A vial of bacteria capable of computation? Injectable cells that survey the bloodstream and produce drugs on demand? These ideas might not be as far-fetched as they sound.

BY SIMSON L. GARFINKEL

Biological Computing

Today's silicon-based microprocessors are manufactured under the strictest of conditions. Massive filters clean the air of dust and moisture, workers don spacesuit-like gear and the resulting systems are micro-tested for the smallest imperfection. But at a handful of labs across the country, researchers are building what they hope will be some of tomorrow's computers in environments that are far from sterile—beakers, test tubes and petri dishes full of bacteria. Simply put, these scientists seek to create cells that can compute, endowed with "intelligent" genes that can add numbers, store the results in some kind of memory bank, keep time and perhaps one day even execute simple programs.

All of these operations sound like what today's computers do. Yet these biological systems could open up a whole different realm of computing. "It is a mistake to envision the kind of computation that we are envisioning for living cells as being a replacement for the kinds of computers that we have now," says Tom Knight, a researcher at the MIT Artificial Intelligence Laboratory and one of the leaders in the biocomputing movement. Knight says these new computers "will be a way of bridging the gap to the chemical world. Think of it more as a process-control computer. The computer that is running a chemical factory. The computer that makes your beer for you."

As a bridge to the chemical world, biocomputing is a natural. First of all, it's extremely cost-effective. Once you've programmed a single cell, you can grow billions more for the cost of simple nutrient solutions and a lab technician's time. In the second place, biocomputers might ultimately be far more reliable than computers built from wires and silicon, for the same reason that our brains can survive the death of millions of cells and still function, whereas your Pentium-powered PC will seize up if you cut one wire. But the clincher is that every cell has a miniature chem-

PHOTOGRAPHS BY JOHN SOARES

Knight vision: Tom Knight sees great possibilities for computers built into cells.

ical factory at its command: Once the organism was programmed, virtually any biological chemical could be synthesized at will. That's why Knight envisions biocomputers running all kinds of biochemical systems and acting to link information technology and biotechnology.

Realizing this vision, though, is going to take a while. Today a typical desktop computer can store 50 billion bits of information. As a point of comparison, Tim Gardner, a graduate student at Boston University, recently made a genetic system that can store a single bit of information—either a 1 or a 0. On an innovation timeline, today's microbial programmers are roughly where the pioneers of computer science were in the 1920s, when they built the first digital computers.

Indeed, it's tempting to dismiss this research as an academic curiosity, something like building a computer out of Tinker Toys. But if the project is successful the results could be staggering. Instead of painstakingly isolating proteins, mapping genes and trying to decode the secrets of nature, bioengineers could simply program cells to do whatever was desired—say, injecting insulin as needed into a diabetic's bloodstream—much the way that a programmer can manipulate the functions of a PC. Biological machines could usher in a whole new world of chemical control.

In the long run, Knight and others say, biocomputing could create active Band-Aids capable of analyzing an injury and healing the damage. The technology could be used to program bacterial spores that would remain dormant in the soil until a chemical spill occurred, at which point the bacteria would wake up, multiply, eat the chemicals and return to dormancy.

In the near term—perhaps within five years—"a soldier might be carrying a biochip device that could detect when some toxin or agent is released," says Boston University professor of biomedical engineering James Collins, another key player in the biocomputing field.

The New Biology

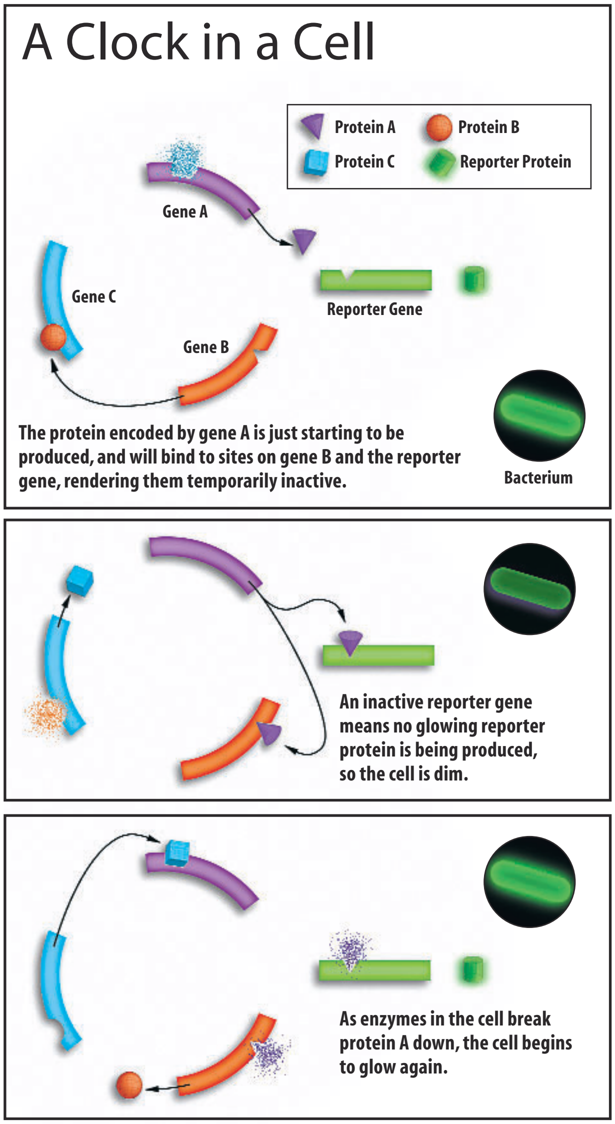
Biocomputing research is one of those new disciplines that cuts across well-established fields—in this case computer science and biology—but doesn't fit comfortably into either culture. "Biologists are trained for discoveries," says Collins. "I don't push any of my students towards discovery of a new component in a biological system." Rockefeller University postdoctoral fellow Michael Elowitz explains this difference in engineering terms: "Typically in biology, one tries to reverse-engineer circuits that have already been designed and built by evolution." What Collins, Elowitz and others want to do instead is forward-engineer biological circuits, or build novel ones from scratch.

But while biocomputing researchers' goals are quite different from those of cellular and molecular biologists, many of the tools they rely on are the same. And working at a bench in a biologically oriented "wet lab" doesn't come easy for computer scientists and engineers—many of whom are used to machines that faithfully execute the commands that they type. But in the wet lab, as the saying goes, "the organism will do whatever it damn well pleases."

After nearly 30 years as a computer science researcher, MIT's Knight began to set up his biological lab three years ago, and nothing worked properly. Textbook reactions were failing. So after five months of frustratingly slow progress, he hired a biologist from the University of California, Berkeley, to come in and figure out what was wrong. She flew cross-country bearing flasks of reagents, biological samples—even her own water. Indeed, it turned out that the water in Knight's lab was the culprit: It wasn't pure enough for gene splicing. A few days after that diagnosis, the lab was up and running.

Boston University's Gardner, a physicist turned computer scientist, got around some of the challenges of setting up a lab by borrowing space from B.U. biologist Charles Cantor, who has been a leading figure in the Human Genome Project. But before Gardner turned to the flasks, vials and culture dishes, he spent the better part of a year working with Collins to build a mathematical model for their genetic one-bit switch, or "flip-flop." Gardner then set about the arduous task of realizing that model in the lab.

The flip-flop, explains Collins, is built from two genes that are mutually antagonistic: When one is active, or "expressed," it turns the second off, and vice versa. "The idea is that you can flip between these two states with some external influence," says Collins. "It might be a blast of a chemical or a



A Clock in a Cell

The protein encoded by gene A is just starting to be produced, and will bind to sites on gene B and the reporter gene, rendering them temporarily inactive.

An inactive reporter gene means no glowing reporter protein is being produced, so the cell is dim.

As enzymes in the cell break protein A down, the cell begins to glow again.

BETSY HAYES

The "ticking" of this bacterial clock is a visual phenomenon—the bacterium glows and dims as the production of a fluorescent "reporter" protein is turned on and off. The gene for this reporter protein is controlled by three other genes—A, B and C—that work together in a cycle: Protein A (encoded by gene A) represses gene B, protein B represses gene C, and protein C represses gene A. Since protein A also turns off the reporter gene, this cycle ultimately controls the timing of the clock.

change in temperature." Since one of the two genes produces a protein that fluoresces under laser light, the researchers can use a laser-based detector to see when a cell toggles between states.

Within a few months, James Collins hopes to launch a **biocomputing company.** Its early products might detect toxins or food contamination.

In January, in the journal *Nature*, Gardner, Collins and Cantor described five such flip-flops that Gardner had built and inserted into *E. coli*. Gardner says that the flip-flop is the first of a series of so-called "genetic applets" he hopes to create. The term "applet" is borrowed from contemporary computer science: It refers to a small program, usually written in the Java programming language, which is put on a Web page and performs a specific function. Just as applets can theoretically be combined into a full-fledged program, Gardner believes he can build an array of combinable genetic parts and use them to program cells to perform new functions. In the insulin-delivery example, a genetic applet that sensed the amount of glucose in a diabetic's bloodstream could be connected to a second applet that controlled the synthesis of insulin. A third applet might enable the system to respond to external events, allowing, for example, a physician to trigger insulin production manually.

GeneTic Tock

As a graduate student at Princeton University, Rockefeller's Michael Elowitz constructed a genetic applet of his own—a clock.

In the world of digital computers, the clock is one of the most fundamental components. Clocks don't tell time—instead, they send out a train of pulses that are used to synchronize all the events taking place inside the machine. The first IBM PC had a clock that ticked 4.77 million times each second; today's top-of-the-line Pentium III computers have clocks that tick 800 million times a second. Elowitz's clock, by contrast, cycles once every 150 minutes or so.

The biological clock consists of four genes engineered into a bacterium *(see "A Clock in a Cell," p. 72)*. Three of them work together to turn the fourth, which encodes for a fluorescent protein, on and off—Elowitz calls this a "genetic circuit."

Although Elowitz's clock is a remarkable achievement, it doesn't keep great time—the span between tick and tock ranges anywhere from 120 minutes to 200 minutes. And with each clock running separately in each of many bacteria, coordination is a problem: Watch one bacterium under a microscope and you'll see regular intervals of glowing and dimness as the gene for the fluorescent protein is turned on and off, but put a mass of the bacteria together and they will all be out of sync.

Elowitz hopes to learn from this tumult. "This was our first attempt," he says. "What we found is that the clock we built is very noisy—there is a lot of variability. A big question is what the origin of that noise is and how one could circumvent it. And how, in fact, real circuits that are produced by evolution are able to circumvent that noise."

While Elowitz works to improve his timing, B.U.'s Collins and Gardner are aiming to beat the corporate clock. They've filed for patents on the genetic flip-flop, and Collins is speaking with potential investors, working to form what would be the first biocomputing company. He hopes to have funding in place and the venture launched within a few months.

The prospective firm's early products might include a device that could detect food contamination or toxins used in chemical or biological warfare. This would be possible, Collins says, "if we could couple cells with chips and use them—external to the body—as sensing elements." By keeping the modified cells outside of the human body, the startup would skirt many Food and Drug Administration regulatory issues and possibly have a product on the market within a few years. But Collins' eventual goal is gene therapy—placing networks of genetic applets into a human host to treat such diseases as hemophilia or anemia.

Another possibility would be to use genetic switches to control biological reactors—which is where Knight's vision of a bridge to the chemical world comes in. "Larger chemical companies like DuPont are moving towards technologies where they can use cells as chemical factories to produce proteins," says Collins. "What you can do with these control circuits is to regulate the expression of different genes to produce

Addressing the Problem

Before they can turn living organisms into computational systems, biocomputing researchers need a way to create and connect multiple "circuits"—switches, clocks and so forth—within a single cell. That's the thrust of the research of Adam Arkin, who works at the University of California, Berkeley, and the nearby Lawrence Berkeley National Laboratory.

"The idea is to make it a generic type of switch," explains Arkin, "where the wires to the inputs and outputs are easily modified genetically, where there is component standardization—meaning we can reuse components within the same circuitry."

But reusing components within a silicon chip is a simpler proposition than reusing circuits within a living cell: A hardware engineer can arrange wires and silicon traces so that a particular circuit is directly connected to another circuit. Inside a cell, however, the genes and proteins float in a molecular soup where they can potentially be affected by just about every other gene and protein in the cell. To get around this problem, Arkin is developing techniques for identifying different circuits inside the cell. If each circuit is given a different address—sort of like an apartment number in a large condominium complex—multiple copies of the same switch could be put inside a single bacterium and activated separately.

A biologist by training, Arkin specializes in modeling complex biological systems from an engineering perspective. "I've tried to be as…rigorous as possible," he says, down to the point of knowing where every atom in a system is. It's this degree of modeling that will be necessary to make the biological-based computing actually work.

Forward engineers: Tim Gardner (left) and James Collins want to "program" cells at will.

your proteins of interest." Bacteria in a large bioreactor could be programmed to make different kinds of drugs, nutrients, vitamins—or even pesticides. Essentially, this would allow an entire factory to be retooled by throwing a single genetic switch.

Amorphous Computing

Two-gene switches aren't exactly new to biology, says Roger Brent, associate director of research at the Molecular Sciences Institute in Berkeley, Calif., a nonprofit research firm. Brent—who evaluated biocomputing research for the Defense Advanced Research Projects Agency—says that genetic engineers "have made and used such switches of increasing sophistication since the 1970s. We biologists have tons and tons of cells that exist in two states" and change depending on external inputs.

For Brent, what's most intriguing about the B.U. researchers' genetic switch is that it could be just the beginning. "We have two-state cells. What about four-state cells? Is there some good there?" he asks. "Let's say that you could get a cell that existed in a large number of independent states and there were things happening inside the cell...which caused the cell to go from one state to another in response to different influences," Brent continues. "Can you perform any meaningful computation? If you had 16 states in a cell and the ability to have the cell communicate with its neighbors, could you do anything with that?"

By itself, a single cell with 16 states couldn't do much. But combine a billion of these cells and you suddenly have a system with 2 gigabytes of storage. A teaspoon of programmable bacteria could potentially have a million times more memory than today's largest computers—and potentially billions upon billions of processors. But how would you possibly program such a machine?

Programming is the question that the Amorphous Computing project at MIT is trying to answer. The project's goal is to develop techniques for building self-assembling systems. Such techniques could allow bacteria in a teaspoon to find their neighbors, organize into a massive parallel-processing computer and set about solving a computationally intensive problem—like cracking an encryption key, factoring a large number or perhaps even predicting weather.

Researchers at MIT have long been interested in methods of computing that employ many small computers, rather than one super-fast one. Such an approach is appealing because it could give computing a boost over the wall that many believe the silicon microprocessor evolution will soon hit *(see "The End of Moore's Law?" p. 42)*. When processors can be shrunk no further, these researchers insist, the only way to achieve faster computation will be by using multiple computers in concert. Many artificial intelligence researchers also believe that it will only be possible to achieve true machine intelligence by using millions of small, connected processors—essentially modeling the connections of neurons in the human brain.

The DNA-based prototypes biocomputing researchers build today may be steppingstones to computers based on neurochemistry, Roger Brent says.

On a wall outside of MIT computer science and engineering professor Harold Abelson's fourth-floor office is one of the first tangible results of the Amorphous Computing effort. Called "Gunk," it is a tangle of wires, a colony of single-board computers, each one randomly connected with three other machines in the colony. Each computer has a flashing red light; the goal of the colony is to synchronize the lights so that they flash in unison. The colony is robust in a way traditional computers are not: You can turn off any single computer or rewire its connection without changing the behavior of the overall system. But though mesmerizing to watch, the colony doesn't engage in any fundamentally important computations.

Five floors above Abelson's office, in Knight's biology lab, researchers are launching a more extensive foray into the world of amorphous computation: Knight's students are developing techniques for exchanging data between cells, and between cells and larger-scale computers, since communication between components is a fundamental requirement of an amorphous system. While Collins' group at B.U. is using heat and chemicals to send instructions to their switches, the Knight lab is working on a communications system based on bioluminescence—light produced by living cells.

To date, work has been slow. The lab is new and, as the water-purity experience showed, the team is inexperienced in matters of biology. But some of the slowness is also intentional: The researchers want to become as familiar as possible with the biological tools they're using in order to maximize their command of any system they eventually develop. "If you are actually going to build something that you want to control—if we have this digital circuit that we expect to have somewhat reliable behavior—then you need to understand the components," says graduate student Ron Weiss. And biology is fraught with fluctuation, Weiss points out. The precise amount of a particular protein a bacterial cell produces depends not only on the bacterial strain and the DNA sequence engineered into the cell, but also on environmental conditions such as nutrition and timing. Remarks Weiss: "The number of variables that exist is tremendous."

To get a handle on all those variables, the Knight team is starting with in-depth characterizations of a few different genes for luciferase, an enzyme that allows fireflies and other luminescent organisms to produce light. Understanding the light-generation end of things is an obvious first step toward a reliable means of cell-to-cell communication. "There are cells out there that can detect light," says Knight. "This might be a way for cells to signal to one another." What's more, he says, "if these cells knew where they were, and were running as an organized ensemble, you could use this as a way of displaying a pattern." Ultimately, Knight's team hopes that vast ensembles of communicating cells could both perform meaningful computations and have the resiliency of Abelson's Gunk—or the human brain.

Full Speed Ahead

Even as his lab—and his field—takes its first steps, Knight is looking to the future. He says he isn't concerned about the ridiculously slow speed of today's genetic approaches to biocomputing. He and other researchers started with DNA-based systems, Knight says, because genetic engineering is relatively well understood. "You start with the easy systems and move to the hard systems."

And there are plenty of biological systems—including systems based on nerve cells, such as our own brains—that operate faster than it's possible to turn genes on and off, Knight says. A neuron can respond to an external stimulus, for example, in a matter of milliseconds. The downside, says Knight, is that some of the faster biological mechanisms aren't currently understood as well as genetic functions are, and so "are substantially more difficult to manipulate and mix and match."

Still, the Molecular Sciences Institute's Brent believes that today's DNA-based biocomputer prototypes are steppingstones to computers based on neurochemistry. "Thirty years from now we will be using our knowledge of developmental neurobiology to grow appropriate circuits that will be made out of nerve cells and will process information like crazy," Brent predicts. Meanwhile, pioneers like Knight, Collins, Gardner and Elowitz will continue to produce new devices unlike anything that ever came out of a microprocessor factory, and to lay the foundations for a new era of computing.

Who's Who in Biocomputing

ORGANIZATION	KEY RESEARCHER	FOCUS
Lawrence Berkeley National Laboratory	Adam Arkin	Genetic circuits and circuit addressing
Boston University	James J. Collins	Genetic applets
Rockefeller University	Michael Elowitz	Genetic circuits
MIT	Thomas F. Knight	Amorphous computing