FEATURE COMMENTARY

Synthetic biology: Can we make biology easy to engineer?

Drew Endy

The following is an edited transcription of a plenary talk given by Dr. Drew Endy at the BIO Pacific Rim Summit, Vancouver, British Columbia, Canada, 11 September 2008.

I'll start by relating an anecdote from a friend who's a young venture capitalist. He had sought out an old, successful partner at a firm near San Francisco, a fellow who had made a lot of money by investing in biosynthesis companies. My young friend asked for some advice, inquiring of the colleague how he decided what investments to make in biosynthesis and industrial biotechnology. The old, successful partner said he had three rules of thumb: first, that the company platform have no more than three steps in its process—three enzymatic steps. The second rule of thumb was a potential billion-dollar market. The third rule of thumb was a 90% margin.

This was an interesting story. At the time, I didn't know what to make of it. But as I thought about it, it struck me that the absolute requirement of these criteria would have precluded some of the success stories presented at this *2008 Pacific Rim Summit on Industrial Biotechnology and Bioenergy*. It also suggests that biotechnology, as an industry, is unbelievably immature. If you could imagine the software industry having to meet those constraints (a billion-dollar market required in order to start a company), the world would be a different place.

This is how I want to frame my remarks today: How do we make biology easy to engineer? How do we enable all of biotechnology to "come true", not in some notional point in the future, but over a period of 10, 20, or 30 years? This is my starting point.

My thoughts on this resolve to two obvious things—we need more people working on the problem, and we need better tools.

As far as the people requirement goes: I've been working on this problem for a little while. One fun project I got caught up in was making a comic strip trying to explain to teenagers and others how one might begin to think about programming DNA. In that comic strip, you have a young boy learning from his elder and who gets excited about doing something with a bacteria: finding some genetic material inside, thinking about reprogramming it, trying to brainstorm an idea.

Now, in this case, the young gentleman would like to make a bacterial balloon, some sort of self-assembling biofilm that goes off and does whatever he might want it to do. The boy's teacher is a little bit skeptical, but the boy realizes that all he needs to do is make a little bit of hydrogen gas, then get a biofilm around it, and so on. Of course, he's going to get the components to do this not by going off and doing some primary research (who knows how long that would take?), but by looking up the needed functions in a catalog and getting that piece of the DNA "off the shelf".

The transformation occurs, and over time, something very exciting happens—it looks like he's got himself a nice little bacterial balloon. But the balloon explodes, so in this fictional case, things don't play out necessarily as expected. The boy soon realizes he forgot to tell the program to stop. But the experience was a nice first introduction.

Now, to my surprise, last year, as part of the International Genetically Engineered Machines competition (the iGEM Jamboree), a team of students from Melbourne, Australia, showed up with a new standard biological part, BioBrick Number I-750016 (*Box 1*). Of its own, the number doesn't mean anything – but linked to that particular number is a sequence of DNA, adapted from a natural organism, just over 6,000 bases long. It has about 10 or 12 coding regions that, when expressed, self-assemble into a small 16-nanometer-diameter protein balloon in the cytoplasm of a bacterium. And by adjusting the abundance of these balloons inside a cell, you can make a cell that floats or sinks or is neutral.

Before that competition, I hadn't known anything about this biology; I didn't know anything about this biotechnology function that I might use. Who knows what could be achieved with such a function in, say, a fermentation process?

I was embarrassed that I didn't know this. I was doubly embarrassed because the scientist who identified the genetic elements originally in the soil bacterium worked at University of Massachusetts, Amherst; that is, in my neighborhood. *[Editor's note: Endy was previously on MIT's faculty; see author bio notes at end of article.]* A researcher in my vicinity had figured this bacterial function out, yet it took students' literally coming across the world for this information to come back to me, with a now standard genetic part that I might reuse, off a shelf, in order to learn about this function and get access to it.

These objects and others are now in a collection called the Registry of Standard Biological Parts. This is a pilot project. At the website, http://partsregistry.org, you can find access to several thousand different genetically encoded functions. We have shipped about 200,000 of these objects as pieces of DNA around the world over this past summer to students doing their own genetic engineering projects.

And so, for example, you can find J-45200. This is a genetic object that will catalyze the transition of isoamyl alcohol to isoamyl acetate, producing a banana odor. That part can be put, for example, under the control of regulated growth-phase-dependent production and this characteristic used as an assay readout. (It also allows you to take stationary phase cultures and have them smell like banana milkshakes.)

Through the parts registry, one could also find a visible-light generator, G-10001; you could find a red fluorescent protein generator. Some of these functions are familiar to those in the audience – but what's interesting is that now these are available in a form with a data sheet that lets you get access to them more readily. You don't have to be a PhD student or a post-doc to work with these; you could be a high school student.

Take F-2620, which was recently published in *Nature Biotechnology*. F-2620 is a genetically encoded receiver that allows cells to communicate one from the next. Like the other BioBrick parts, it's ready to go, off the shelf. Its data sheet is adapted from electrical engineering, where you might see the input/output relationship, the latency; but unlike in electrical engineering, we've had to characterize new things, such as the evolutionary stability of the component: How many generations will persist in culture in the absence of selection until it fails? What's the license on the component? Is it patented? Is it public domain? And so on.

Box 1. from the BioBricks Foundation website...

The BioBricks Foundation (BBF) is a not-for-profit organization founded by engineers and scientists from MIT, Harvard, and UCSF with significant experience in both non-profit and commercial biotechnology research. BBF encourages the development and responsible use of technologies based on BioBrick™ standard DNA parts that encode basic biological functions.

Using BioBrick standard biological parts, a synthetic biologist or biological engineer can already, to some extent, program living organisms in the same way a computer scientist can program a computer. The DNA sequence information and other characteristics of BioBrick standard biological parts are made available to the public free of charge currently via MIT's Registry of Standard Biological Parts.

Any individual or organization is welcome to design, improve, and contribute BioBrick standard biological parts to the Registry.

The reason I got pushed into working on stuff like this, getting more people and making more tools – better tools to help with the engineering of biology – comes from my experiences at MIT over the last six years. When I went to MIT, there wasn't a Department of Biological Engineering; there was an abandoned program in applied biological sciences that folks might be familiar with, and a lot of work to do, to be quite frank.

Almost 70 years ago, the president at MIT had written a paper published in the *Scientific Monthly*. It was called *The Genesis of a Curriculum in Biological Engineering*. Being at MIT recently and not having a biological engineering department, it's a little bit puzzling to find this paper from World War II basically outlining a very rigorous and exciting program in biological engineering, where, over a period of five years, one could get a dual degree in biophysics and biotechnology at that time.

So it was interesting to me as an engineer to try to figure out why that program never worked out and how we might do better now. What this led to, was my imagining someone being 17 or 18 years old, matriculating at MIT as a first-year undergraduate, deciding what to major in, and thinking about what one might expect to have learned after coming away from MIT newly minted with a degree in biological engineering.

The exercise to run through quickly is to imagine that you're that teenager; you've looked at the world, you or others you've known have maybe attended conferences, and maybe you're just reading some papers—perhaps the paper of Mitsubishi, where, over a period of seven years, folks took two genomes and combined them, making an eight-million base pair construct that produces a viable organism. Or maybe you saw a press release about a group constructing an almost viable, just-under-600-kilobase fragment of DNA. Or maybe you saw the reports by the group in France that looked at the human genome sequence, found 20 copies of a defunct endogenous retrovirus, inferred what the ancestral retrovirus was that existed about five million years ago, and reconstructed that. If you're really smart, you'll recognize that that achievement was like *Jurassic Park*, but without having the physical copy of the DNA in amber.

Or maybe you saw the movie *Fantastic Voyage* growing up. [To remind you of the plot: A crew and submarine were shrunken down to microscopic size and injected into a diplomat's body to rescue him after he had been shot and developed a potentially fatal blood clot.] You might think that that strategy was a pretty good idea, but you also know that physicists don't actually have "shrinking" rays; so if we really wanted to program objects to go into our bodies and fix stuff, we need small things that we can program. Well, that's probably called "biology", and a paper was published out of University of California, San Francisco, describing how researchers took *E. coli*, implemented genetically encoded sensors, logic, and activators to program cells to identify tumor microenvironments and invade and destroy cancerous cells at those locations.

Maybe some of you are familiar with the work coming out of Cal Tech (California Institute of Technology) where scientists have implemented all of Boolean logic using an engineered ribosome, getting that logic system to work inside cells for the first time. Maybe you've heard of the work coming out of Jay Keasling's labs, where they're trying to adapt biotechnology to help solve problems related to malaria. Or maybe you've seen the structures on the roofs at MIT where folks are trying to recycle carbon coming out of power plants, pulling down nitrogen compounds in the meantime.

Maybe you came across the paper describing the introduction of a new photoreceptor into mice. Mice apparently see in two colors; now they can see in three (whatever that means to a mouse). Maybe you saw the paper where researchers added a single photoreceptor to *E. coli;* as a result, when the *E. coli* were spread out on a surface and exposed to a light image, the bacteria effectively "took a picture" of that image. (That sounds like a little gimmick—"coliroids", or something like that.) But if you think about it, you've got a micron-scale package; that's just under a gigapixel per square inch. This could be controlled with a mirror array—and so you have, essentially, across a surface, a manufacturing platform that's programmable. What might you achieve with such a platform?

You might be aware of the work of scientists now able to implement cell-cell signaling in band detectors to create spontaneous bull's eye patterns and other shapes in plates; this now works in bacterial systems, yeast, and mice stem cells.

You might have seen the self-assembling DNA patterns coming out of Cal Tech, where you can make happy faces or the world map or spell out your name.

And maybe you've seen things in nature itself; most certainly you have. For instance, there are self-assembling sponges that produce very interesting skeletons that the group of Joanna Aizenberg (then at Bell Labs, now at Harvard) was interested in, due to the optical properties of the materials. Essentially, this thing "makes itself". How would we make such a system?

If you were a student in the architecture department at MIT, you would know about the Fab Tree Hab—a design for homes built from living trees. It's the kind of advanced technology you would expect to see, as a new architecture or engineering student joining that school. Now, what if you were 17 or 18 years old and you wanted to become a biological engineer and thought that MIT should have a department of biological engineering? It turns out that the students do expect to be able to achieve similar kinds of innovations—just as if they were in electrical engineering, except the substrate's different: They want to learn how to design and build living organisms that behave as expected, and how to write DNA programs that do their bidding:

"I'm not interested in doing a bunch of research. I want to be able to do the equivalent of going to my computer, typing some text, hitting 'Return', then have my input command result in something being printed to the screen; so it should be when I try to express a gene. I don't want to have to get caught up in a whole bunch of crazy basic research."

Now, it's not that the basic research is bad; certainly, it's where we are oftentimes with the technology. But here, in this discussion, the initial expectations are those of matriculating undergraduates. So for me at MIT, and now at Stanford (which is probably four years behind in curriculum development), we've got to figure out how to be responsive to this student demand.

What students now want to be able to do is consistent with the early promises of biotechnology—recombinant drugs and other compounds, gene therapy, nitrogen-fixing crops. I'll note that not all of these have been delivered over the last 35 years. Global cycles of elements and energy, distributed manufacturing, water treatment, food, shelter, medicines: we haven't scratched the surface yet of the full potential of biotechnology applications. And so this idea of getting more people and getting better tools isn't an idle concern; it's a pressing concern if we'd like to see some of these things happen.

How to take this substrate (nucleic acid), one that's beautiful, that we're responsible for, that's responsible for our existence, on which our civilization utterly depends, and make it more "engineerable", and lead its deployment in a way that's overwhelmingly constructive – this is our challenge.

A long time ago humans thought that rocks were useful, and we started making stuff with rocks; and that was hard, because the rocks we inherit from nature aren't designed to be easy to build stuff with. They're all different. And at the time you could imagine some folks wanted to go study all the rocks and so became geologists.

But a different set of people decided to invest their life energy into making rocks more regular, producing standard rocks with better interfaces, making construction better. Later on, still in the Stone Age, but a more advanced Stone Age, people grew fed up with rocks and started to grind up rocks and make new "synthetic" rocks; reinforced concrete comes into existence. And all this has culminated in such structures as the Millau Viaduct in France – representing the culmination of Stone Age technology to levels it has reached today.

Similarly, if we were to walk back 50 years, computer technology would appear vastly different. People then were deploying silicon-based technology and related things to help design hydrogen bombs, basically; that is, it was exclusive technology and large-scale. Only 25 years later, or thereabouts, folks become so fed up with limited access to computing technology that they found ways to make the personal computer. And now, a generation later, we're all familiar with Web-based computing and handheld computing devices and iPods.

So how – and it's the transition that's interesting for me – how do we go, in one generation, from huge mainframes to the kind of computing technology we have now? I think a lot of that is driven by the fact that silicon-based technology is, essentially, really cool. Well, biotechnology is even cooler. You've got reproducing machines; it's the stuff of life. You can program it with DNA. It's a nanotechnology that actually works. And if you get into the front lines of education, you'll see that the excitement and interest that students have for learning how to become engineers of biology is overwhelming.

If you look at the transitions in electronics, a lot has been driven by great ideas and great opportunities in the markets – but it's also driven by tools: theory related to information, signal processing, device design, computer languages and grammars, standardization, abstraction, electronic design, automation, control of dynamic systems.

Interestingly, electronics, which has kept developing as an industry, is now starting to produce tools that are becoming still more interesting to biotech. There is an esoteric subject area in computer science called amorphous computing. Amorphous computer scientists, as it were, try to figure out how to program independent objects that aren't coordinated with each other in any global way, such that they self-assemble into predetermined patterns. For example, Radhika Nagpal of Harvard has written a language to create a simulation of an individual cell reproducing – making more cells – using all local information; that is, there's no clock or synchronization. Eventually this simulation results in a four-legged object. With this language, you basically have circles grow to a particular diameter, you identify a point on the circle, you start growing another circle, and so on.

In other words, computer scientists have a lot of experience developing languages that operate systems in time; that is, they can make computers operate over time. They're now starting to develop programming languages for making shapes in space, in three dimensions. So if you wanted to grow your own house or furniture or a replacement organ, you could imagine writing a single genetic program in a cell that reproduces and divides and assumes the prescribed shape.

So back to biotech: What are our basic tools?

- Recombinant DNA for cutting and pasting preexisting fragments
- PCR
- Automated sequencing
- Automatic construction (synthesis)
- Abstraction
- Standardization
- Reliable behavior

Actually, these last four tools aren't so mature, and a lot of the success over the last 35 years depends on the first three (recombinant DNA, PCR, automated sequencing), which allow us to write and read genetic material – very primitive tools. What if we could bring online more and more of these tools? And I'm not going to say that the list above is a comprehensive or even a best list, but it's a start.

If I talk about automatic construction for a second: folks will be familiar with A, T, C and G. These are phosphoramidites derived from sugarcane. Plug these into a DNA synthesizer, and it will make DNA for you. Generally, lab-size bottles contain 5 or 10 grams of each of the reagents; that's about a hundredth of a mole. A hundredth of a mole is 6 x 10^{21} molecules. So just park that number by the side.

Now let's take the world population and round up to 10 billion people on the planet (we're heading in that direction). Now take the human genome; also rounding up, it has about 10 billion bases. So 10^{10}x10^{10} is 10^{20}. In other words, if you take that number as a representation of the total amount of human genetic information available, it's 60 times less than the amount of reagent one might have in four standard bottles of nucleoside phosphoramidites.

In other words, for under $1,000, you can right now buy enough material to make 60 copies of every human genome on the planet. That would be like having all the silicon you needed to make 60 times the number of all the microprocessors currently available.

Now, our synthesizers and our construction of DNA technology stink, to be frank. There's still a lot to be desired, with all respect to Marvin Carruthers and everybody who's come beyond, and the teams I've been involved with; we've got a long way to go.

So if I look across scales of length and begin to evaluate opportunities, where is our roadmap? If I were in semiconductor manufacturing—say, if I were Intel, and AMD and others were competing viciously—I would still have a public roadmap that outlined how we would get better at this basic technology of manufacturing wafers of silicon.

I submit for our consideration here that we should, as a biology industry, figure out how to have some coordination of development of our tools. Getting better at building DNA would help everybody. We've wanted to deploy biotechnology for constructive purposes. It will help to bring in many more people to the field who would otherwise never put up with having to clone and do PCR and all other kinds of tedious routine tasks.

As we get better at building DNA, the next problem that comes up is, What do you make? How do you figure out what to say? And so most of our genetic engineering, as we teach it to the students, is at the level of DNA sequences: "TAATACGACTCACTATAGGGAGA", for example (this being a consensus promoter for the T7 RNA polymerase; *Figure 1A*). But there are a lot of sequences of DNA one would have to memorize in order to be a successful genetic engineer, and to require students to memorize libraries of sequences would be immoral, in many respects. It would be like teaching students to write computer programs in machine language, with zeroes and ones. And at one point in time, that teaching method was dominant; and yes, knowing sequences is still a valuable skill—but it's not the only thing to rely on. And so the idea would thus be to begin to "hide" information at this level, yet still preserve for reliability of function.

And so, in trying to build up the biological parts registry, we've started declaring things. We're going to say simply that things exist called parts. Parts are basic biological functions encoded as genetic material. We'll ignore the sequence, "hide" that behind a number-label, and link functional information that we care about to that label (*Figure 1B*).

Devices are combinations of parts (*Figure 1C*). For example, one could wish to make an inverter—something that takes in one signal and produces the opposite output (either high-to-low, or low-to-high). I could think of starting to put devices together to make systems. If you were to put three inverters in a ring, you could create a ring oscillator (*Figure 1D*). And so, in making such a device work, what you end up with is what a computer scientist would recognize as an abstraction hierarchy. Some people become systems experts and engineers; other people understand how to build DNA. They don't need to know the same things. You could perhaps have biological engineers who don't know (and don't necessarily need to know) that DNA is made up of four bases, if we could figure out how to organize the work and separate it into different layers.

This is what we've started to try out, with the Registry of Standard Biological Parts. So one could feasibly adapt some biology discovered in the light organ of a squid from the Pacific Ocean, build on the basic biological research, convert that biological function into a genetic object that recapitulates cell-cell communications, standardize that object, write up a data sheet by doing characterization around that function, and end up with an abstract object—a black box, literally—that can be redeployed in combination with thousands of other so designed objects.

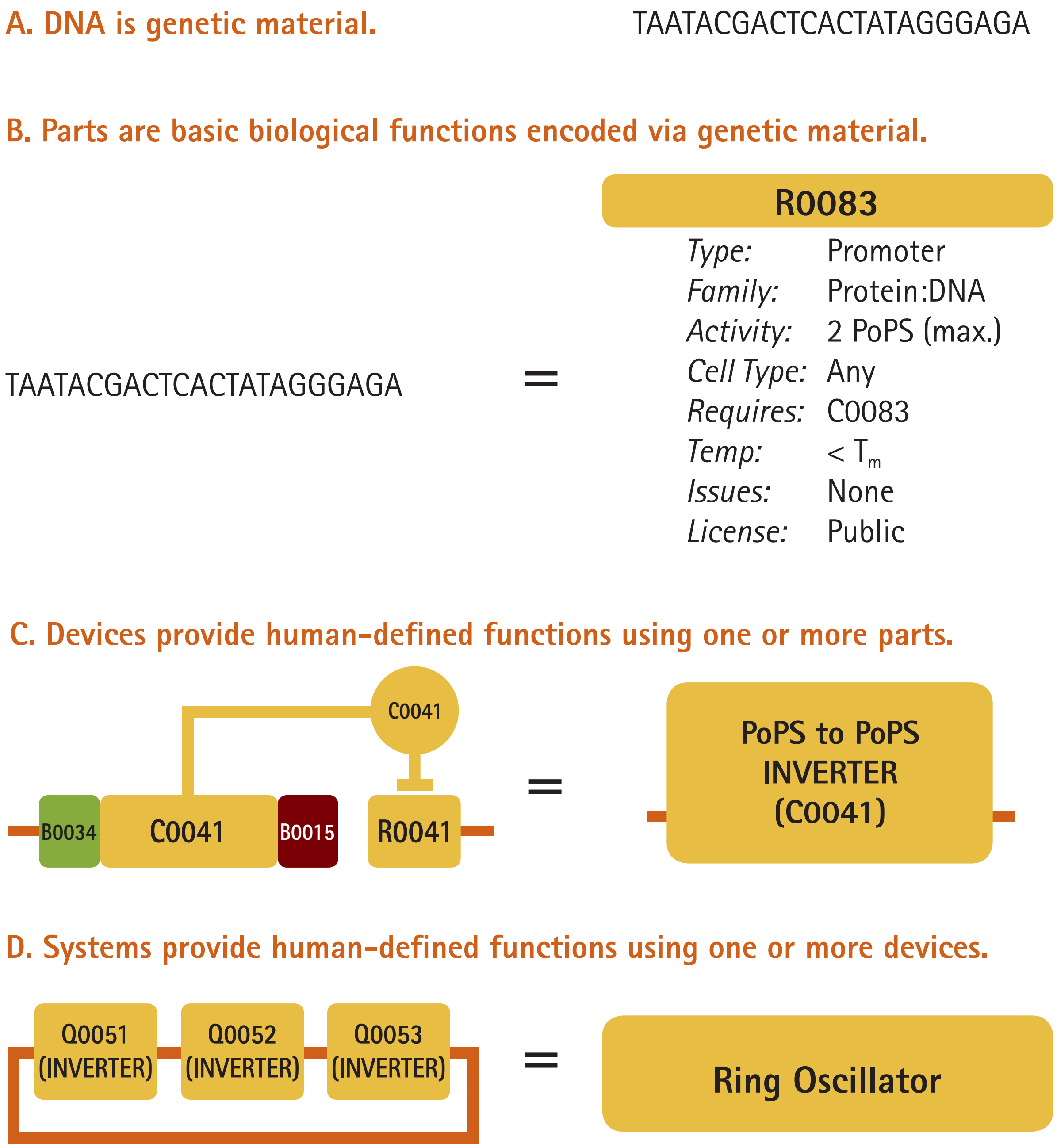
One of the projects I'm starting up in San Francisco is to make a public-benefit factory that will focus on production of tens of thousands of data sheets, each an open-source genetic object. Do we really know how to enable reliable functional composition of genetic objects? Not at all; and we won't know how in any case, until we collect good data on failures and debugging and so on, and work together on that.

As we've started to make very modest investments in some of these tools—like getting better at constructing DNA, implementing a registry of standard biological parts, figuring out the beginnings of an abstraction hierarchy for managing biological complexity—the students have been very responsive. In a first course taught in synthetic biology at MIT back in 2003, we had 16 students; they tried to recapitulate Mike Elowitz's work, where he'd made a bacterial oscillator. We failed. We placed orders for gene synthesis, and we ordered 60 parts; 30 of them came back, and the other 30 were toxic and couldn't be cloned. It was a nightmare.

But in going through that cycle and being unapologetic about it, we learned; and we eventually got 58 of the 60 things we ordered, and it was successful enough that we got to teach the course again next year; then we had 20 students. None of their projects worked.

That summer, though, other people thought the idea of what we were doing was fun, and so we had a five-school competition. You remember the "coliroid" bacterial photography system, which was produced by the team from Texas and UCSF, published in *Nature.* Not bad.

The next year we had 13 schools in the competition, three outside of the US. And the year after that we had 37 schools, 20 outside the US. What was interesting this year was that the students made a



A. DNA is genetic material.

B. Parts are basic biological functions encoded via genetic material.

C. Devices provide human-defined functions using one or more parts.

D. Systems provide human-defined functions using one or more devices.

Figure 1. Schematic of equivalence of biological material and activity with functions, devices, and systems.

world map of where all the teams were coming from; at that point we had the International Genetically Engineered Machines competition or iGEM. The students noticed that there were no teams from, for example, China, Australia, and many other countries. And so during their summers off, without our prodding them, when the students were traveling, they figured out who was traveling where – and so, for example, students traveling in China visited six of the best universities in that country. Next thing you know, Christina Smolke and myself, from Cal Tech and MIT, respectively, find ourselves in Tianjin having a two-day workshop with students from those universities. Five of those universities fielded teams for iGEM, and the team from Peking University won the competition last year, competing against 600 or 700 students from about 57 schools.

This competition's happening again. It's doubling to a first approximation every year. We're expecting about 900 students on the first weekend in November at MIT, presenting their work. *[Editor's update: The event was held, with the undergraduate winning team, from Slovenia, producing a synthetic vaccine for* Helicobacter pylori.*]*

As for the sorts of projects one might see, here's one example:

```
if {growing}
 call wintergreen()
else
  call bananas()
```

Second-year undergraduate students at MIT decided they didn't like the odor of *E. coli.* So they figured out that they could knock out production of indole, removing the fecal odor, and add in production of methyl salicylate (wintergreen) odor and isoamyl acetate (again, banana): their "Eau d'*coli*" project.

They specced out a full biosynthesis from chorismate to methyl salicylate; they didn't get that working. They did get salicylic acid to methyl salicylate working in a period of ten weeks, and with growth phase-dependent production of the odor. Conversion of isoamyl alcohol to isoamyl acetate did work. The students basically were able successfully to implement a rapid prototyping of a system, before encountering a classic metabolic engineering problem of how to get the yields up and what have you.

I had an interesting conversation with a perfumer from the manufacturing company International Flavors and Fragrances; he was incredibly excited about making various things as chemicals, leading into perfumes. But we ended up having a much more interesting conversation about distributed manufacturing – in situ manufacturing of scents and fragrances wherever you might like them, using biotechnology.

Another example is work done at UCSF, which (since it doesn't have undergrads) borrowed high school students from the biotechnology program at nearby Lincoln High School in San Francisco; as they described it, their summer job was, "I was a teenage genetic engineer". These students had done some computer programming, and what they had recognized in their computer programming lessons is the power of languages like Java. Java's a very interesting language; you can write a computer program on Java, and it'll run on many different types of computers – a Mac, a Dell PC, a Linux computer, and whatnot.

This is because the same Java program doesn't run directly on the computer hardware. It interfaces with something called a Java Virtual Machine. The Java Virtual Machine is built out on each hardware and operating system platform, and then translates the Java program to the common environment.

The students were fed up with the experiences of other genetic engineering teams they had interacted with or heard about, and thought it would be better to provide a common operating environment inside cells, such that, if you wanted to do some genetic engineering, you wouldn't just toss your program into whatever chassis you happened to want to work with. You instead could have a virtual machine that provided a common environment across all organisms.

They started to make progress on this. What they decided they could do in yeast, in this case, was redirect receptor endocytosis, so that vesicles that come off, instead of going to the lysosome, begin to accumulate – basically beginning to make a synthetic organelle. The students called this a synthon, and the idea would be that, by providing an additional address (which they did by bringing an orthogonal phospholipid system and showing that you could bind GFP to that), you could target your genetic programs to operate inside this organelle.

Over the period of a summer, the students were beginning to show they could get vesicles forming and could target those vesicles, with a tagged receptor being properly localized to the unique phospholipid species the students designed. This are high school students; they blew me away, because they were teaching me biochemistry, and bringing that to bear on a problem of advancing engineering theory to first approximation.

Let me end with some constructive questions (*Box 2, p.344*). I have strong opinions on these, but I don't have answers, and I'll end with a comment about that.

First, should teenagers practice genetic engineering? This question is really about safety, one that many of us are familiar with.

The second question is about security. This is a national-security question. Should there be secret Biosafety Level 4 facilities? If you go back and read *Scientific American* from 1970, Matt Meselson and his colleagues argued, “No way.” And that was much of the response during the Nixon administration. We have a different world now.

Third, is garage biotechnology going to happen, and is that going to be good or bad?

Fourth, should these parts (the BioBrick parts, for example) be patented or freely shared? The Registry has shipped 200,000 standard BioBrick parts, that are really easy to reuse, to teams of scientists in 40 countries. It’s an amazing thing, what you can do when you have 2,000 pieces of DNA ready to go. You don’t have to go look them up. They’re there for you, well organized, and designed to work together. What would happen if we could transition biotechnology from an economy which today might be described as balkanized monopolies, to a network economy, where there are common platform technologies upon which people compete to deliver products and services? Should genetic engineers sign their work? Should they be licensed? I’m trained as a structural engineer (undergrad). Structural engineers work in reinforced concrete and steel; and the idea of being a professional engineer and signing your bridge designs is normal. What about things that we engineer and release into the environment?

Again, these are questions I’m raising. I have opinions, and lots to discuss. These are tricky questions. And it’s interesting: If you were to ask of the IGEM students, “How many of you know that, in 1975, a generation before you, there was a conversation involving recombinant DNA technology and biosafety?” about 1% or half a percent would know just that there was a conversation (not that they would know what the outcome of that conversation was). So even if some of these questions are old questions, we’re at an interesting point where, for example, because of a new generation coming up, many of whom come from electrical engineering, computer science, or physics (that is, not being trained in microbiological safety to begin with), it’s time to figure out how to scale the conversations and bring these students into the knowledge and discussions we’re familiar with.

Box 2. Discussion/solution frameworks should be adaptive:

- Should teenagers practice genetic engineering?
- Should there be secret Biosafety Level 4 labs?
- Is garage biotechnology inevitable?
- Should BioBrick parts be patented or freely shared?
- Should genetic engineers sign their work?
- Should genetic engineers be licensed?

Post-presentation Q&A

QUESTION: What do you expect to be the biggest challenges for the synthetic biology industry?

DREW ENDY: One of the biggest challenges is to have a practical impact in a way that is measured in economic return. I think that you’ve seen (for example, in the gene synthesis industry) a very quick collapse to a commodity marketplace that makes it challenging to sustain capital investments that really drive forward improvements in the technology.

If you look at the other foundational technologies, we’ve just now seen the first start-up beginning to provide standardization of components as a service, and we’ll see how that venture goes. Actually taking tools businesses, figuring out which ones are going to be real businesses, and seeing them get going so they can have an impact is a first challenge.

I think many of the other challenges are summarized in the questions I posed at the end of my talk. In my own life, I probably now spend 60 to 75% of my energy dealing with those questions, not with the research itself. And I freely admit that I feel as though I live in a country that lacks a strategy for biosecurity, that doesn’t have an ownership, sharing, and innovation framework well matched to where biotechnology is going, and so on. These become, very quickly, limiting factors in progress. Working through the social issues of human practice is easily as challenging as anything else.

QUESTION: Given your obviously unbridled enthusiasm for research, and as an academic, are you comfortable with, or do you see problems with the large amount of commercial dollars funding research in relation to synthetic biology? I’m referring here to the British Petroleum/UC Berkeley deal. I’d like to hear your thoughts on that, given the very strongly commercial nature of research around industrial biofuels in academia.

DREW ENDY: That’s a great question. I don’t think there’s enough investment. I’ll tell you about a conversation I had with an executive at BP who was involved with the establishment of the Energy Biosciences Institute. I gave him a hard time. I said that if the energy companies were really serious about solving the challenge of renewable energy, they wouldn’t take a couple weeks’ profit; they would take, say, one quarter of the profit from Exxon Mobil or other company and would endow six Cal Tech equivalents: one Cal Tech equivalent working on bioenergy, one working on nuclear, one working on solar, one working on conservation, one working on other aspects.

If you go to the website Wattzon (www.wattzon.org) you’ll find a pretty accessible analysis of human civilization, global energy cycles, and global carbon cycles, and you’ll quickly come to appreciate the challenge of how we are going to get to 15 to 18 terawatts of renewable energy, and do that within our lifetimes.

Take biofuels. Say you want half a terawatt of renewable biofuels. It you work the calculations through, that would be equivalent, as I understand it, with taking the best algae technology available today and bringing online one Olympic-size swimming pool every second for the next 25 years. That's a big challenge. That feels to me like it's at risk of triggering a kind of reorganization of manufacturing that I'm not familiar with except from the history books (for example, World War II). You'll see the same thing with bringing on nuclear energy and other energy approaches.

I had been trying to give this gentleman from BP a hard time, but what was interesting is that he agreed. He said that they didn't know how to make such investments, and that the experience at Berkeley as it plays out is going to be an opportunity for them to learn.

Now, I'm ignoring many different facets of that relationship which could be discussed that I'm not qualified to comment on. But I think there is an urgency in particular with respect to global energy and carbon cycles, to get off our butts and get it done.

Susan Hockfield, the president of MIT, published an editorial in the *Washington Post* [*Editor's note:* Reimagining Energy*, Thursday, September 11, 2008, p. A17*], talking about the energy challenge that we're facing and what we might start to do to step it up. (Substantial government investment into basic research will be key.)

Vis-à-vis other investments in synthetic biology, I think that the challenge I've experienced is figuring out how to get enough capital to make investments while also being in a position to give away many of the tools. The reason that the BioBricks framework is succeeding is because we're not trying to capture. We're instead a worldwide network of people who feel comfortable freely contributing and reusing these biological objects. Solving the legal challenges around that, getting the quality control in place, and whatnot—all of that is going to require a lot of help, not just in academia, but throughout all of history. I'd really welcome an opportunity to work with folks and talk through things with people who have a lot more experience around this from a commercial side.

So I don't think there's enough investment, is the short answer, and I do think there are a lot of issues that could be raised around this. But, to be fair, many people are learning how to work together. I'd love to live in a world where there's a lot more cooperation in biotechnology, one in which we recognize that we haven't scratched the surface in terms of what can be done with the technology platform.

QUESTION: My question pertains to the comparison you've drawn between synthetic biology and computing. The big difference, I think, between computing and synthetic biology is the framework in which computer software is run. We design and build computer hardware. Then computer software, although it's self-replicating and such, is confined to computer hardware. It's not going to go outside of that physical system; and in the case of catastrophic failure, we can just unplug computers, and that software will remain completely confined to that environment. Of course, I'm referring to computer viruses.

The difference with synthetic biology is that there's no such framework in which synthetic biology is confined. We cannot just unplug life in which this biology is existing. What kind of systems do you, as somebody working and doing research in this field, see being put into place to ensure that there is no chance of catastrophic failure of synthetic biology?

DREW ENDY: There is going to be failure. Go read about the Tay Bridge disaster in Scotland in the 1800s, when the gale knocked the bridge over as a train was going through. All examples of technology developed by humans that I know of have catastrophic failure as a part of it.

Now, the question becomes, How do we minimize that to a level that's acceptable, and how do we make the technologies more reliable such that failures are distributed more broadly? I found it interesting, the stories about the GloFish (zebrafish with an engineered, jellyfish fluorescence), and the fact that company couldn't go to market with the fish in California because the state couldn't afford the environmental impact assessments (at least according to how the details are presented on the producing company's website).

Is this going to be like electronics? Sure. Of course, we're not going to be able to control the humidity and temperature like we do with our microprocessors. An Apple laptop, for instance, is only rated for a certain temperature range. But could we in biotechnology and bioengineering do better? The answer is, Absolutely, yes. Exactly how to do this will probably require a lot of coordinated effort.

Could you, for instance, develop a genetic code such that every point mutation were selected against? The answer is probably yes. What you would do is expand from a triplet code to a quadruplet code. You wouldn't add in any more amino acids or tRNA. You'd still only have 20 coding codons. But by going from a triplet to a quadruplet code, that takes you from a 64-base to a 256-base code. If you only have 20 coding codons and a 256-base code, there'd be enough empty space to bracket every coding codon with a null.

Pete Schultz has demonstrated that you can get a four-base tRNA. Now, you can imagine this would be a crazy project, in terms of feasibility and how long it would take—but what if you implemented a four-base genome, such that it were a "fail-fast genome", as an electrical engineer would call it. If there's a mistake, it either dies, or it stops growing.

So do we have room for improvement? You bet. I think it's interesting to imagine how tools like synthesis of DNA could be deployed to do things heretofore completely impractical.

Another example would be what happened around poliovirus synthesis. Researchers had demonstrated that you could construct poliovirus de novo from mail-ordered oligos. A couple years later, it was shown that by making use of dicodon bases, you could recreate a derivative of polio that was attenuated, in such a way that it has 1,000 different mutations relative to the more virulent strain. All of a sudden, instead of biological synthesis being a risk generator (say, for reconstructing pathogens), it can be deployed as a technology platform for rapid, if not automated, production of vaccine platforms for the very same disease in question.

Your question is a great one. There are two parts to it, in a way. Synthetic biology isn't going to be like electrical engineering, and I did make reference to that. But I also made reference to structural engineering. I could also talk about mechanical engineering and standardization of screw threads and so on and so forth. Biotechnology's both different from and the same as these engineering disciplines. Maybe we can learn some things, and maybe other things will be dangerously irrelevant, or just dangerous. We'll have to figure it out. I don't know how to get to the right answers besides trying.

QUESTION: I think that my question was mainly in reference to the self-replicating nature of biology.

DREW ENDY: Let me take that one straight on. We don't have a theory that supports the design of reproducing machines, because no engineers have ever encountered reproducing machines. If you want to read a book on this, the only book I could recommend would be *Theory of Self-Reproducing Automata* by von Neumann, and ironically he ran out of time at the end of his life, so he didn't finish it. (The book was later compiled posthumously by a colleague of his.) And so it's not a satisfying treatise on how to design understandable reproducing machines. Right?

QUESTION: Should we have biological engineering students study material beyond engineering proper, for example, make them take a philosophy course as a requirement?

DREW ENDY: It's a great question. I'm not going to say that we've nailed this, but I'll give you a module from a sophomore-level course. We talk about what's changed in the last 35 years. We've now got databases on the Internet populated with sequence information (*Box 3*). So we've got the Internet, we've got DNA synthesis and construction, we've got overnight shipping, and all of a sudden, since the anthrax attacks in the US in the fall of '01, we've had a huge concern around synthetic biology applications in the US. After all, we can find information on anthrax in public databases. But what's changed since the 1970s? NCBI (the US National Center for Biotechnology Information) came into existence. It contains genome sequences for hemorrhagic fevers, and the sequence of *Ebola*.

But then you could ask the students to work through the following scenario: "If you were in charge of federal policy in the US, would you be supportive of a world in which the sequence information for pathogens were available? Yes or no?" Ask the students to debate that, reflect on it, write it up, and come up with recommendations. In this case, we tend to a slight majority in favor of open distribution of the sequence information, the argument being that there isn't a history of solving challenges in therapeutics and vaccination in secret, and the way to go is to keep things open and have folks work together. And so the students talk about that.

We also have them play a game where they try to assemble a fully integrated genetic system, and where they need to license six different genetic functions. Their colleagues in the class each own a different function, and they try to figure out what the term sheets are going to be. They have to do this quickly; there's a time limit to close their deals.

Box 3. What's changed since the 1970s?

- Databases populated with sequence information
- The Internet
- Early returns on investments in DNA construction technology
- Overnight shipping
- Expanded concern about active misapplication of biotechnology

The first time we did this, nobody could get licenses. One very bright, obnoxious student found a bottleneck in the system and tracked down who held the licenses to all of those and hoarded them until the very last minute, right before the deadline; and everybody realized they had to do a deal with him.

So for all the different areas around those questions: yes, we're trying to bring these right into the classroom. I can't say absolutely that it's sufficient, but we're working on it. And if you have suggestions, I'd really welcome your bringing those forward and trying to develop modules around it.

QUESTION: You've been talking about explosive innovation—things happening, people getting great ideas. In the real world, you run up against several paradigms of thought that really are more influential on innovation than anything else. Business education, the bankers—that's where the money comes from in the end.

You also mentioned legal framework and regulatory frameworks and safety. These areas don't seem to like explosive innovation; they seem like big brakes.

We have these different cultures and different ways of seeing the world. I'd like some more enlightenment on those questions you asked. What are some of the main points in discussions among the thought leaders in this area? Can you elaborate a bit on how this discussion is going in what you'd call the public domain? How is synthetic biology going to unfold in the future?

DREW ENDY: I'm happy to address this, but it's a dangerous question because it's a big question. Is there a particular issue-question I mentioned that you'd like to use as a case example? Do you want to take security? Do you want to take ownership, sharing, and innovation frameworks? Do you want to take safety?

QUESTION: Maybe intellectual property and how it gets handled. How do you license? What should be licensed?

DREW ENDY: I'm incredibly biased on this topic, as one of the founders of the BioBricks Foundation. You know, running my lab, I experience firsthand (though, I know, to a lesser degree than do folks in industry, given the sort of working research exemption in the US) the challenges and latency associated with getting access to tools. Many projects in my lab have not gone forward because I can't get a material transfer agreement in place, for example, and it's to the point now where we simply refuse to deal with those parties. Thanks to synthesis technology, if I can get access to the sequence information, we'll just go build it ourselves and short-circuit those limitations.

But look at the BioBricks parts collection (as I mentioned, we're shipping hundreds of thousands of parts around the world) and the legal framework by which we do this. When a paper came out in *Nature* describing the parts registry, one of the first e-mails we got was from a company's researchers saying, "This is cool. We really want this parts registry. Who owns all the parts?" We didn't know how to respond to that. So we just kept focusing on undergraduates. Switzerland, I now know, has a very nice research exemption, and we thought about moving there. But...

So, how do we get to a world where we actually could have an open source collection of genetic reagents that aren't just in the public domain? Because the public domain is weak in one critical respect: It doesn't protect against encumbrance of reuse and combination. So what legal mechanism would you use? Would you set up a license? Would you do something else? Set up a new type of law specific to the technology? I think that's where we're going in biotech.

But in the meantime, what we've been trying to craft (and this is work done in collaboration with the Samuelson Law, Technology, and Public Policy Clinic at Berkeley Law School, and the law firm, Fish and Richardson, who represent the BioBricks Foundation) is a very permissive opt-in license that will allow people to contribute more readily—either via the public domain or via patented technologies—standard genetic objects that can be reused readily.

What we're informed by is, in part, the past experiences around software. If you wind the clock back 20 years and look at the beginnings of GNU and Richard Stallman's work with the Free Software Foundation, all of that was a reaction to what was happening around the early days of Microsoft and others, and the closing of what had previously been a software technology community in the open.

That experience was a dysfunctional experience, in my opinion. It in a way framed religious poles around property rights, and it took the software industry about 20 years to fill in the richness of the innovation ecology between those two poles; that is, free software and Microsoft, to use some names. And so you didn't see Red Hat, for example, coming on until much, much later.

I think that in biotech, since we're obviously dominated by patents and then using public domain to an extent (in the US, at least, right now), we're better positioned to have a much more constructive population of alternative legal frameworks that don't disallow investment, commercialization, and returns, yet allow us still to begin to accrue common, sharable platform technologies that are free to use.

Companies like Google come into existence, in part, because of the code base that they don't have to pay for, such that they can go raise $100,000, start something in a garage, and then go convince other investors to put in a lot more money, eventually. I think, off the top, based on what I've been involved with, those are the thoughts I'd have on that topic. My sense is we've got to go with existing law for the foreseeable future, and then, on longer time scales (maybe five to ten years) implement something that's more specific and better suited to the domain. It will be very interesting to see what happens as sequencing and synthesis technologies get better, as they will make genetic material and genetic information interconvertible. That will impact the security frameworks, but also who owns things. It becomes a lot more like something copyrightable, and if copyright weren't such a mess, based on what I know, you could think about using that as a model.

Sometime in the future we're going to have something like an iTunes store for genetic reagents; you download stuff, you print it on your DNA synthesizer, and there's your megabase of DNA doing something great. It shouldn't cost the Gates Foundation $42 million to make the replacement for artemisinin, when artemisinin-resistant mosquitoes, and malaria, are all over the place. We need that project to be a $5 million or a $500,000 project.

QUESTION: In spite of Richard Feynman saying there's plenty of room at the bottom, if DNA replaces silicon, how many times more can Moore's law take effect? Five times? Ten times? I think mathematically you run out after about ten.

DREW ENDY: Tom Knight pointed out to me that, via the work of one of his students, DNA-based computers (self-mixing computers) have a performance limit that makes "solid-state computers" (if you will) pretty exciting. I don't think DNA computing will displace silicon-based computing.

But Tom also pointed out that biology is a nanotechnology that works—you know, *really*. You could use biology to put atoms exactly where you want. If you think about what's happening in silicon, as feature sizes get smaller: with dopant atoms, which are currently deposited via plasma process, you get really bad statistics in the dopant concentrations across your devices. What biology could do is provide scaffolding that puts dopant atoms exactly where you want, and that might get us to smaller scales in a "solid-state" computing device. I think we'll evolve from using our computers to build biology, to building our computers using biology; that'll be the more interesting thing.

Drew Endy is assistant professor in the department of bioengineering at Stanford University. He previously was a junior fellow, and later an assistant professor, in the department of biological engineering (founded 1998) at MIT (Cambridge, Massachusetts). Endy is also president of the BioBricks Foundation (website: http://bbf.openwetware.org/Board_Members.html) and a founder of the Registry of Standard Biological Parts (website: http://partsregistry.org/Main_Page).

* The preceding text is an edited version of a plenary talk given by Dr. Endy at the 3rd annual *BIO Pacific Rim Summit on Industrial Biotechnology and Bioenergy,* held 10–12 September 2008, in Vancouver, British Columbia.

For more information, contact Paul Winters, BIO: pwinters@bio.org. Web: www.bio.org/pacrim.