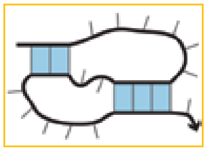
PROGRAMMING MOLECULAR APPS

By Marcus Y. Woo

By learning how to program molecules to do everything from assembling a nanorobot to fighting cancer, we may be embarking on the next technological revolution.



Computers haven't taken over—yet. There are no Hals or homicidal cyborgs on Harleys, but computers still pervade every facet of our daily lives. As adding machines and vacuum tubes gave way to electronic transistors in the 1940s and 1950s, and as room-sized computers gave way to laptops several decades later, computer technology has been evolving faster than anyone ever anticipated. From cell phones to coffee makers, nearly every electronic device is now equipped with a silicon chip.

These chips not only allow people to program their latest smart phone, they can embed autonomous decision-making—some might even say intelligence—into what's otherwise a dumb machine. "When you attach a computer to something, it becomes much more powerful," says computer scientist Paul Rothemund (BS '94). But while the computer revolution continues, Rothemund is among a group of scientists and engineers who think we're on the verge of yet another revolution.

That next step? *Molecular* programming. Instead of telling electrons how to call Mom, brew an espresso, or solve a complex equation, these researchers hope to tell molecules how to diagnose diabetes, assemble into a nanobot, or attack a cancer cell. Such molecules might not only seek out the cancerous cell, but based on their evaluation of the cell type, its environment, and the cancer's state of progression, they would release the appropriate drug at the proper dose and time.

"It's hard to tell where things are going to go," says engineer Richard Murray (BS '85), who along with Rothemund is part of Caltech's Molecular Programming Project, or MPP. "But I suspect we'll use molecular programming the way we now think of electronics." Like computer programming, molecular programming is an engineering endeavor, he adds. If the field advances as rapidly as Murray, Rothemund, and their colleagues hope it will, in a few decades molecular programming could be as ubiquitous as the electronic kind, changing not just how we live, but how we understand the world and life itself.

FROM BINARY TO BASIC

Fortunately, a powerful computing language already exists in the form of DNA molecules, which encode all the information any organism needs to develop, grow, and reproduce, whether it's a bacterium, an elephant, or a towering redwood. This information is spelled out by sequences of four chemical "bases" (commonly called A, T, C, and G) that act as the letters of the DNA alphabet. The letters follow strict rules—A is paired with T, and C with G—so the sequence of letters in a strand determines the sequence of letters in the strand that will bind to it. Couple this intrinsic logic to the ability to write any sequence of letters you want into a DNA strand—now a routine part of bioengineering—and you're on your way to writing a molecular program.

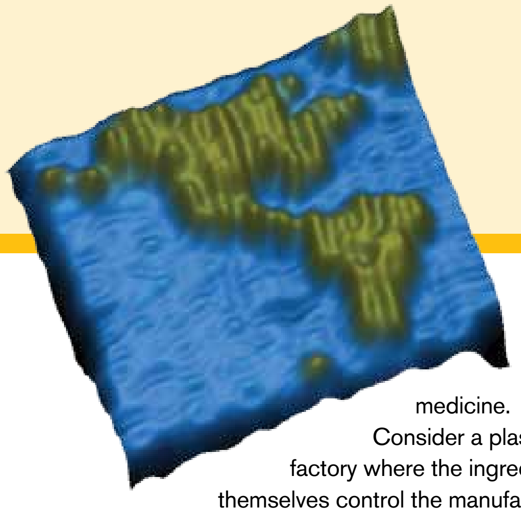
DNA and its cousin, RNA, are easily made in labs and are integral to biology. If you want to learn how proteins function, or inject molecules into the body to combat cancer, then it only makes sense to consider DNA.

The potential for molecular programming, however, goes far beyond

Left: *City of Life*, by Ann Erpino (http://www.annerpino.com/). Copyright 2006. Reprinted with permission.

To the left: This map of North and South America, made using DNA origami, is only about 100 nanometers wide. The image was taken using an atomic-force microscope, which works by dragging a needle across the object, measuring the bumps of every atom.

To the right: Woo and Rothemund have developed a jigsaw-puzzle-like way to assemble DNA tiles. Here, four tiles bearing the raised letters A, B, C, and D bind only if the shapes of their edges match. Top: a diagram of the scheme. Bottom: an atomic-force microscope image of the actual tiles.



medicine.

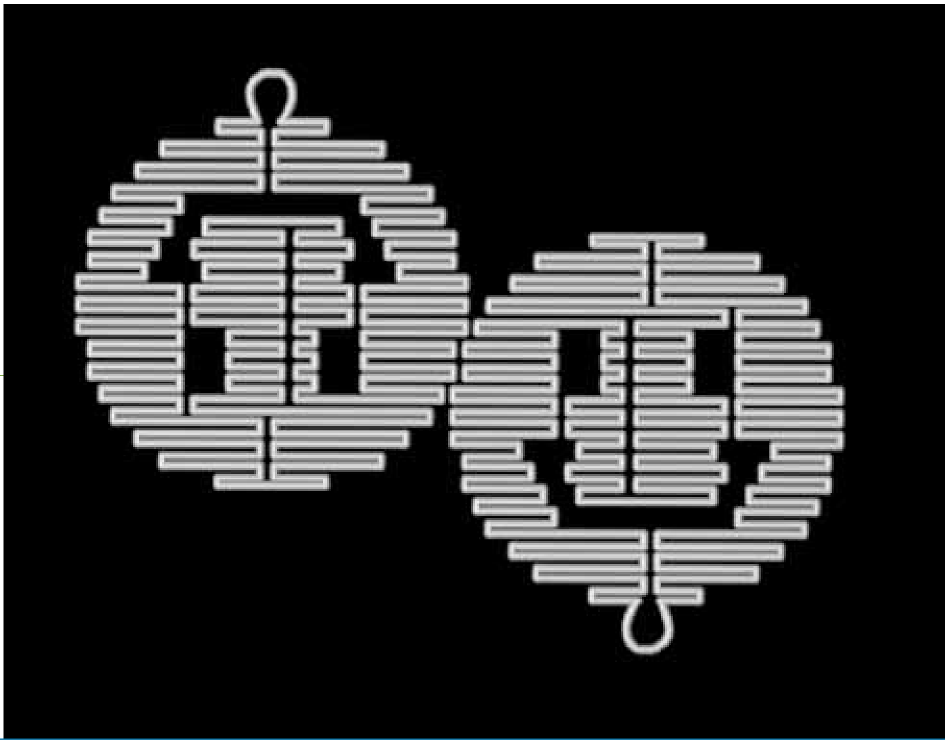
Consider a plastics factory where the ingredients themselves control the manufacturing process through built-in feedback loops, rather than relying on humans (or even computers) to mix the right amounts of chemicals A and B to get the maximum production of C. Programmable molecules could assemble themselves into entirely new kinds of composite materials, or even complicated structures. Maybe one day our cell phones (or whatever communication devices we'll have in the future) will be *grown*, molecule by molecule. Perhaps more fundamentally, molecular programming is already providing tools for studying biology like never before.

The MPP researchers envision a world where molecular programming is accessible to all regardless of technical training. Just as anyone can teach themselves how to write software or design a web page, someone in the future might just as easily write a molecular program. "In a hundred years, someone who has no clue about biology might make a huge contribution to medicine by writing something on the level of an iPhone app," says information scientist Shuki Bruck, another member of the MPP team.

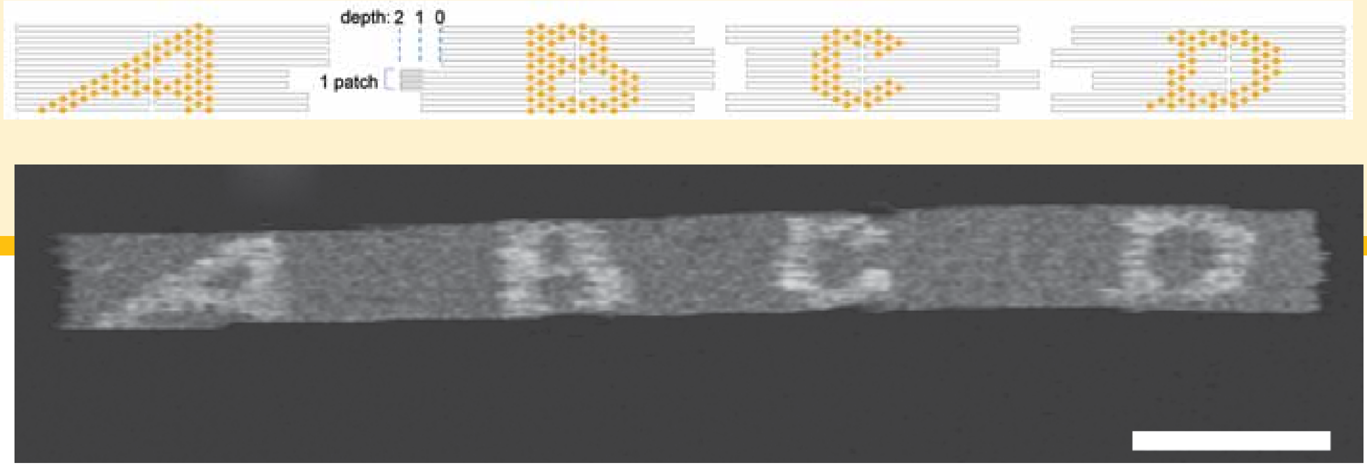
"The difference between the MPP and what other people are doing with nanotechnology and biotechnology is that we're trying to think in terms of information science," Murray explains. "What are the different levels of abstraction we can use to describe the system?" For example, a specific combination of DNA strands that behaves in a certain way can be thought of as an independent component—a black box with a well-defined function. Engineers can mix and match these boxes to create larger, more complex components that will eventually lead to a level of abstraction on a par with the high-level programming languages we now have in computer science. "If you sit down and use Microsoft Word, you don't have to think about what the individual transistors are doing," Murray says. "You think on the level of writing a macro to do something you want to do."

The key is to be able to program molecules without having to wade knee-deep into the intricacies of molecular biology. Writing out the strings of bases needed for individual DNA strands to perform specific tasks is the equivalent of hand-coding applets in binary. The MPP is creating the molecular equivalent of machine language—developing the first low-level assembly languages and the compilers that read them—that will enable the next generation of molecular software developers to write the equivalent of BASIC and FORTRAN.

The MPP "is a quintessential engineering activity, in that we're trying to understand how you design things in a systematic way," says Murray, whose expertise is in feedback and control systems. Since feedback is also crucial to biology—it's how your body regulates your blood-sugar levels, for instance—he's applying control-system principles to molecular programming. In fact, Eric Klavins of the University of Washington, another member of the MPP and a former postdoc of Murray's, has built networks of genes that behave in a homeostatic manner. In other words, just as the amount of glucose in your blood remains within certain limits, regardless of whether you're lifting weights or watching TV, these networks adjust the production rate of a given substance depending on how much of it is being consumed.



Each sheet of DNA origami "paper" is a closed loop of DNA called a plasmid. Any two-dimensional shape you like can be made by folding the loop back and forth on itself.



Woo and Rothemund, *Nature Chemistry*, vol. 3, pp. 620–627. Published online July 10, 2011. Copyright © 2011, Nature Publishing Group.

FROM ORIGAMI TO NANOBOTS

Paul Rothemund is perhaps best known for his tiny smiley faces. In 2006, he pioneered DNA origami—a method for folding strands of DNA into any two-dimensional shape, including smiley faces just 100 nanometers across and 2 nanometers thick. (He has also made a DNA map of North and South America, spelled out the letters "DNA" in DNA, and formed DNA snowflakes.) Each shape consists of a long strand of DNA folded at intervals so that it runs back and forth in a series of parallel line segments that fill in the shape. The lines are held in place by "staples," about 200 of them—short strands of DNA that bind to the long strand as it folds, guiding the adjoining segments into their proper positions. Researchers around the world now use DNA origami to build everything from tiny boxes to nanoscale transistors.

Just as the wingspan of a folded crane is limited by the size of the sheet of paper, the size of a single piece of DNA origami is limited by the length of the strand being folded. Strands longer than the ones Rothemund is using have proven hard to come by, so over the last two years, Rothemund and grad student Sungwook Woo have worked out new techniques for assembling origami building blocks into larger structures. The standard high-school biology view of a DNA molecule is as a twisted ladder, with the rungs consisting of pairs of bases that cling to each other. However, each rung also sticks strongly to the rungs above and below it through a so-called stacking interaction—in fact, this interaction appears to be the main force holding the ladder together.

If you put a hinge between two rungs and fold the ladder in half, the rungs next to the hinge will be "hanging in the breeze, without a partner to stick to on one side," says Rothemund. This blunt end, as it's called, will readily stick to any other blunt end. "You can visualize the blunt end as the flat, sawed-off end of a log," Rothemund continues, noting that the origami blocks—or, more accurately, two-dimensional tiles—look like tiny log-cabin walls. By including logs that ended in floppy, nonadhesive loops of DNA as well as sticky blunt ends, the researchers created a binary system: "0" for non-sticky and "1" for sticky. Woo and Rothemund demonstrated that these tiles bound preferentially to other tiles whose edges encoded the same binary sequence, meaning that such tiles could be strung together in any order one might care to program. "A lot of people, including us, have made DNA shapes with blunt ends that cause the shapes to stick together into random clumps—piles of junk," Rothemund says. "But if you are careful, you can harness the power of blunt-end stacking interactions—converting something that used to be considered a 'bug' into a 'feature.'"

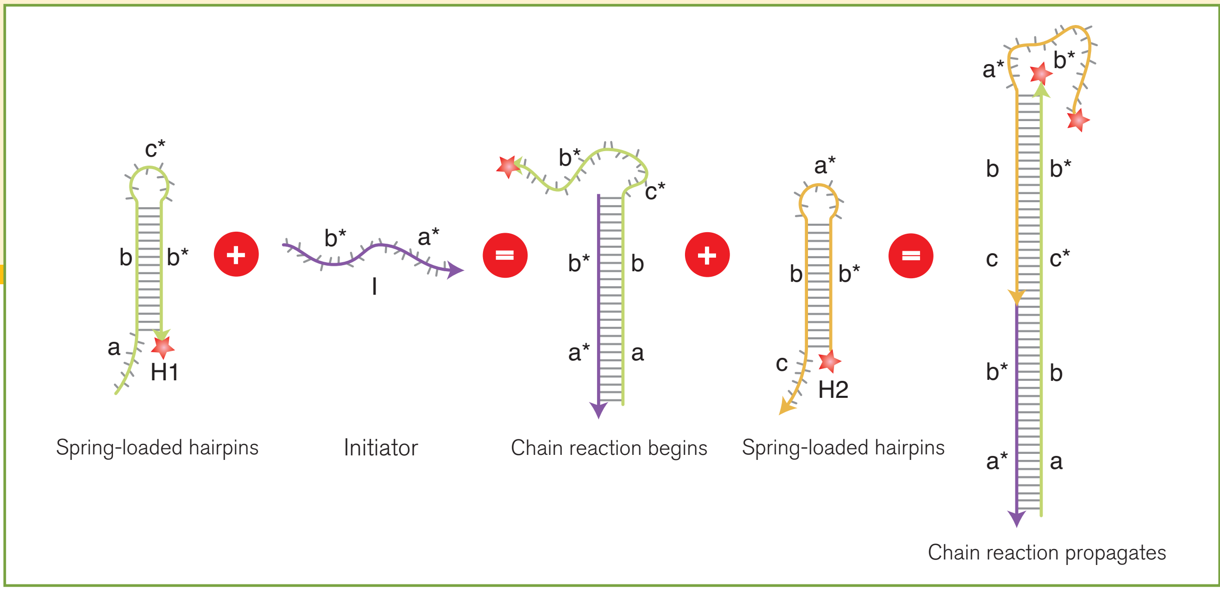
The binary-sequence tiles were rectangular, with straight edges. But Woo and Rothemund also created a set of shape-recognizing tiles with jagged edges: all of the logs had sticky ends, but the logs themselves were of varying lengths. And behold, properly matched edges clicked into place like the pieces of a jigsaw puzzle.

This particular set of jigsaw tiles had 16 possible edge shapes; 16-bit binary tiles would have thousands of easily distinguishable edge patterns. But merely having 16 possible edges to play with would enable people to make much more complex devices than can be made through origami, including simple logic circuits rather than the single transistors that have been made so far.

In addition to enabling larger structures, these techniques could ease us past a sticking point on the way to a full-on, sci-fi, self-assembling nanobot: the problem of moving parts. "A human-scale analogy would be to take all the parts for a car, paint them with glue, throw them in a bag, shake it, and have a working car pop out," Rothemund says. But DNA base-pair binding makes for a very powerful molecular adhesive—your car's motor wouldn't run and the wheels wouldn't turn. Even the wipers would be stuck to the windshield. "We think that we will be able to design stacking bonds in which the parts of a nanomachine will be able to self-assemble and then slide freely past each other," he continues. "The parts won't look exactly like interlocking log walls, but they will work on the same basic principle."

FROM NANOBOTS TO CELL PHONES

In order to make these parts self-assemble, however, each one has to know exactly



where it's going and how it fits into the grand design. And this is a problem: say you had several thousand unique shapes at your disposal—that's still not enough to orchestrate the spontaneous coalescence of a piece of human-scale technology.

"If we wanted to self-assemble a cell phone, origami won't do it," says Erik Winfree (PhD '98), a computer scientist and director of the MPP. "But algorithmic processes could." For the last 15 years, first as a grad student and now as a faculty member, Winfree has been working on ways to embed algorithms, the abstract ideas at the heart of computer programs, into the DNA itself. "You design a set of molecules that fit together according to a certain logic, and by controlling which molecule fits at which location, you can program the growth of a whole structure," he explains. Then all you need to do is pour the molecules into a beaker, stir briskly, and voilà! The algorithm executes itself. If you are clever enough writing the rules, you can create very complex structures with just a few different molecular bricks.

But why work molecule by molecule? "To build things that are more structured than current technology allows," Winfree says. "If you can build things out of bricks that are a few nanometers in size, you can do a lot more in the same space." Compare a whale and a submarine, he explains. Both are about the same size, and both propel themselves underwater. A submarine is made up of relatively large pieces—steel panels and pipes, screws and bolts. But a whale's internal structure extends all the way down to the arrangement of the individual protein molecules in its cells. In other words, a chunk of whale has a lot more going on than an equal-sized chunk of submarine.

Another advantage of molecular self-assembly is the potential for cheap manufacturing. It costs billions of dollars to build a factory that makes silicon chips. "Biology has always been the opposite of that," Winfree says. "If you have a few seeds, they'll just grow. Mold will grow in your refrigerator even though you didn't want it to." DNA self-assembly mimics how biological organisms grow, cell by cell, protein by protein, and as a result, it's much cheaper than conventional manufacturing. Using molecular programming to "grow" a cell phone would be more akin to kitchen chemistry than to an expensive, factory-based process, he says. (He also adds that we're a long way from growing anything as big and complicated as a cell phone.) "Our lab," Winfree says, "is very simple. Most of our procedures involve ordering a few DNA strands, mixing them together, and letting the molecules do the hard work."

FROM CELL PHONES TO THE BRAIN

"In a really fundamental sense, algorithmic self-assembly *is* a form of computation," says Rothemund. "In fact, it is far *more* powerful than circuits." But these hardworking molecules can also be made to "compute" in the traditional fashion: Winfree and his colleagues are building DNA circuits, replacing handfuls of transistors with test tubes full of molecules. In a digital logic gate, electrons either flow or they don't. In a DNA-based logic gate, the DNA strands either bind or they don't. Early this year, Winfree and postdoc Lulu Qian created, from scratch, the largest and most complex DNA circuit ever made. It used a set of standardized components—a crucial requirement for developing higher-level molecular programming languages, as well as for scaling up the circuits themselves.

This circuit consists of 74 kinds of DNA molecules, and can calculate the square root of any integer up to 15—or, for the technically inclined, any four-bit number. (The circuit does round the result down to the nearest whole number, however, as dealing with a decimal point is a bit beyond its capacity at the moment.)

The original dream for DNA computing was to solve big, complex problems. "That hasn't panned out," Winfree says. DNA computing just isn't efficient enough; calculating that square root took 10 hours. "Nevertheless, a tiny bit of computing goes a long way in the molecular world," remarks Rothemund. "Molecular computers, no matter how simple, can be used to control other molecular phenomena." No silicon computer has this power, but it's DNA's natural role. "The whole process of embryonic development is controlled by a molecular computer performing logical operations—'If *X* then *Y*, but only if *A* hasn't happened.' And even though

A single strand of RNA can fold onto itself to form a hairpin molecule. A fluorescent molecule, labeled H1, is attached to such a hairpin. When an RNA initiator comes along, it binds to the hairpin and pops it open. The resulting molecule attaches to yet another hairpin, which carries another fluorescent molecule (H2). The chain reaction continues, producing a long molecule with many glowing attachments to form a really bright marker.

it doesn't solve anything we'd recognize as a computationally difficult problem, its computation serves to make a very complicated object."

Still, if doing sixth-grade math isn't enough for you, Qian, Winfree, and Shuki Bruck have created the first-ever DNA circuit that has *brainlike* behavior. That's right—using the same methods employed to design the square-root calculator, they have made a neural network that plays a mind-reading game. To play, you first think of a scientist. Then you answer one or more of four previously defined yes-or-no questions—for example, "Was the scientist British?"—by dropping the DNA strands corresponding to those answers into the test tube. Now you shake the test tube vigorously, and if the facts you provided match any of the four scientists programmed into the network's memory, the test tube lights up with a color-coded fluorescent signal. The network also tells you if it doesn't know the answer. It even tells you *why* it doesn't know: whether the scientist you picked is not in its memory, or whether you didn't give it enough clues for it to narrow its choice to just one person.

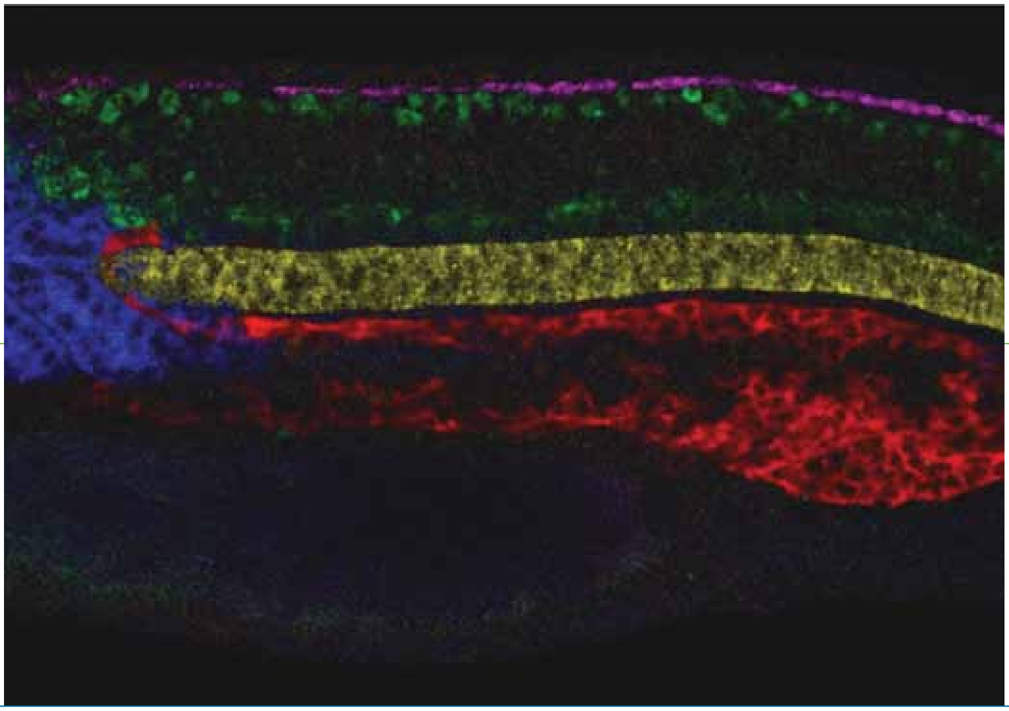
Taking an incomplete pattern and figuring out what it might mean has been one of the hallmarks of a living brain. Maybe it's not just silicon chips that we have to worry about becoming self-aware and taking over the world. We might have to watch out for those molecules, too.

FROM THE BRAIN TO THE ORGANISM

While his colleagues were concerned with computational power, bioengineer Niles Pierce was all about motive power. In 2004, he and grad student Robert Dirks (PhD '05) came up with a way to make DNA "fuel" in the form of spring-loaded hairpins of single-stranded DNA that could pop open on cue. Four years later, Pierce and postdoc Peng Yin demonstrated that a properly scripted set of cues can nudge molecular machines into performing a surprising variety of feats. One of their creations was a molecular "tree" that grew dendritically from seed to leaf; another was a DNA "walker" that strolled along a DNA track. "In theory, DNA motors could provide us a different way to build things," says Pierce. "Rather than just letting things stick together, if pieces can be moved around *actively*, by DNA walkers, it may be possible to build complex objects much more quickly and efficiently."

Meanwhile, Pierce has begun applying what he had learned from engineering molecular machinery to developing molecular instruments—tools he hopes will revolutionize biological research. "There are profound questions hanging over the heads of biologists about how development works, about how diseases work," he says. "Biologists will be able to make faster progress with more powerful experimental techniques."

Answering many of these questions requires finding out when and where genes are switched on in different cells. For the last 40 years, biologists have been using a method called in situ hybridization, or ISH for short, to pinpoint the locations of messenger RNA (mRNA) molecules that serve as proxies for their activated genes. (RNA, the chemical first cousin to DNA, encodes information in an almost identical set of bases, and mRNAs deliver the DNA's protein-making instructions to the cell's protein factories, the ribosomes.) After flooding a thinly sliced tissue sample or a Petri dish full of cells with an RNA probe designed to bind to the mRNA of interest, the unbound probe molecules are rinsed away. A fluorescent "tag" on the probe then lets you image the mRNA targets under a microscope.



This image shows the fluorescent amplification technique being used in a zebrafish embryo, illuminating five different kinds of mRNA with five different colors.

To the right: A “smart drug” might one day consist of a molecular robot that recognizes malignant cells by their surface proteins. After docking with the marker protein, the robot could crawl along the cell’s membrane, slicing it open and destroying the cell.

But when you try ISH on a vertebrate embryo, which are the ones of most interest to scientists studying human development and diseases, you need to boost the fluorescence in order to see it. Biologists add an enzyme that deposits extra tag molecules near the probe, amplifying the fluorescence. To see two mRNAs at once, you repeat the process with the same enzyme and a different dye.

This is a big problem, Pierce says. “It’s cumbersome, and an impediment to studying genetic circuits, where biologists want to look at many genes interacting with each other.” With the standard approach—first dyeing one mRNA red, say, and then another one green—it takes about five days to do three colors.

But if the fluorescent amplifiers were programmable, they could operate independently, seeking out their various mRNAs at the same time. Pierce realized that the spring-loaded hairpins that powered the walker could run an amplifier as well. With each type of hairpin carrying a different dye molecule, the only limit to the number of colors would be the number of dyes that can be distinguished through the microscope.

After five years of hard work, Pierce and his coworkers, including grad student Harry Choi (PhD ’10) and biology professor Scott Fraser, successfully tested fluorescent ISH, or FISH—on, appropriately enough, zebrafish embryos—by targeting five species of mRNA simultaneously. The method passed with, well, flying colors. “These amplifiers, which started as a proof-of-principle exercise in molecular programming, are now a research tool at Caltech,” Pierce says; while Fraser continues with the zebrafish studies, biologists Marianne Bronner, Dianne Newman, and Eric Davidson are using the technique to study genetic circuits in birds, bacteria, and sea urchins respectively. And the method is catching on elsewhere—the Pierce lab is providing probes and hairpins, as well as technical support, to biologists around the world.

While the researchers are continuing to enhance this technique—for example, figuring out how to zoom in and achieve molecule-scale resolution in order to map mRNA locations quantitatively—they’re also pondering other types of molecular instruments. One possibility, Pierce says, is to design tools that would turn gene B on (or off) depending on whether gene A is already on or off. “This would provide unprecedented tools for studying genetic circuits at specific times and locations within developing embryos,” he says. Being able to program conditional gene activation (or silencing) would also have medical potential, he adds.

“These are all dreams right now,” Pierce says. But he hopes that in the next 10 to 15 years molecular instruments will become indispensible for research. “The possibilities,” he says, “are essentially endless.”

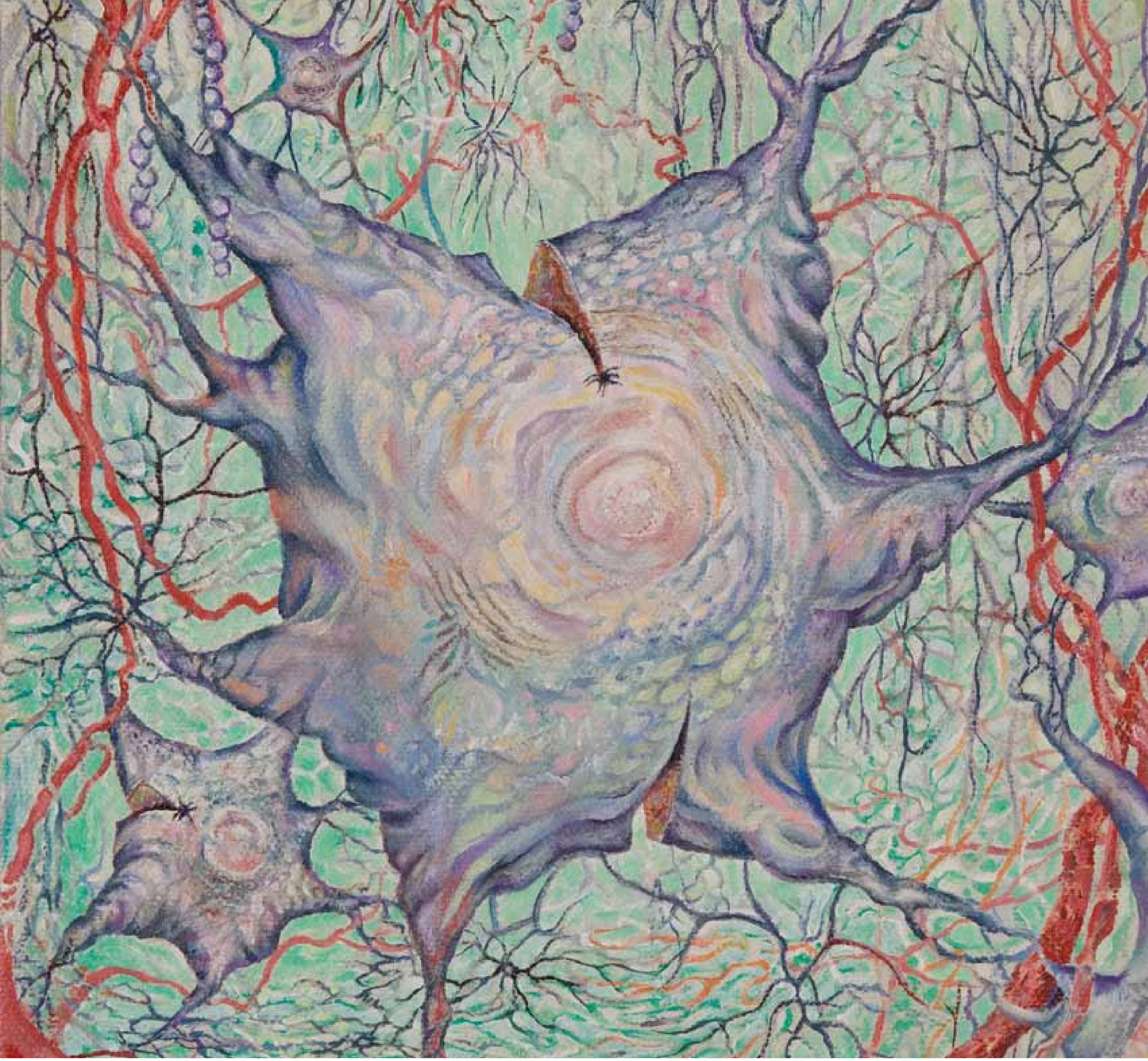
FROM THE ORGANISM BACK TO BASICS

Although the MPP is informed by the computer revolution, it’s rooted in aspirations to understand how life works. “Understanding biological systems is the most important challenge for the next 100 years,” says Bruck, whose background is electrical engineering, but who now focuses his research on computing with biological circuits. “If you compare our world to anything else in the universe, based on what we know so far, we have life here and not anywhere else. That’s the most precious thing we have here, and we still don’t understand it.”

For a computer scientist like Winfree, the MPP is about the idea that information is the essence of nature, that life is driven by the programming power of DNA. “The universe just happens to be that way,” he says. “Biology has exploited that inherent essence of nature to do what biology wants to do: to reproduce, to evolve, to build really complex animals, and to build brains.” The ambitions behind molecular programming go beyond a descriptive understanding of biology, as researchers strive for deeper insight into what life is at its most fundamental level. “Technological developments have historically led to new concepts, and the languages needed to express them,” says Winfree. “And these new languages change how we look at the world and reason about it.”

Just as miners and engineers tinkering with pumps led to the development of the steam engine, which in turn led to the discovery of the laws of thermodynamics and eventually to the development of statistical mechanics—which is now used to analyze everything from galactic evolution to stock-market rallies—the languages that might spring from molecular programming could give us the conceptual tools needed to think about complex biological systems in a whole new way.

The Industrial Revolution and the Computer Age came about by accident and happenstance, but Rothemund hopes that the Molecular Era will occur by design. “The fact that we have started to recognize the features of such revolutions gives society tolerance for us to play around and see if we can build another one,” he says. “This kind of forethought is, I think, a hallmark of our age. Research grants and start-up companies have regularized, ritualized, and mechanized innovation. The MPP, for example, is supported by a very forward-looking program run by the National Science Foundation called Expeditions in Computing. It’s really amazing to me that we can come up



Process of Elimination, by Ann Erpino (http://www.annerpino.com/). Copyright 2007. Reprinted with permission.

with a futuristic, almost science-fiction vision, organize around it, and have society buy into it."

***Paul Rothemund (BS '94)* is a senior research associate in bioengineering, computer science, and computational and neural systems. Richard Murray (BS '85) is the Everhart Professor of Control and Dynamical Systems and Bioengineering. Jehoshua "Shuki" Bruck is the Moore Professor of Computation and Neural Systems and Electrical Engineering. Erik Winfree (PhD '98) is a professor of computer science, computation and neural systems, and bioengineering. Niles Pierce is a professor of applied and computational mathematics and bioengineering and executive officer for bioengineering.**

Besides the National Science Foundation's Expeditions in Computing Program, various parts of this work were supported by the National Institutes of Health, the International Human Frontier Science Program, the Ralph M. Parsons Foundation, the Charles Lee Powell Foundation, the Semiconductor Research Corporation, Microsoft Corporation, Nanorex Corporation, the Benjamin M. Rosen Family Foundation, the Caltech Center for Biological Circuit Design, the Beckman Institute at Caltech, and the Gates Grubstake Foundation at Caltech.