky, he says. "I was moving into a discipline that had no name and none of my colleagues knew what I was talking about," he says. He also had to bridge a large cultural gap and language barrier, which have both shrunk a lot since. "But I didn't make a 90-degree turn. You do it gradually and build your confidence in the field over time."

Advice...

ON Taking the Plunge

Find Your Passion

"Not every computer scientist will fall in love with the field like I did," says Ron Shamir. "And that's an essential part of doing research: to be fascinated, excited and enthusiastic about what you do."

It helps, Bruce Donald says, to find a great lab where people are doing computational biology that excites you. "You must develop the ability to admire the work and decide whether you'd like to do work like that."

When he was a master's student at Stanford, Michael Black told a professor he wanted to do his PhD in cognitive science and study human perception. The professor told him, "If you were my brother, I'd tell you to get a computer science PhD because you'll make more money." Black took that advice, but managed to find his way back to cognitive science through the study of computer vision. The advice he gives his students is different: "Follow your heart."

Haussler agrees: "You're limited only by your passion and commitment."

Learn Some Biology

Certainly a computer scientist who wants to work in biomedicine must learn some biology. The question is, how much? According to Shamir, "Initially, a computer scientist can pick up what he or she needs to know about a biological problem by reading chapters in one good book. To get more seriously into the field, one has to attend conferences and follow the recent literature."

Daphne Koller suggests that instead of taking introductory biology classes, which can be descriptive rather than quantitative, computer-science graduate students should start by reading a more advanced textbook and some more computationally oriented papers in the good journals. "Get a sense for the kinds of work people are doing," she says. "Find a problem that interests you and then find the background courses and reading you need."



*Ron Shamir, PhD, professor in the School of Computer Science,
Raymond and Beverly Sackler Chair of Bioinformatics and
head of Edmond J. Safra Bioinformatics Program, Tel Aviv University*



David Haussler, PhD, professor of biomolecular engineering at the University of California, Santa Cruz

David Haussler: The Chance to Address Great Scientific Questions

David Haussler got a taste of biology when he worked in his older brother's biology lab at the University of Arizona in the 1970s and again when working

the field of bioinformatics really existed, Haussler says, but Ehrenfeucht led discussions of how to analyze DNA using computer algorithms.

But it wasn't until the 1990s that Haussler began using his computer expertise for biological applications. He was interested in artificial neural net-

Ehrenfeucht, Haussler proposed to his postdoc, Anders Krogh (now a professor at the University of Copenhagen), that they should apply neural nets and hidden Markov models to protein and DNA sequences. "So we tried it and it worked perfectly," Haussler says. Their paper on hidden Markov models is now a mainstay of bioinformatics. "It was one of those magic moments where things took off and very rapidly we were revisiting the earlier work and pulling together a unified viewpoint for the field," Haussler says.

Since then, Haussler's research has gradually become completely focused on biology. After more than ten years as a professor of computer science, he became a professor of bio-molecular engineering in 2004—reflecting his shift. He now supervises both experimental and computational biological research.

What draws him to apply computer science to biomedicine? Two things, he says: First, the chance to address some of the great scientific questions. "The questions we look at are among the greatest. How did we become human? How does the cell work? How did life come to be?" he says.

Second, he says, the chance to really affect medicine. Haussler works on the cancer genomics and cancer genome atlas projects, which apply

large-scale analysis to find all of the mutations in a tumor and determine which ones are driving the cancer.

Haussler also heads the Genome 10K project, which is dedicated to sequencing the genomes of 10,000 vertebrate species. The goal is to map out the evolutionary changes that produced the amazing diversity of life on

"The questions we look at are among the greatest.
How did we become human? How does the cell work?
How did life come to be?"

with his PhD advisor, Andrzej Ehrenfeucht, in the 1980s. "He was a real polymath, interested in all aspects of science," Haussler says. It was before

works and hidden Markov models—trying to get a handle on what was learnable by a machine. Then one day, recalling his happy days with

this planet, he says. "The computer science challenges are nothing short of enormous. This is an incredibly exciting time to be alive."



Michael Black, PhD, professor of computer science at Brown University

Michael Black: Changing Lives with Computation

Michael Black has a longstanding interest in human perception. He contemplated a graduate degree in cognitive science (but was advised to stick with computer science because “you’ll make more money”) and later enjoyed hanging out with cognitive scientists at NASA/Ames while working on his PhD on optical flow estimation. Yet Black’s career remained firmly rooted in computer science until he described his computer vision research to his wife’s French-Canadian grandmother. According to Black, she shook her head and said, “That’s all a lot of excess baggage. I’ve got my garden, my health, and my family. I’ve put away vegetables in the cellar for the winter. That’s all I need.” And Black thought, “She’s right!”

On the flight home from Canada, Black considered whether he might be able to use his skills to help people do the most basic and important things. And he sketched out an idea for a brain-machine

interface (BMI) to help paralyzed people gain back some of their independence. When initial support for these ideas evaporated at Xerox PARC where he worked at the time, he put it aside for a while. But when he landed a job at Brown University in 2000, he confided his interest in BMIs to a colleague. “I was sort of embarrassed because it sounded kind of crazy,”

Black says. But he was told, “that’s not crazy—there’s a guy here working on that.” Thus was launched a successful collaboration between Black and John Donoghue, a neuroscientist at Brown.

Ten years in, Black says, patients are using the brain-machine interface systems he helped develop. As a result, he’s driven less by computational elegance than by the patients’ needs and what’s practical for them. “It’s not just a scientific question anymore; it’s a usability question,” he says.

Although Black still does basic computer science work, his experience with biology has changed him, he says. Computer scientists are trained to think like engineers or mathematicians rather than experimentalists, he says. “Learning to think like a biologist has made me a better computer scientist.” He’s also developed a drive to work on problems that could change someone’s life. “I’m a little addicted to finding some of that in everything I do. It doesn’t have to be a biological impact, but I want to somehow affect peoples’ lives outside the academic realm.”

Advice...

When Gene Myers made the leap 30 years ago, “I was lucky enough not to have to know a darn thing,” he says. “I would have a conversation and do the best I could.” But now, because the level of sophistication in computational biology is increasing, he thinks more is needed. “Take some biology courses or go study with somebody in the field. I think at this point that’s a requirement.”

Donald agrees. In the early days, he says, people felt you didn’t need to know a lot of biology and biochemistry to pick a deep problem and work on it. “I’m not sure that’s true anymore,” he says. “I’m not sure it’s good enough to learn a little.”

Find Great Collaborators

Computer scientists agree that working in biomedicine depends on personal connections with biologists you can trust. Virtually all of those profiled here say they had great collaborators early on. “You need people you can ask, ‘am I doing the right thing?’” Leonidas Guibas says.

Today, because there are more biologists with a quantitative background, it’s easier to find people who both understand what computer scientists can offer and speak the same language, Myers says.

But beware collaborators who have a naive view of the computer scientist’s skill set, says Black. “They might see the computer scientist as the programmer who comes in and writes some code,” Black says. “Collaborators should understand that a computer science collaborator brings ideas and ways of looking at the problem and understanding the data and maybe whole new ways of thinking about what the biological system is doing.” Computer scientists also shouldn’t make the mistake of seeing biologists as a source of data, Black notes. “Collaborations require people to appreciate each other.”

Experiment with Experiments

All of the students in David Haussler’s lab have the opportunity to work in his wet lab. He doesn’t expect that the computer science-oriented students will remain there, but many do a stint out of curiosity and to broaden themselves.

Daphne Koller: Hammer Looking for a Nail (at first)

About ten years ago, Daphne Koller was working on a project to extract meaningful networks of relationships from complex heterogeneous data. She tested it on a dataset of scientific papers and authors and also on a database of movies, actors and directors, but wanted to try it on something even more complex. “I basically had a hammer and was looking for a nail,” she says. And because biological datasets were rich and readily available, she decided to see if her techniques would be valuable in biological analyses. She says, “Over the course of the first few months of working on the problems, I became more interested in the nail than the hammer.”

Koller’s hammer was useful for studying networks both in the clinical and molecular setting. Initially, she used her tools to study networks of tuberculosis patients in San Francisco. Koller has worked partially in computational biology ever since, while still researching hard-core machine learning and other computer science problems as her mainstay.

Koller likes the fact that her biological research can have a much more direct effect on peoples’ lives than can much of her computer science research. For example, she’s developed a tool to evaluate a neonate’s risk of developing major complications. Using only noninvasive data collected by a heart rate monitor during the first 3 hours of life, it calculates a risk score that is considerably more accurate than any other risk score previously proposed. She also developed a tool that finds pathways in cancer, the first step in identifying new drugs or personalizing cancer treatments. “I also really like the puzzle nature of trying to figure out how to take a new problem that no one has looked at computationally and thinking about how to model it, what’s the right way of thinking about it, what’s the right algorithmic approach. That’s very satisfying.”



Daphne Koller, PhD, professor of computer science at Stanford University

Eran Halperin: Having an Impact

Eran Halperin began his academic career as a computer scientist working on purely mathematical problems with little regard for applications. But, while working on his PhD, he joined a bioinformatics company. It completely changed his view. Compared with designing a new theoretical algorithm, if you find a new gene or potentially new treatment, he says, “you feel the impact on society much more strongly.”

When he moved on to postdoctoral research, Halperin gradually changed the focus of his research to the study of applications of computer science to biology. With Halperin's background in theoretical computer science, a natural way to choose the research problems would be to look for problems that are based on computational interest. However, Halperin doesn't choose his research problem this way. “It's not the driver,” he says. He chooses based on the potential impact of the project and whether his background provides some kind of advantage.

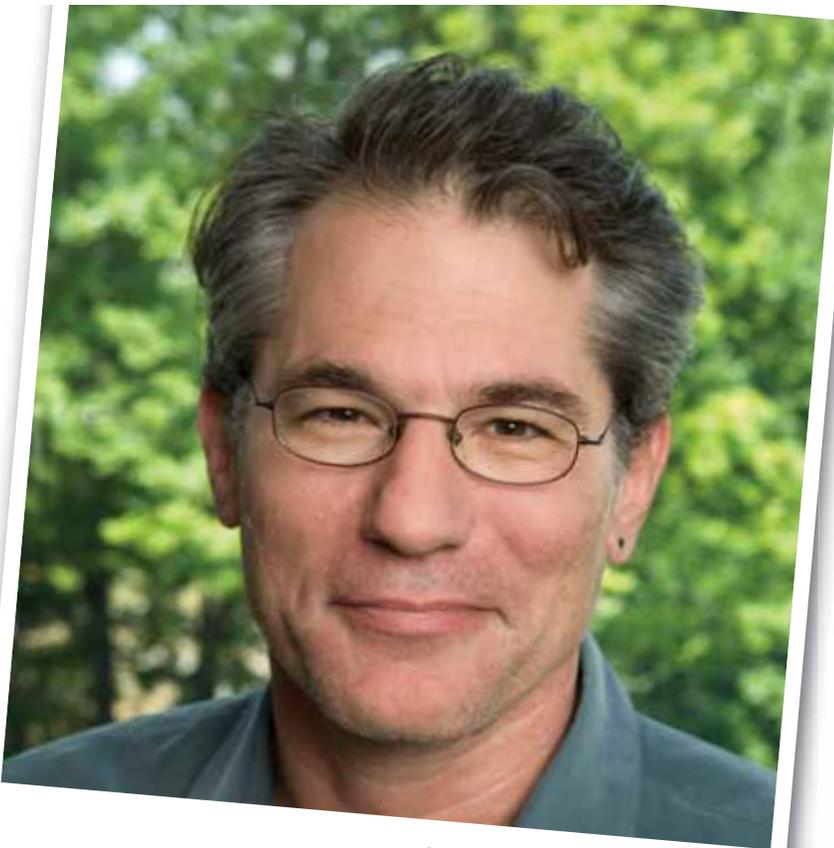
“What you learn in math and computer science is a way of thinking about a problem and how to attack it,” he says. That's something he brings to the table. Halperin is perfectly willing to set his ego aside to serve the needs of biomedical research. “Everything we do is service,” he says. “Eventually it serves the purpose of advancing science.”

“What you learn in math and computer science is a way of thinking about a problem and how to attack it.”



Eran Halperin, PhD, Senior Lecturer, Blavatnik School of Computer Science, and the Department of Molecular Microbiology and Biotechnology, Tel-Aviv University

Advice...



*Gene Myers, PhD, Group Leader,
HHMI Janelia Farm Research Campus, Ashburn, Virginia*

Gene Myers: Seeking Uncharted Territory

In the early 1980s, biology was primarily a descriptive field rather than a quantitative one, Gene Myers says. “Most biologists could not compute. And this created the opportunity.”

Myers was drawn to apply his computer science skills to biology—primarily in gene sequencing—because “it was a cool source of problems,” he says. “I was being challenged to extract inter-

esting variations on traditional problems. I’m a big fan of that.”

Myers believes that the most interesting computational work in molecular biology in the next 10 to 15 years will involve using microscopy to understand phenotype. “The genotype/phenotype correlation is not going to yield itself just by looking at genotype, which is what the DNA sequencing people are doing,” he says. Microscopy yields rich, high-dimensional data for phenotyping, which will

“It was nice to be one of the few researchers in the field because you had a lock on a niche.”

help researchers get the “most bang for the buck” from genomic data, he says. “So I realized that if I want to be in it, I’ve got to be an imaging guy.”

Because computer scientists divide themselves by technique, he says, “it’s very hard to get people to go from sequences to images.” In addition, most academicians at his career stage are “walking this tightrope of seeking funding and managing a large group of students,” he says. “They are basically supertankers, and changing the direc-

tion is a huge task. You have to have a very strong vision and a very strong team to pull it off.”

“The best way to learn is to teach,” Shamir says. “Teaching in a different discipline is hard, but it is also very rewarding.” Preparing course materials can be a bigger commitment than just writing papers, Shamir says. That was especially true in the 1990s when he started creating bioinformatics courses and there were virtually no textbooks. “I had to create the course out of primary journal papers. It was very hard work – but it taught me a lot and was fun.” And lecturing is a learning experience too. “You interact with young minds, force yourself to organize your knowledge systematically, and through the process come up with new research questions.”

“It’s important to learn and understand the other person’s language, concepts and worldview,” Haussler says. In the end, some might find they are adept at both the pipette and the keyboard. “They can lead a complete and rich double life,” Haussler says. “But not everyone has to do that to be successful in this field because we can do work in teams with people who complement each other.”

Ask Lots of Questions:

“Ask the right questions and don’t assume you know the biology,” Paul Groth says. A little biological knowledge is a dangerous thing. “You may miss something important when helping [biologists] design new systems or designing new computational approaches to what they’re doing,” he says.

Be Adaptable

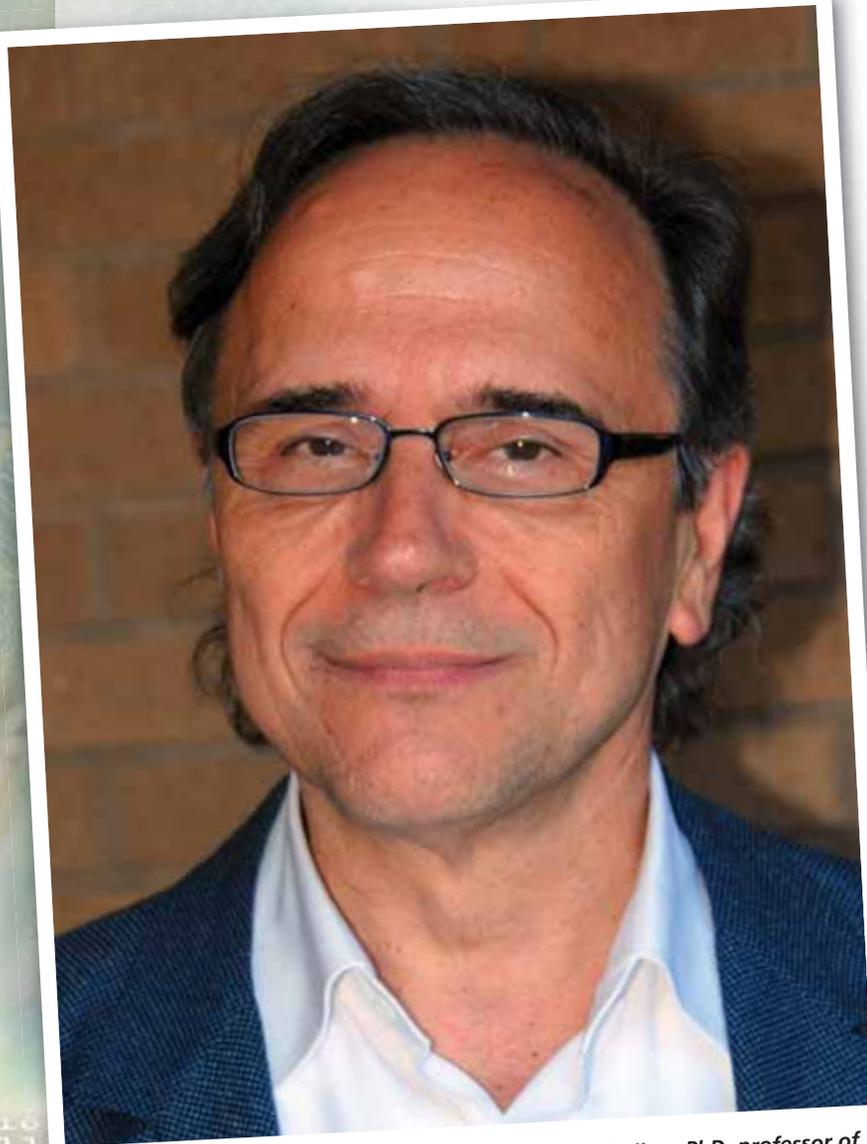
“If you only want to prove theorems, you will not get very far in biology,” Shamir says. “You have to compromise: if you can’t provide an elegant formal solution, you should be willing to sometimes work with heuristics and algorithms for which you aren’t able to prove much. And you have to work with real data and interact with biomedical experts who think differently and have different goals. This requires adaptation.”

Teach

“The best way to learn is to teach,” Shamir says. “Teaching in a different discipline is hard, but it is also very rewarding.” Preparing course materials can be a bigger commitment than just writing papers, Shamir says. That was especially true in the 1990s when he started creating bioinformatics courses and there were virtually no textbooks. “I had to create the course out of primary journal papers. It was very hard work – but it taught me a lot and was fun.” And lecturing is a learning experience too. “You interact with young minds, force yourself to organize your knowledge systematically, and through the process come up with new research questions.”

tion of the supertanker is hard.” But Myers made the switch anyway. “I was lucky to come to Janelia Farm and really have a chance to retread myself.”

When he moved to Janelia Farm, Myers says he felt like a postdoc for a few years as he got up to speed on imaging methods and developed an intuition about what techniques should work to solve computer vision problems. Now, Myers is exactly where he wants to be. He’s addressing imaging problems that have requirements no one has addressed before, he says. “It’s great. I’m in new territory.”



Leonidas Guibas, PhD, professor of computer science at Stanford University

Leonidas Guibas: Feeding on Biology’s Abstractions

Leonidas Guibas is driven to understand abstractions. He deals in the mathematics and algorithms for describing the shape and motion of things. For many years, he taught a course on geometric modeling in computer science graphics that covered only manufactured shapes such as car hoods, airplane fuselages and the like—geometric forms that people have designed. To take on the challenge of applying these same ideas to biological shapes, in 2003 he moved from Stanford University’s computer science building to the Bio-X program at Stanford’s Clark Center, where interdisciplinary research is encouraged.

For Guibas, biology offered the opportunity to study imprecise shapes such as protein surfaces, which have electrons floating around them. Studying proteins requires fundamentally different kinds of tools than those used to model the shape of a car or airplane, Guibas says. “That’s feeding me something interesting to work on.”

Guibas enjoys his interactions with biologists, but he’s clear where his interests differ from theirs. “I care about computation as an object of study by itself,” he says. “The biologists are interested in proteins because they are essential to life. I don’t have this predilection. I study proteins as something that has geometry to it. I’m interested in something more abstract—something with shape and motion that can help me develop mathematical tools and representations that are appropriate to proteins but may also have many other uses.”

“I study proteins as something that has geometry to it. I’m interested in something more abstract—something with shape and motion that can help me develop mathematical tools and representations that are appropriate to proteins but may also have many other uses.”

Advice...

Be Prepared to Slow Down

Some computer scientists find the pace of experimental science frustrating, Black says. Particularly in cognitive science, the area in which Black works, it takes time to train animals, perform required surgery, deal with governmental regulations, and obtain experimental observations. Human studies can be even more frustrating. People leave the study, patients die, "many things are out of your control," Black says. "So I've had some computer science students back away from the biology to stick with computer science."

Mind the Gap

When you work in interdisciplinary science, Guibas says, you have to decide what community you want to be part of. "There's a danger of falling in the gap between fields. Your work might be too computational for biological publications or too biologically specific for computer science publications." Computer science done for a biologist might end up in the fine print at the end of a biology paper, Guibas warns.

Be Both Bold and Careful

In computer science, Donald says, people are excited about creativity, spontaneity, innovation and boldness. "Computer science has an element of surprise," Donald says. "You're trying to make the computer do something that it couldn't obviously do before, such as redesign an enzyme to have a novel function."

Experimental scientists have a different set of values built around being careful and controlled, Donald says. They dot their i's and cross their t's. Being bold, as computer scientists are wont to do, might seem risky to them, Donald says.

But really, the kind of care that is necessary to doing biology can be useful in computer science, Donald says. And at the same time, "the boldness and creativity of computer science in the hacker generation can be really exciting for trying new approaches in biomedical research." Computational biologists can do both: be bold and careful at the same time.



Paul Groth, PhD, postdoc in the artificial intelligence department at the Free University in Amsterdam, the Netherlands.

Paul Groth: Biology Drives Interesting Computation

Paul Groth's career has been built around e-Science, computationally intensive science carried out over a network. "In e-Science you get really difficult computer science problems as well as really simple ones," Groth says. And many of the more complex ones come from biology. "That's the key interest for me as a computer scientist," he says.

In any kind of scientific research, it's essential to know where the data come from—i.e., their provenance—Groth says. In biology, where vast storehouses of remote data continue to grow and change, figuring out how to connect provenance information to the data themselves has become an interesting area of computer science research involving both knowledge representation (how to describe where results come from) and distributed systems (if data are coming from many different places, how do we capture

and store that information?).

"Biology drove this computer science problem," Groth says, "because biologists were the first to really use publicly available data provided by web services that they didn't control and that could be updated remotely."

Often, Groth says, biologists might ask for a solution to a simple problem. "As the computer scientist, you have to ask what they would really want," he says. "You end up discovering the bigger computer science problem behind the little problem."

For example, Groth worked with a bioinformatician who wrote many different scripts but would then forget which version was the one that produced his results. "This sounds like a simple thing of being more organized," Groth says, "but in the end, the question was how to help him automatically determine what he did, which turns out to be not a simple problem." And

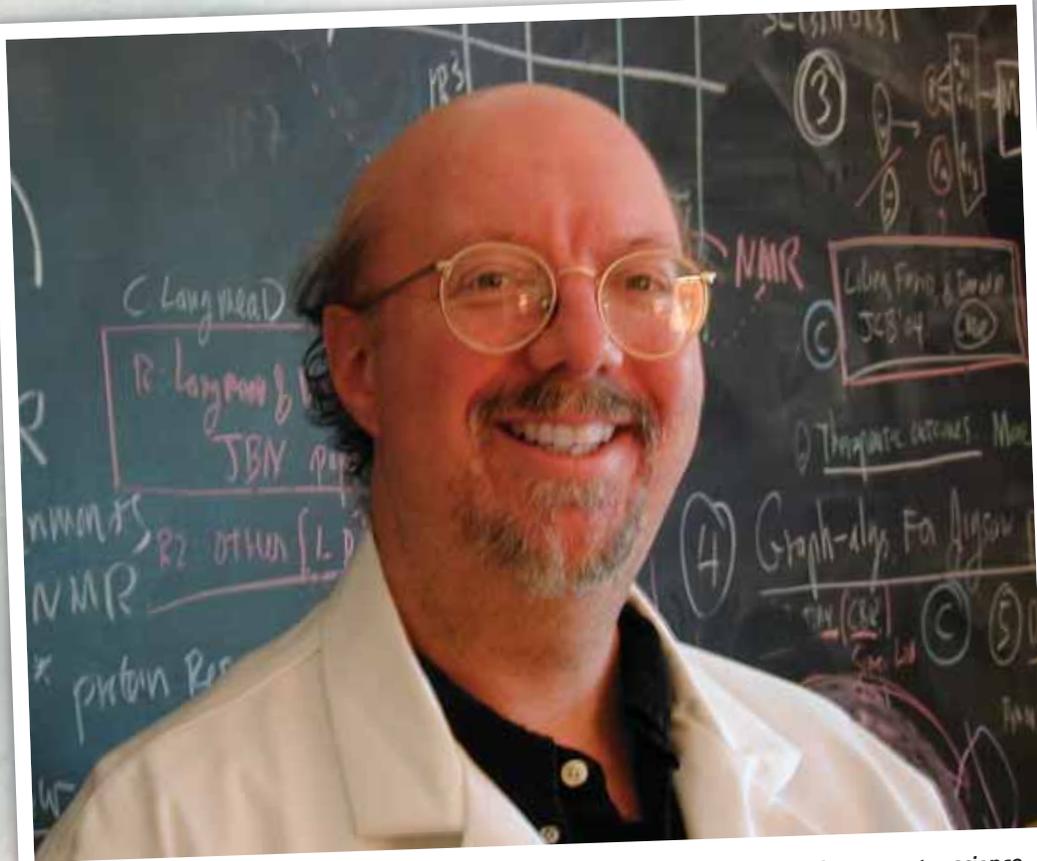
when the researcher moved from a desktop to a supercomputer, Groth had to address the more complex provenance questions raised by a supercomputer consisting of multiple machines where the mechanics could fail in many different ways. Solving this problem for one researcher led to

Bruce Donald: End-to-End Computational Biology

When Bruce Donald began his career in robotics in the early 1980s, he was excited about the opportunity to do what he calls “end-to-end” work. In

remained his predilection. While his lab focuses on developing mathematical and highly sophisticated algorithms, they don’t stop there. They will also engage in a substantial software project, implement it, and test it experimentally. That might mean, for example, performing nuclear magnetic resonance on certain proteins, developing algorithms to determine a realistic structure that captures real properties of the proteins (for example, its flexibility), predicting algorithmically how that structure would interact with a library of possible drugs, and then testing that prediction experimentally to find a drug with the desired characteristics.

To make that start-to-finish approach a reality, Donald collaborates with experimentalists. But, more recently, he gathers together the necessary experimental techniques in his own lab. “That’s what I’m most excited about and proud of,” he says. “A real algorithmic accomplishment is one that when applied to real data and real protein systems, really works and produces some insight in bio-



Bruce Donald, PhD, professor in the computer science and biochemistry departments at Duke University

software that could then be used by others. “The common representation we helped develop, called the open provenance model, is now becoming widely deployed,” Groth says, as are several other systems developed by other groups. “Biology is a very good example of why you need this sort of provenance model.”

Often, Groth says, the computer scientist has to be clear that he or she is not a programmer for the biologist. “I’m not here to design perhaps the program that helps you immediately,” he says. “I might do that because I’m a nice guy and it helps me understand the collaboration. But in the end, I’m looking for the computer science research challenges that will help you eventually.”

robotics, a researcher could go from math, to algorithms, to software, to simulations, to actually making metal or silicon move.

medical research.”

And working on problems with relevance to human health has another benefit, Donald says. “I don’t really

“A real algorithmic accomplishment is one that when applied to real data and real protein systems, really works and produces some insight in biomedical research.”

In 1998-99, when Donald turned his attention to using computer science for structural biology, “end-to-end” work

have to ask myself why it’s important. It’s manifestly important and manifestly interesting as well.” □