KURZWEIL: There are a couple of characteristics that affect the danger of a new virus; obviously how deadly it is and how easily it spreads. But probably the most important is the stealthiness. SARS spread pretty easily, and it's pretty deadly, but it's not that stealthy, because the incubation period is fairly short. New naturally emerging viruses don't tend to have the worst characteristics on all of these dimensions, so one could, if one were pathologically minded, try to design something at the extreme end of these various spectrums.

I did testify before Congress recently advocating that we greatly accelerate the development of these defensive technologies. It's true that it's not easy to create an engineered biological virus, but the tools and the knowledge and skills to create such a bioengineered pathogen is more widespread than the tools and the knowledge to create, say, an atomic bomb, and could potentially be more dangerous. We are pretty close at hand to some exciting broad-spectrum antiviral techniques. We could apply, for example, RNA interference and other emerging techniques to provide an effective defensive system. It's a race: we want to make sure we have an effective defense when we need it. Unfortunately the political side doesn't get galvanized unless there's some incident. Hopefully we can interest the funding sources before it's needed.

14
Engineering Biology

Drew Endy

[February 17, 2008]

Drew Endy is a professor of bioengineering at Stanford University.

How can I make biology easy to engineer? Going back hundreds of years, people imagined that you could always design and build or make life, but nobody could do that much about it. As the 1970s rolled out, human beings invented a lot of technology: recombinant DNA for cutting and pasting preexisting fragments of genetic material; the polymerase chain reaction, invented in the seventies but not really figured out until the eighties; automatic sequencing, with Fred Sanger in 1977.

Now, thirty years after the initial successes of biotechnology, it has realized only one of the early promises. The early promises were: first, making therapeutics via recombinant organisms, producing drugs like insulin via bacteria, which has worked. Gene therapy was the second promise—fixing genetic defects by patching our DNA, and this has not yet worked. And third, to develop crops that could fix nitrogen, so that agriculture wouldn't have to rely on synthetic fertilizers. That hasn't worked either. So, of the three great early promises that were rolled out with the beginning of genetic engineering, we have realized one of them.

Nevertheless, biotechnology exists. It's a huge positive contributor to our health and economy and the human condition

generally. So the question is, Can we realize the initial promise of biotechnology? Or, forget that question: How do we make biology easy to engineer, so that anything we might want to manufacture out of the living world is something we can pull off?

Imagine you're fifteen or seventeen or eighteen years old. You're an ambitious youngster, and you're showing up as a first-year undergraduate, and you're choosing what to major in. Well, you could choose to major in biology or electrical engineering or computer science or . . . Oh, now you can major in biological engineering! What would you expect to learn? What would you expect of your faculty colleagues, your professors? What would they be able to teach you?

You look to your friends who will study electrical engineering; they can learn how to design and build computers, or write computer programs, and the objects they make don't have emergent properties unless that's what's intended; instead, they behave as expected. Then you look at biological engineering and you say, "Well, yes, I'd like to design and build living organisms, or program DNA to execute genetic programs that behave as expected." But nobody can teach you how to do that.

Thirty years into biotechnology, despite all the successes and attention and hype, we still are inept when it comes to engineering the living world. We haven't scratched the surface, and so the big question for me is, How do we make biology easy to engineer? For comparison, take modern electronics. During and following World War II, people were building computers. John von Neumann was building a nice machine in the basement of the Institute for Advanced Study at Princeton. The official purpose of this machine was to design hydrogen bombs and compute the trajectories of munitions. And he, of course, is apparently running artificial-life programs on it, because that's what he's

more interested in. Let's say it's 1950. The Apple-1, the personal computer is only twenty-five years later.

Will we ever get to the point where biotechnology is not an exclusive technology, not a technology that requires experts? Will we ever get to the point where we can make many-component integrated systems? Will we ever get to the point where we have separation of the types of work in biological engineering, so that one person might be an expert designer, another might be an expert constructor, as we have expert architects and builders and what not?

And a parallel question is, What are the consequences of success? If you look around the room we're in, everything in the room is a synthetic or engineered artifact, right? Even the air we're breathing has been engineered for temperature and humidity. The only thing that hasn't been engineered are the living things—ourselves. Again, what's the consequence of doing that at scale? Biotechnology is thirty years old; it's a young adult. Most of the work is still to come, but how do we actually do it? Let's not talk about it, let's actually go do it, and then let's deal with the consequences, in terms of how this is going to change us; how the biosecurity framework needs to recognize that it's not going to be nation-state-driven work necessarily; how an ownership, sharing, and innovation framework needs to be developed that moves beyond patent-based intellectual property and recognizes that the information defining the genetic material will be more important than the stuff itself and so you might transition away from patents to copyright; and so on and so forth.

So, to zoom out, how to make biology easy to engineer? And how do we do this in a way that leads to constructive culture around the technologies that's overwhelmingly positive in terms of the consequences of its being rolled out?

What happens when the technology in support of engineering biology is sufficiently advanced that somebody like Stefan Sagmeister, the graphic designer, could sit down and design a life form he would consider interesting or beautiful? How do we get from what we've got today, where we're basically celebrating a bunch of stunts and we've delivered only a third of the initial set of promises of biotechnology and there are so many other things we can imagine that are fantastical because they're just too complicated given the current state of affairs? How do we get from that to "Yeah, the graphic designers are making beautiful living objects?"

There is also the issue of addressing energy needs. A lot of people drive investments in biotechnology from the application side, and that's good. There are lots of pressing human needs and problems: Food, which is an energy of sorts for people and animals. Liquid fuels for cars and jets, and then you've got health and medicine, and then you've got environmental issues, and then you've got materials construction, and ta-da-da. What's interesting about biotech is that the applications have always been so unbelievably pressing.

So let's wind the clock back thirty years; to a first approximation, there's been an underinvestment in tools, right? Say you're running a team that's trying to figure out how to make insulin in bacteria, or how to make artemisinic acid for treating malaria in bacteria or yeast, and somebody says, "Hey, why don't you take 5 percent of your project's budget and, instead of spending it on delivering your product as fast as you possibly can, leave behind a little bit of engineering infrastructure, so that the next time you do a project like this it doesn't cost you $40 million? So that the next time you do a project like this, it's much, much easier?" The arguments in response to those sorts of suggestions are: If we delay

shipping our product by a day, we will lose to our competitor, or 10,000 additional people will die, or something. On a short time scale, it's impossible to argue against these positions, right?

But if you take a longer view, absent making such foundational investments in technologies that support the engineering of biology, the engineering of biology will always be hard. We have to figure out how to solve that problem. This comes into play when you think about energy. What do I think about the biological production of energy? Terrific, right? Yeah, we'd rather not be burning dinosaur juice. It seems like important work, and the lab next to mine, when I was earning my PhD, was working on cellulosic ethanol. If the price of oil went up by a factor of 2, cellulosic ethanol would be cost-competitive. That was in 1994.

I hope bioenergy succeeds. But consider that there was a trap for John von Neumann when he was building those early computers to compute the trajectories of munitions. It turns out that the utility of computers is much more than we could have imagined, much more than the military applications and accounting databases. But with a few exceptions, nobody back then had a clue what those applications would be. Thus I'm not interested in pursuing any one application in biotechnology right now, because I want them all to come true, and I want them all to come true on a time scale relevant to me—I can be very direct and selfish about this. And the only way that will ever happen is if I don't go work on bioenergy. There are enough people who'll work on that, because it's a problem everybody can understand; you'll be able to raise resources around it and go do the work. There's the complementary problem, the metalevel problem, which is, Let's make all of biotechnology easier for everybody.

The underlying goal of synthetic biology is to make biology easy to engineer. What does that mean? It means that when I

want to build some new biotechnology, whether it makes food I can eat or a biofuel I can use in my vehicle, or I have some disease I want to cure, I don't want that project to be a research project. I want it to be an engineering project. In the science of biology, the people you're talking to are scientists, they're not engineers, and—not to be arrogant, just to be an observationalist—the question is, If you're an engineer looking at biotechnology, what do you need to do in order to make it easy to engineer? That's what synthetic biology is about.

You could start talking about historical examples of what engineers do when faced with situations of this sort: in America, in 1860, machinists built objects—steam engines, what have you. Nuts and bolts that held together machines were specific to the particular machine shop that manufactured them. What that means is, if you buy a machine from a machine shop in Newark, New Jersey, and it breaks down in Chicago, you have to send it back probably to that specific machine shop, where the machines are set to tool things on a particular set of designs, in order to get the replacement part or get the thing fixed.

In April of 1864, somebody said, "Enough!" William Sellers, of the Franklin Institute in Philadelphia, gave a paper on a system for nuts and bolts. And he proposed the Sellers Screw Thread standard, which is a 60-degree angle squared off at the top screw-thread design, easier to manufacture than the English Whitworth standard of a 55-degree-angle, rounded screw thread. As a result, eventually everybody in the U.S. retooled their machine shop to produce screws, nuts, and bolts in accordance with the Sellers standard. The consequence of this today is that when I go to the hardware store and get a nut and a bolt, so long as they don't screw up the English/metric thing, I can take those two objects and put them together.

That's an example of what an engineer would call reliable physical composition. Take two objects and put them together. The other thing that happens is that when you have the nut and the bolt together as a composite object, when you pull on the nut it stays put. It doesn't come flying off. The composite object has the expected behavior; it doesn't have some emergent property. That's reliable functional composition. The function of the two things when you put them together is what you'd expect. What's amazing is that I've taken this standard for granted my whole life. Even though I have three engineering degrees, I didn't know about this until a couple of years ago, when Tom Knight of MIT pointed out to me that it would be nice if we had standard biological parts that could snap together and behave as expected when we did.

George Church has been exposed to this, but it's not of his mother culture. He's a geneticist; he's reverse-engineering natural biological complexity. That's a great thing to be doing. Engineers hate complexity. I hate emergent properties. I like simplicity. I don't want the plane I take tomorrow to have some emergent property while it's flying. If you look at the science of genetics, which has been in the business of trying to figure out the relevant information in coded DNA, the most important thing has been technology. Before DNA sequencing existed, people would find mutations and they'd map them, in the process, to different regions on the DNA. And the mathematics that was being used was based on simple logic. Then a number of great people drove forward with the sequencing of DNA, and as a result of that technology we can now read DNA, and that technology continues to get better.

It's important to put the impact of advances in DNA-sequencing technology in context. In 1990, nobody had sequenced anything except for a couple of bacterial viruses and maybe some other vi-

ruses. In 1995, the first bacterial genome, *Haemophilus influenzae*, was sequenced. In 2001, there's a draft of the human genome sequence. How did we, in the 1990s, go from stinking at sequencing DNA to "Yeah, we just sequenced human beings," and now, only seven years later, the personal-genome projects coming online? It's not because George Church and Craig Venter and Eric Lander and Francis Collins got 10 billion times smarter during the Clinton years. It's because the technology for sequencing DNA got automated and scaled up sufficiently to do it.

The impact of underlying technologies on what's possible is an important thing to recognize, and something that, if successful, you want to be able to ignore in the same way I want to be able to ignore nuts and bolts and screw-thread standards. Genetics changed in response to sequencing technology. You could read DNA. We have no idea what it says. The mathematics is now pattern recognition, to try and look at many sequences and find the conserved patterns that might have relevant functions. Synthesis technology is coming online next.

2008 is our 1995, if you will. This is the year when a bacterial genome was synthesized from scratch. Ahead of that work, chloroplast genomes, mitochondrial genomes, were constructed; in fact, a project from Japan a couple of years ago made a 10-million-base-pair fragment of DNA from existing fragments, which is fifteen times larger than anything getting attention these days.

So, what happens to the science of genetics as a new set of tools come online that lets us build whatever DNA molecule we want, and you get to make changes and see what happens? Instead of being called genetics, this is called reverse genetics, and the mathematics driving this is probably going to be perturbation design. What changes do you want to make, and how do you choose what to make? First, genetics goes from presequencing

technology and it's based on logic. Then it's postsequencing and it's pattern recognition. And next there's going to be postsynthesis genetics, and it's going to be, "Make whatever you want." Perturbation design becomes the mathematics. And the whole field's going to change.

When sequencing technology was developed, the scientific community—not to mention the rest of the world—did an incredibly poor job of anticipating the resulting challenge of, "What the heck does all this DNA-sequence information mean? How big is the pattern recognition problem?" Fields of science like bioinformatics are purely reactionary and have poorly planned responses to technology advances, and we're going to get the same thing again with synthesis.

For example, how do you manage the information going into a DNA synthesizer so that you can construct some useful object that will help you do genetics? This is the reverse bioinformatics problem. George Church and Craig Venter have a lot to contribute to it, which will be terrific. It will be part of synthetic biology, but it will be synthetic biology impacting science, which is the worst-case scenario for synthetic biology. We fail to actually deliver any useful artifacts that people want, but at least we'll fail, and we'll debug our failures, which will prioritize our misunderstandings of biology much more ruthlessly than anything else, and which is much better than an NIH study section.

What else might happen? I've got an invitation to give a talk at the Chaos Communication Congress, which is the largest hacker meeting in Europe, about 4,000 people—people who like to make stuff, people who like to understand how things work. And they're very interested in learning how to program DNA and how DNA works. One consequence of actually making biology easier to engineer, whether you're standardizing the components

or figuring out how to develop higher-level programming language, is that other people besides the usual suspects are going to have access to the technology.

If you think about what happened between 1950 and 1975, when you went from von Neumann's machine to the Apple-1, a key part of this transition was that folks were so stoked about computing and so fed up with limited access to centralized computing resources that they went out and built their own computers—by definition, the personal computer. As a result, today we have a worldwide community of folks who are excited about building electronics and writing software, which includes school kids, professionals, big companies, small companies, governments, you name it—a diverse ecology around that technology.

Programming DNA is more cool, it's more appealing, it's more powerful than silicon. You have an actual living, reproducing machine; it's nanotechnology that works. It's not some Drexlerian fantasy. And we get to program it. And it's a pretty cheap technology. You don't need a Fab Lab, like you need for silicon wafers. You grow some stuff in sugar water with a little bit of nutrients. My read on the world is that there is tremendous pressure, just starting to be revealed, around what heretofore has been extraordinarily limited access to biotechnology.

Take some of the writings of Freeman Dyson. He's imagining genetic engineers of the future winning the Philadelphia Flower Show, the San Diego Reptile Show, or whatever it is. How do you get there? And as you start working through that path of getting there, what you find is that there are vast communities of people who want to be doing this. But the people promoting the technology tend to favor exclusive ownership and limited access and present themselves as godlike creators—as opposed to "We're constructing things; we could use your help; anything we

do today is going to pale in comparison to what's coming, so let's figure out how to work together on this."

As a different example: In 2003, I taught a course at the Synthetic Biology Lab at MIT with some colleagues, and we had sixteen students. For the last four years, this course has been doubling every year, and it's now taught independently at about sixty schools in thirty or forty countries worldwide. It's called IGEM, the International Genetically Engineered Machines competition. There are teams of teenagers from Germany programming DNA happily there, as well as in Australia, Russia, Japan, China. The competition was won by the team from Peking University this year, and 600 or 700 students participated.

How do you recognize this potential and serve it and bring more people to participate in it? The rewards of doing so are greater than any one group's project. For instance, the team from Melbourne, Australia, showed up with a 6,000-base-pair fragment of DNA they found, which somehow—I don't know how this actually works—folds up; the proteins get made and the proteins self-assemble into a fifty-nanometer sphere filled with gas. The protein shell is somehow gas-impermeable, and these little balloons, these protein balloons, get booted up inside the cytoplasm of cells, and you can control how many different balloons there are. Depending on the number of balloons, the cells will either float or sink or be neutral.

Who knew? I didn't know anything about this biology, and they showed up, they made this standard biological part, such that we can now snap it together with the 2,000 other parts we've got in our collection so far, which is a free collection. We shipped over 100,000 parts around the world last year for free, and the collection's doubling in size every year.

If you make biology easy to engineer, and you make it ac-

cessible, by definition people will learn about it. You can talk to computer-programming conferences about it. It's a very different world from going around claiming you've created life. It's a very different world from going around filing patent applications that say you've invented the idea of a synthetic genome. It's a very different world from spending $40 billion on a classified biological defense facility at the site of the past U.S. Offensive Biological Weapons program. And so there's a cultural mismatch.

The mismatch is largely generational, and it's also largely perspective-driven. By that I mean that the previous generation of people working in biotechnology were scientists, and the ones coming up now are engineers. We'll have to invent our new world of biotechnology, and I suspect we'll learn lessons on biological safety from the past generation, but all the other lessons are up for grabs. The biosecurity framework will collapse. The IT framework based on patents won't scale, and the questions of playing God or not are so superficial and embarrassingly simple that they won't be useful in discussion.

The more serious situation is that these issues of human practice don't get resolved in a six-month conversation. It's not like what happened in Cambridge, Massachusetts, in the seventies, when recombinant DNA work got shut down for a bit and then became OK. The technologies are being developed and distributed so quickly, yet there's still so much more to do in improving the work of biological engineering. The conversations we need to set up are conversations that need to persist in constructive ways for decades.

The open-source world is one thing; if you're trying to invent a language for programming DNA, having a proprietary language seems stupid. If Oxford University had supported privatization of the English language hundreds of years ago, the dictionary they

made wouldn't have been so useful. There will be a core collection of standardized genetic objects that can define families of languages people can use to program DNA. And those have to be made a public resource.

This will be a big transition from today. Biotechnology today derives investments from business models that support the exclusive application of different biological functions for a small number of problems. For example, there are wonderful companies that have locked up most of the relevant intellectual property around how to engineer proteins to bind DNA. The products that they can deliver will be measured in small positive integer numbers, a few diseases.

But the real value associated with being able to engineer proteins that bind DNA are in the uncountable applications people could use the proteins for. It's like a programming language where it would be a big downstream economic cost if you owned "if-then" and you were the only person who could use it. We need to be able to reuse this stuff in combination. Note that the ownership of biotechnology will play out in a landscape that's surfing along a technology transition where, as automatic construction of DNA gets better and better and better, you'll care less about the specific material you have, you'll care more about the information on a computer database and the computer-design tool that lets you organize that information, compile it down to a DNA sequence, and print it. As soon as you start to manage information, all sorts of new ownership, sharing, and innovation schemes become allowable.

Where will we be thirty years from now? 1995: *Haemophilus influenzae* sequence. 2001: draft of human genome available. 2007: multiple chromosomes assembled from scratch, bacterial virus, or organelle. By 2012, the design of eukaryotic chromo-

somes should be routine. Also, five years from now we may have just begun to make some good progress on reliable functional composition of standard biological parts. Nobody knows how expensive solving that problem will be, but because biology works, there are plenty of existence proofs. If I had to guess, I'd say we'll have a collection of tens of thousands of genetic objects that support reliable functional composition between ten and fifteen years from now.

OK, let's cut it a different way: I'd estimate the cost of synthesizing the DNA of every human being on the planet born in the next year at $10 trillion. That's 20 percent of the world's economy. That number is dropping by about a factor of 2 every twelve to eighteen months. On what time scale does it become worth considering whether or not we can afford to construct every new human genome that will come into existence, and we can decouple the designs of human beings from the natural constraints of direct descent and replication with error? My sense is that technology will support this well in advance of our ability to have any conversation on the consequences of using the technology. It's not a fifty-years-off thing, it's not a thirty-years-off thing, it's probably not even twenty years off, in terms of where the technology needs to get.

I like to build stuff, and biology is the best technology we have for making stuff—trees, people, computing devices, food, chemicals, you name it. I somehow found my way to biology and, along with the ambition of getting better at engineering biology, there's this wonderful complementary puzzle of how the hell does this stuff work? All these living systems we inherit from evolution.

I was fortunate in the early '90s to find an engineer, John Yin, now at Wisconsin, who knew something about DNA. He had

just come back from working with Manfred Eigen in Germany, and was studying virus evolution. He was at Dartmouth College, so I did my PhD up there and had an interesting experience as an engineer trying to develop computer models to help biologists understand the architecture of the natural genetic systems they were changing. I had some hypotheses coming from my work, and I tried to get some biologists to do experiments for me; I was not successful in doing that. In hindsight, I recognize that that's because any good biologist who does experiments has multiple lifetimes of work to do. They're never going to do your experiment, so you need to get in the lab and do it yourself.

That took me to Austin, Texas. At the University of Texas, I worked with Ian Molineux, who had done the early PCR work at MIT; he was running one of the last bacterial virus labs in the country, and he taught me how to map and clone DNA and do my experiments. I then spent a summer in Madison, Wisconsin, and then went to Berkeley, where I ended up working with Sydney Brenner and Roger Brent, two good biologists, in an independent not-for-profit; our mission was to do the next generation of biology, whatever made sense. Part of my work there included taking a look at my results from Texas, and I noticed that all of the predictions I had made using my computer models, about how these natural biological systems would behave when we changed them, turned out to be wrong, especially the interesting predictions. I would want one behavior, and when I went to make the change, exactly the opposite would happen. In this situation, engineers do what's called a failure analysis. So I made some predictions: "I'm going to make these changes to the architecture of this virus and as a result, the virus will grow faster." I would go into the lab, make those changes, and it would grow slower. My modeling tools weren't good enough to support

purposeful determinative changes that would result in the behavior I expected.

This was a sufficiently painful process to give me a lot of time to think about why things weren't working out. And the conclusion I came to at Berkeley was that evolution is not selecting for designs of natural biological systems we can understand; the things we inherit from the living world have not been selected for ease of understanding, let alone ease of manipulation. It's not part of evolution's objective function.

If I wanted to be able to model biological systems, to predict their behavior when the environment or I made a change to them, I should be building the biological systems myself. That, for me, was the transition to what's now called synthetic biology. I started broadcasting that idea in the 1990s. The only person who returned a coherent signal intellectually was Tom Knight, in the electrical engineering department at MIT. Tom had self-started in biology five years earlier and is now, in addition to being one of the best engineers I've ever met, one of the best microbiologists I've ever met. Tom was interested in it from his own perspective, having mostly to do with building computers. We need to use biology not to be a computer but rather to build our computers, because we're going to need to put atoms exactly where we want; as semiconductor devices get smaller and smaller and smaller, you can't rely on random distribution of the dopant atoms in the devices. The statistics go to heck, and you have to figure out ways of putting exactly one or two or some small number of dopant atoms in every little gate you've got.

So that brought me to MIT, in January 2002. And along with Tom, there was the opportunity to be responsive to this bigger opportunity: let's engineer biology. We're going to pull off a new department and a new venture in biological engineering. In the

context of MIT, this was not the first time it had been tried. There's a 1939 paper by Karl Compton, then the president of MIT, whose title is, "The Genesis of a Curriculum in Biological Engineering." It describes the ambitious and impressive five-year major, where you get a dual degree in biophysics and biological engineering.

Somehow that earlier effort crashed and burned. I haven't completed my failure analysis yet, and I don't know if it was simply World War II and the redirection of interests or other stuff, but what's interesting to consider is that 1939 is exactly when the Rockefeller Foundation was making investments in the science of biology, arguing, correctly, that the relevant physical level of resolution at which to understand the living world is atoms and molecules. That becomes molecular biology. And so biological engineering could have got started at that same time, but didn't. So that's how I get to where I am.

I've resigned my position at MIT; I'll be moving to Stanford next summer. It's a place that will support the scale of foundational research that needs to happen in biological engineering. There's a tremendous engineering community in the Bay Area, in electronics and software, and these are the folks who have the most relevant skills. If you look forward to what the challenges are in biological engineering, the main challenges are how to manage complexity, meaning how to produce simplicity in a many-component integrated system, and how to develop a theory that supports the programming of evolution.

And there's a way to think about this that maps directly onto communications theory, where you think about a sender and a receiver and a message being transmitted along a channel. In evolution, you have a parent generation, which is the sender, the transmitter, and you have the progeny, the children, which are

the receiver. The message that's being transmitted is the design of the living organism, and the channel the signal's being propagated along is the process of replication of the machine. So, in any case, the work force and knowledge base most relevant to the future of biological engineering is now on the San Francisco Peninsula.

When we organized the first Synthetic Biology Conference at MIT in 2004, we were expecting about 150 people; and 500 people wanted to come, given six weeks of notice. Now it's going on four years later; the fourth meeting will be at the Hong Kong University of Science and Technology, which I think will blow doors off most places in the world twenty years from now. The university is in Clear Water Bay in Kowloon.

It's interesting for me to learn how difficult it is for folks to appreciate what an exponential technology implies. The fact that sequencing went from approximately zero to human genomes in ten years. The same thing is happening with construction of genomes. And with the parts collection—the standard biological parts doubling every year. And the same thing is happening with the number of teenagers who'd like to do genetic engineering; it's doubling every year. How do you live in a world where you're surfing that exponential in a constructive and responsible way? Very few people get that.

15
Eat Me Before I Eat You: A New Foe for Bad Bugs

Kary Mullis

[March 17, 2010]

Kary Mullis received a Nobel Prize in chemistry in 1993 for his invention of the polymerase chain reaction (PCR).

We're working on a way to manipulate the existing immune system so it can attack things it's not already immune to. We've been controlling bacteria for years with antibiotics, but the bacteria are catching on. We've never been good at controlling viruses, unless we prepare for them well in advance by vaccination, but now we can use the same method for them, too—and in both cases the cure is not administered until you're infected and it works right away. It sounds too good to be true. So did antibiotics—they were called "miracle drugs."

In order to understand what we're doing, I should explain how the immune system works. Most people know that you've got this system but not how it functions on the level of molecules and cells. It's a collection of lots of different kinds of cells, each with their own purposes. There are about as many as you have in your brain, distributed mostly in special areas all over your body. The business end of the system is a set of hungry cells that will destroy and ingest things that are designated by the whole system as being "other." The rest of the system is charged with preventing them from eating anything else. New cells are always being born, and they are right away tested for their ability to make