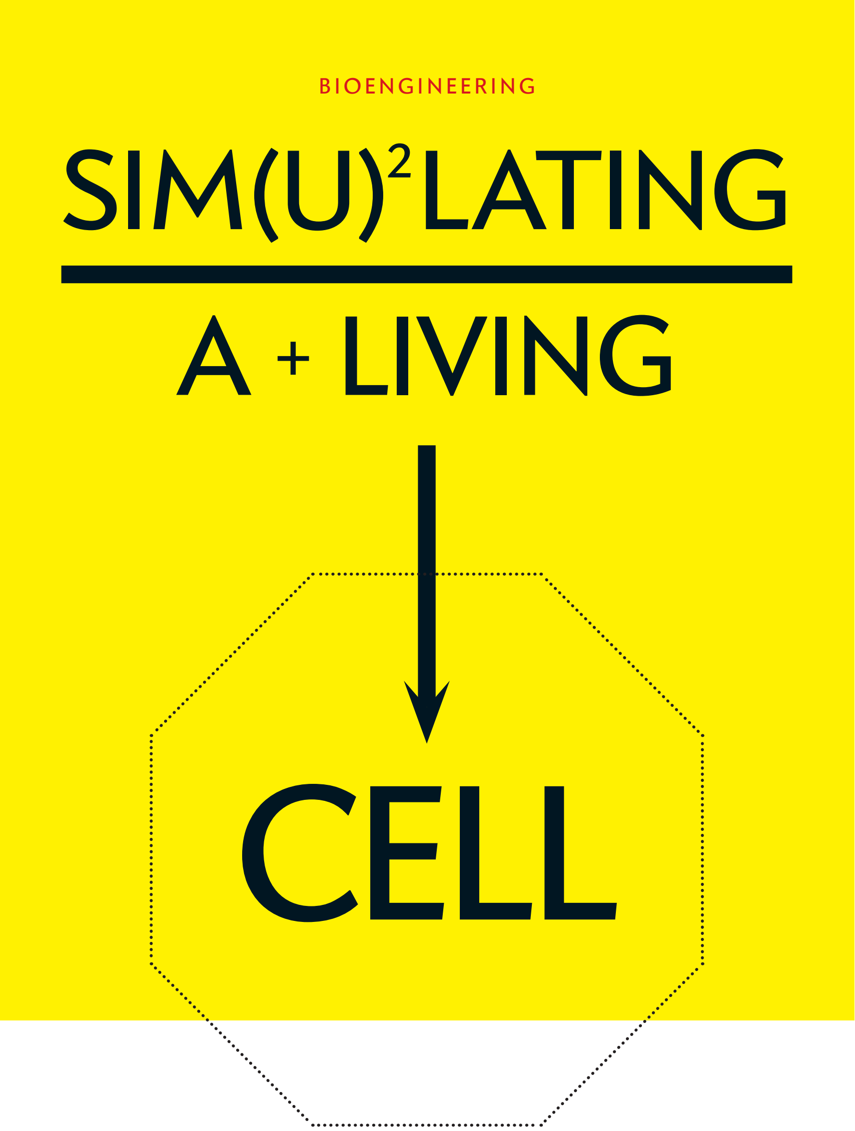


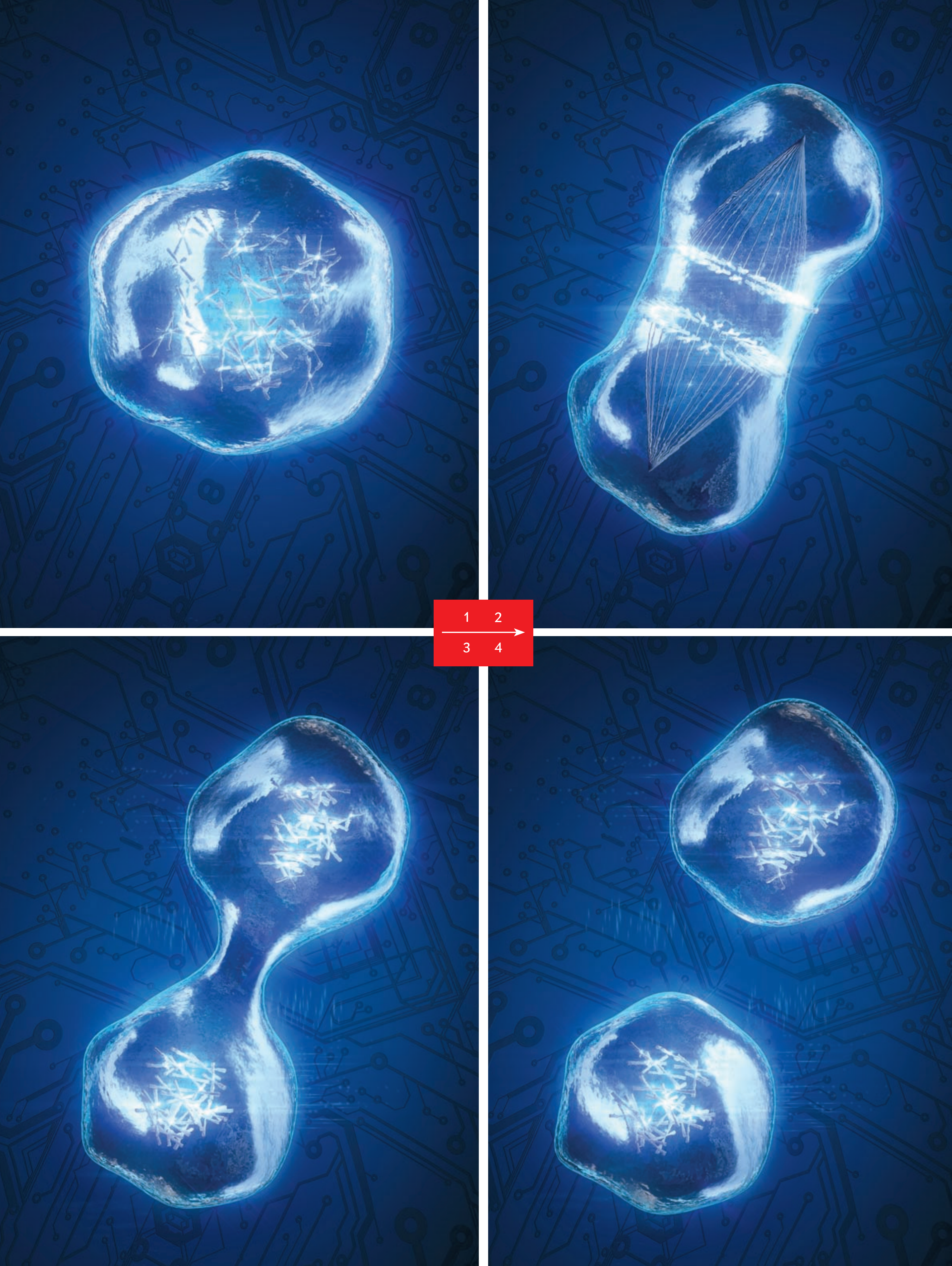
BIOENGINEERING

SIM(U)^{2}LATING
A + LIVING
CELL

In creating the first complete computer model of an entire single-celled organism, biologists are forging a powerful new kind of tool for illuminating how life works

By Markus W. Covert

Illustration by André Kutscherauer


1
2
3
4

Markus W. Covert is an assistant professor of bioengineering at Stanford University, where he directs a laboratory devoted to systems biology.

THE CRUCIAL INSIGHT CAME TO ME AS I LEISURELY RODE MY BIKE HOME FROM WORK. It was Valentine's Day, 2008. While I cruised along, my mind mulled over a problem that had been preoccupying me and others in my field for more than a decade. Was there some way to simulate life—including all the marvelous, mysterious and maddeningly complex biochemistry that makes it work—in software?

A working computer model of living cells, even if it were somewhat sketchy and not quite accurate, would be a fantastically useful tool. Research biologists could try out ideas for experiments before committing time and money to actually do them in the laboratory. Drug developers, for example, could accelerate their search for new antibiotics by homing in on molecules whose inhibition would most disrupt a bacterium. Bioengineers like myself could transplant and rewire the genes of virtual microorganisms to design modified strains having special traits—the ability to fluoresce when infected by a certain virus, say, or perhaps the power to extract hydrogen gas from petroleum—without the risks involved in altering real microbes. Eventually, if we can learn to make models sophisticated enough to simulate human cells, these tools could transform medical research by giving investigators a way to conduct studies that are currently impractical because many kinds of human cells cannot be cultured.

But all that seemed like a pipe dream without a practical way to untangle the web of interlinked chemical reactions and physical connections that make living cells tick. Many previous attempts, by my lab at Stanford University as well as others, had run into roadblocks; some had failed outright.

But as I pedaled slowly through the campus that winter evening, I thought about the work I had been doing recently to record images and video of single living cells. That's when it hit me—a way to make a realistic, functional simulator: choose one of the simplest single-celled microbes out there, a bacterium called *Mycoplasma genitalium,* and build a model of an individual germ. Limiting the simulation to just one cell would simplify the problem enough that we could, in principle, include every bit of biology known to occur in that cell—the unwinding of every rung of its twisted DNA ladder, the transcription of every message in that DNA into an RNA copy, the manufacture of every enzyme and other protein made from those RNA instructions, and the interactions among every one of those actors and many others, all building to cause the cell to grow and eventually divide into two "daughters." The simula-

IN BRIEF

Computer models that can account for the function of every gene and molecule in a cell could revolutionize how we study, understand and design biological systems.

A comprehensive simulation of a common infectious bacterium was completed last year and, while still imperfect, is already generating new discoveries.

Scientists are now building models of more complex organisms. Their long-term goal is to simulate human cells and organs in comparable detail.

tion would generate, nearly from first principles, the entire drama of single-celled life.

Previous attempts had always tried to simulate a whole colony of cells because that is how almost all the data we have on cell behavior were collected: from populations, not individuals. Advances in both biotechnology and computing, however, had started to make single-cell studies much easier to do. Now, I realized, the tools were at hand to try a different approach.

Ideas whirred around in my head. The minute I reached home, I started sketching out plans for a simulator. The next morning, I began writing software code for just a couple of the many, many distinct processes that go on in a living microorganism. Within a week, I had completed several prototype modules, each one a software representation of a particular cellular process. The modules were producing output that looked fairly realistic.

I showed the work to a handful of other biologists. Most of them thought I was nuts. But I felt I was on to something, and two exceptional and daring graduate students, Jonathan R. Karr and Jayodita C. Sanghvi, saw enough potential in the approach that they agreed to work with me on the project.

Completing this model would mean creating dozens of such modules, combing through nearly 1,000 scientific articles for biochemical data, and then using those values to constrain and tweak thousands of parameters, such as how tightly enzymes bind to their target molecules and how often DNA-reading proteins bump one another off the double helix. I suspected that, even with the diligent help of collaborators and graduate students, the project would take years—but I also had a hunch that, at the end, it would work. There was no way to know for sure, except to try.

A GRAND CHALLENGE

AS WE SET OUR SIGHTS on summiting this mountain, we took inspiration from the researchers who first dreamed of modeling life. In 1984 Harold Morowitz, then at Yale University, laid out the general route. He observed at the time that the simplest bacteria that biologists had been able to culture, the mycoplasmas, were a logical place to start. In addition to being very small and relatively simple, two species of *Mycoplasma* cause disease in humans: the sexually transmitted, parasitic germ *M. genitalium,* which thrives in the vaginal and urinary tracts, and *M. pneumoniae,* which can cause walking pneumonia. A model of either species could be quite medically useful, as well as a source of insight into basic biology.

The first step, Morowitz proposed, should be to sequence the genome of the selected microbe. J. Craig Venter and his colleagues at The Institute for Genome Research (TIGR) completed that task for *M. genitalium* in 1995; it has just 525 genes. (Human cells, in contrast, have more than 20,000.)

I was a graduate student in San Diego when, four years later, the TIGR team concluded that only 400 or so of those genes are essential to sustain life (as long as the microbes are grown in a rich culture medium). Venter and his co-workers went on to found Celera and race the federal government to sequence the human genome. They synthesized the essential genes of one *Mycoplasma* species and showed they functioned in a cell.

To me and other young biologists in the late 1990s, this gang was Led Zeppelin: iconoclastic, larger-than-life personalities playing music we had never heard before. Clyde Hutchinson, one of the biologists in Venter's band, said that the ultimate test of our understanding of simple cells would come when someone modeled one in a computer. You can build a functional cell in the lab by combining pieces without understanding every detail of how they fit together. The same is not true of software.

I showed the sample code to a handful of other biologists. Most of them thought I was nuts. But I felt I was on to something.

Morowitz, too, had called for building a cell simulator based on genome data from *Mycoplasma*. He argued that "every experiment that can be carried out in the lab can also be carried out on the computer. The extent to which these [experimental and simulation results] match measures the completeness of the paradigm of molecular biology"—our working theory of how the DNA and other biomolecules in the cell interact to yield life as we know it. As we put the puzzle together, in other words, it becomes more obvious which pieces and which interactions our theory is missing.

Although high-throughput sequencers and robotic lab equipment have greatly accelerated the search for the missing pieces, the floods of DNA sequences and gene activity patterns that they generate do not come with explanations for how the parts all fit together. The pioneering geneticist Sydney Brenner has called such work "low-input, high-throughput, no-output" biology because too often the experiments are not driven by hypotheses and yield disappointingly few insights about the larger systems that make life function—or malfunction.

This situation partly explains why, despite headlines regularly proclaiming the discovery of new genes associated with cancer, obesity or diabetes, cures for these diseases remain frustratingly elusive. It appears that cures will come only when we untangle the dozens or even hundreds of factors that interact, sometimes in unintuitive ways, to cause these illnesses.

The pioneers of cell modeling understood that simulations of whole cells that included all cellular components and their webs of interactions would be powerful tools for making sense of such jumbled, piecemeal data. By its nature, a whole-cell simulator would distill a comprehensive set of hypotheses about what is going on inside a cell into rigorous, mathematical algorithms.

The cartoonlike sketches one often sees in journal articles showing that factor X regulates gene Y … somehow … are not nearly precise enough for software. Programmers express these processes as equations—one of the simpler examples is $Y = aX + b$—even if they have to make educated guesses as to the values of variables such as a and b. This demand for precision ultimately reveals which laboratory experiments must be done to fill holes in knowledge of reaction rates and other quantities.

At the same time, it was clear that once models had been verified as accurate, they would take the place of some experiments, saving the costly “wet” work for questions not answerable by simulations alone. And simulated experiments that generated surprising results would help investigators to prioritize their research and increase the pace of scientific discovery. In fact, models offered such tempting tools for untangling cause and effect that, in 2001, Masaru Tomita of Keio University in Japan called whole-cell simulation “a grand challenge of the 21st century.”

As I flipped through the plots and visualizations, my heart began to race. The model was up and running. What would it teach us?

When still a graduate student, I was impressed by the early results of the leading cell modelers of the time [*see box on opposite page*], and I became obsessed with this grand challenge. Even as I set up my own lab and focused on developing techniques for imaging single cells, the challenge remained in my thoughts. And then, on that February bicycle ride home, I saw a way to meet it.

TWO CRUCIAL INSIGHTS

IT WAS CLEAR that before we could simulate the life cycle of a microbial species accurately enough to mimic its complex behaviors and make new discoveries in biology, we would have to solve three problems. First, we needed to encode all the functions that matter—from the flow of energy, nutrients and reaction products through the cell (that is, its metabolism), to the synthesis and decay of DNA, RNA and protein, to the activity of myriad enzymes—into mathematical formulas and software algorithms. Second, we had to come up with an overarching framework to integrate all these functions. The final problem was in many ways the hardest: to set upper and lower limits for each of the 1,700-odd parameters in the model so that they took on values that were biologically accurate—or at least in the right ballpark.

I understood that no matter how exhaustively we scrutinized the literature about *M. genitalium* and its close relations for those parameters (Karr, Sanghvi and I eventually spent two years culling data from some 900 papers), we would have to make do in some cases by making educated guesses or by using results from experiments on very different kinds of bacteria, such as *Escherichia coli,* to obtain certain numbers, such as how long RNA transcripts hang around in the cell, on average, before enzymes rip them apart to recycle their pieces. Without a way to constrain and check those guesses, we had no hope of success.

In that aha! moment in 2008, I had realized that modeling a single cell—rather than a bunch of cells, as almost all previous studies had done—could give us that constraint we needed. Consider growth and reproduction. A large population of cells grows incrementally; the birth or death of an individual cell does not change things much. But for a single cell, division is a very dramatic event. Before it splits in two, the organism has to double its mass—and not just its overall mass. The amounts of DNA, cell membrane and every kind of protein needed for survival must each double. If the scope of the model is constrained to a single cell, the computer can actually count and track every molecule during the entire life cycle. It can check whether all the numbers balance as one cell becomes two.

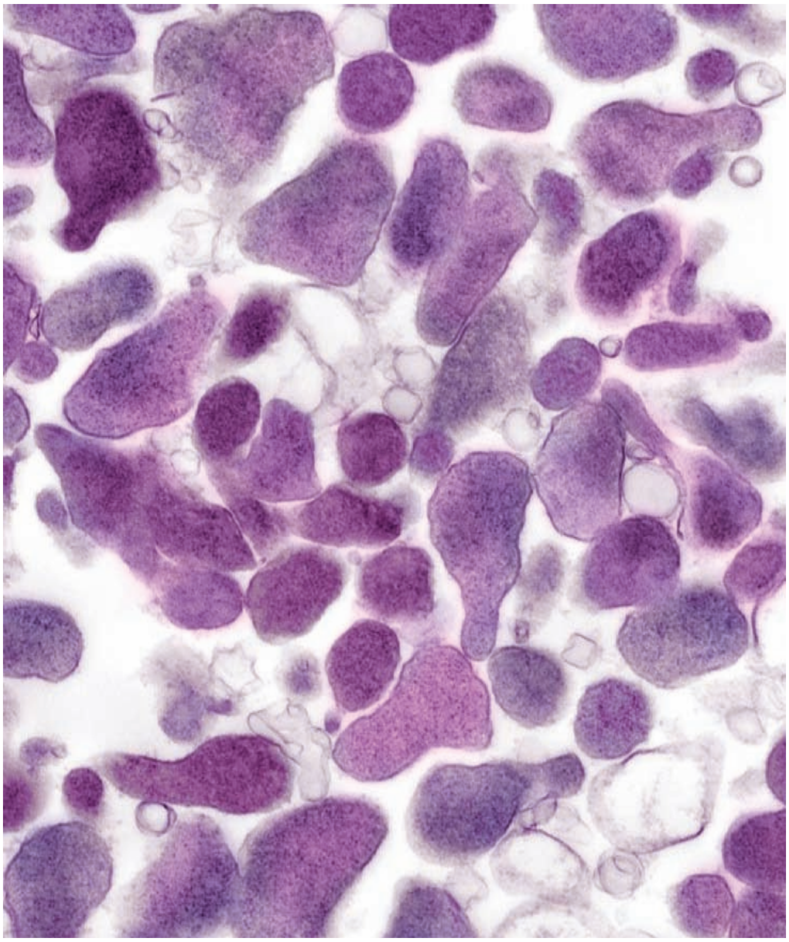
Moreover, a single cell reproduces at essentially a set pace. *M. genitalium,* for example, typically divides every nine to 10 hours in a normal lab environment. It rarely takes fewer than six hours or more than 15. The requirement that the cell must duplicate all of its contents on this strict schedule would allow us to choose plausible ranges for many variables that would otherwise have been indeterminate, such as those that control when replication of the DNA begins.

I put together a team of physicists, biologists, modelers and even a former Google software engineer, and we discussed what mathematical approaches to use. Michael Shuler, a biomedical engineer at Cornell University who was a pioneer in cell simulation, had built impressive models from ordinary differential equations. Bernhard Palsson, under whom I studied in San Diego, had developed a powerful technique, called flux-balance analysis, that worked well for modeling metabolism. But others had shown that random chance is an important element in gene transcription, and cell division obviously involves a change in the geometry of the cell membrane; those other methods would not address these aspects. Even as a grad student, I had realized that no one technique could model all the functions of a cell; indeed, my dissertation had demonstrated a way to link two distinct mathematical approaches into a single simulator.

We decided, therefore, to create the whole-cell model as a collection of 28 distinct modules, each of which uses the algorithm that best suits the biological process and the degree of knowledge we have about it [*see box on page 50*]. This strategy led to a patchwork collection of mathematical procedures, however. We needed to sew them all together somehow into one cohesive whole.

Milestones in Modeling Cells

The long path to the author's first working model of a single cell of a simple bacterium, *Mycoplasma genitalium,* was informed by the theoretical, genetic and modeling efforts of other researchers. Designing a computer model of a human cell is sure to be harder still, given the far greater complexity of mammalian cells. Human cells, for example, contain nearly 40 times as many genes, and those genes are packed into sets of chromosomes that are far more intricate in their physical structure and in the patterns of information they contain. Some critical intermediate steps that need to be accomplished are listed at the bottom right.



SINGLE-CELLED BACTERIUM *Mycoplasma genitalium (purple bodies)* is about as simple as life gets. Yet modeling its life cycle was no easy task.

SCIENCE SOURCE

1967

Francis Crick and Sydney Brenner formulate and propose "Project K: 'The Complete Solution of *E. coli,*'" an effort to figure out the "design" of this common gut bacterium, including fine details of its genetics, energy processing and reproduction.

1984

Harold Morowitz, then at Yale University, outlines a plan to sequence and then model a *Mycoplasma* bacterium.

1984

A team led by Michael Shuler of Cornell University presents a computer model that uses differential equations to capture most of the major biological processes involved in the growth of a single cell of *Escherichia coli.* The model was not able to include gene-level activity, because the *E. coli* genome had not yet been sequenced.

1989–1990

Bernhard Palsson of the University of Michigan releases a comprehensive model of the metabolism of the human red blood cell that includes the effects of pH variation and low blood glucose.

1995

J. Craig Venter of TIGR and his colleagues complete the genome sequence of *M. genitalium.*

1999

Masaru Tomita and his teammates at Keio University in Japan construct E-Cell, a cell-modeling system based on differential equations that includes 127 genes, most of them from *M. genitalium.*

2002

The Alliance for Cellular Signaling, a large collaboration of about 50 researchers, launches an ambitious 10-year, $10-million effort to model mouse B cells of the immune system and heart muscle cells. The project generates some exciting data sets but encounters difficulties manipulating B cells in culture.

2002

Palsson, George Church of Harvard University and Covert, along with several others, complete a genome-scale model of the metabolism of *Helicobacter pylori,* a bacterium that infects humans and can cause stomach ulcers and stomach cancer.

2004

Palsson and Covert, along with three others, publish a computational model of all 1,010 genes involved in regulating the metabolism and DNA transcription of *E. coli* and show that the model accurately predicts the results of lab experiments on real bacteria.

2012

Covert and his co-workers publish a whole-cell model of *M. genitalium* that, for the first time, simulates all the genes and known biochemical processes in a self-reproducing organism.

2013

Covert and his colleagues show that the model accurately predicts the activity of several enzymes.

WHAT'S NEXT

- Complete a whole-cell model for a more typical, better-studied bacterium, such as *E. coli.*
- Model a single-celled eukaryote, such as the yeast *Saccharomyces cerevisiae.* In a eukaryote, the DNA is packaged inside a membrane-bound nucleus, not free-floating as it is in a bacterium.
- Build a model of an animal cell that can be easily cultured, such as a macrophage (a kind of immune cell) from a mouse.
- Construct a first-draft model of a human cell—again, probably a macrophage.
- Model other kinds of human cells, especially those that play the most important roles in common diseases.

INSIDE A VIRTUAL CELL

The Simulator at Work

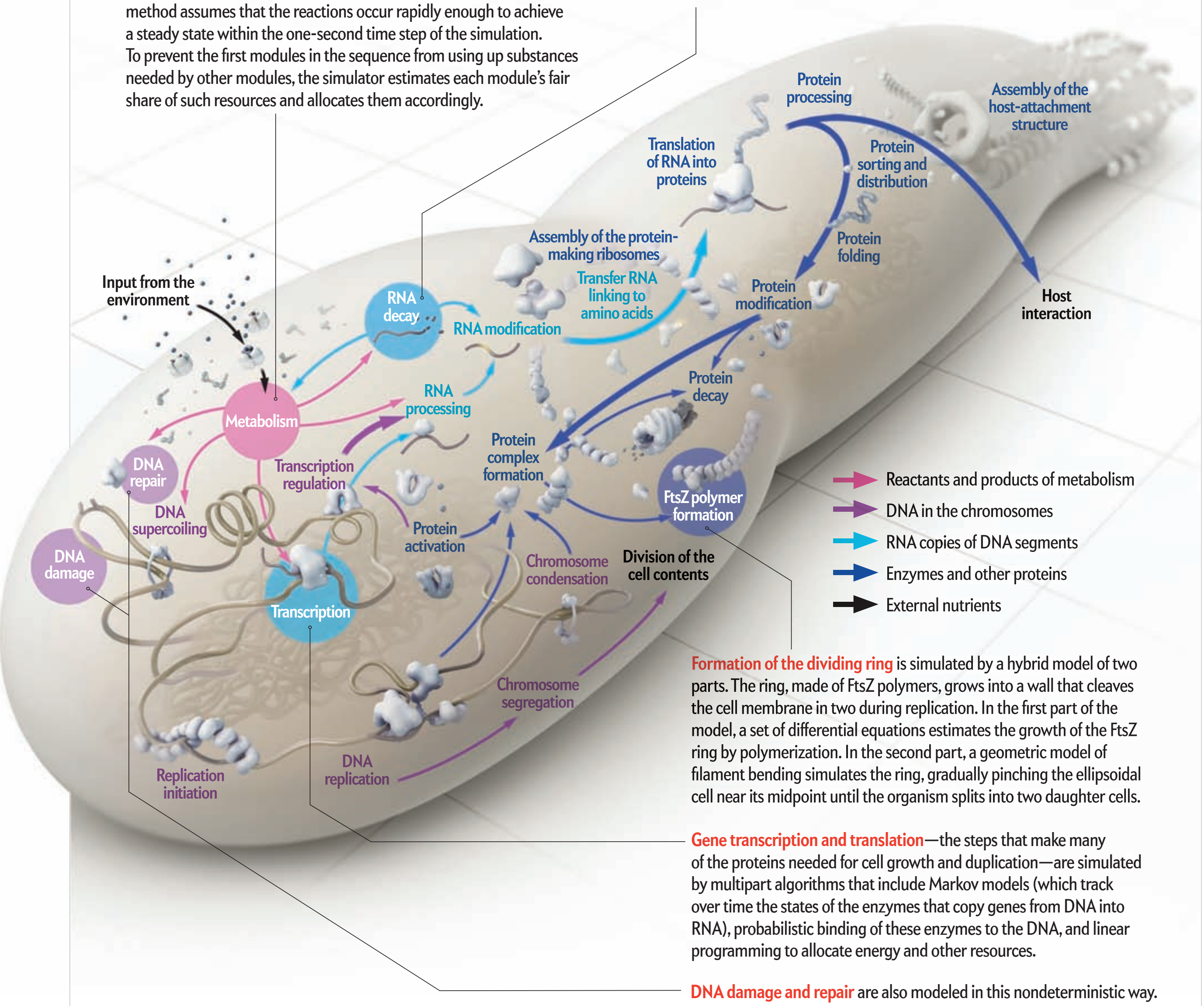
The author's computer model of the infectious bacterium *Mycoplasma genitalium* represents almost every aspect of the life, growth and replication of this microbe. No single mathematical approach can simulate every biological function in the cell, so these functions are divided among 28 distinct modules (*labeled in cell below*), which are involved in the processing of DNA (*purple*), RNA (*light blue*), proteins (*dark blue*), and energy, nutrients and waste (*pink*). For each module, the researchers selected whichever mathematical method worked best—several examples are highlighted below.

The program begins with all modules running in a random sequence to simulate one second of real time. Many input values are drawn from a large table of variables representing their initial states, and some values are selected from ranges or probability functions. Researchers can simulate different scenarios by altering the starting configuration.

After the first time step, the program updates the state table to reflect the outputs of all the modules. The sequence then runs again for another one-second time step, updates the cell-state table, and so on. The loop continues until the cell divides successfully, dies or becomes unrealistically old.

Metabolism of energy, nutrients and waste is modeled by using flux-balance analysis, which exploits linear programming techniques to calculate the reaction rates that produce optimal growth, energy production or some other characteristic the modeler chooses. This method assumes that the reactions occur rapidly enough to achieve a steady state within the one-second time step of the simulation. To prevent the first modules in the sequence from using up substances needed by other modules, the simulator estimates each module's fair share of such resources and allocates them accordingly.

Decay and recycling of RNA and protein are modeled by using Poisson processes, which make use of a random-number generator and probability functions to decide whether a particular piece of RNA or protein decays or survives to the next time step.



Formation of the dividing ring is simulated by a hybrid model of two parts. The ring, made of FtsZ polymers, grows into a wall that cleaves the cell membrane in two during replication. In the first part of the model, a set of differential equations estimates the growth of the FtsZ ring by polymerization. In the second part, a geometric model of filament bending simulates the ring, gradually pinching the ellipsoidal cell near its midpoint until the organism splits into two daughter cells.

Gene transcription and translation—the steps that make many of the proteins needed for cell growth and duplication—are simulated by multipart algorithms that include Markov models (which track over time the states of the enzymes that copy genes from DNA into RNA), probabilistic binding of these enzymes to the DNA, and linear programming to allocate energy and other resources.

DNA damage and repair are also modeled in this nondeterministic way.

SOURCE: "A WHOLE-CELL COMPUTATIONAL MODEL PREDICTS PHENOTYPE FROM GENOTYPE," BY JONATHAN R. KARR ET AL., IN *CELL*, VOL. 150, NO. 2; JULY 20, 2012

SCIENTIFIC AMERICAN ONLINE See examples of the simulator's output at ScientificAmerican.com/jan2014/cell-model

Illustration by AXS Biomedical Animation Studio

I thought back to an undergraduate course I had taken on chemical plant design. For the final class project, we used a powerful simulator package called HYSYS to sketch out a large refinery. HYSYS let us design each principal reaction to occur in a separate vessel. Pipes then connected the output of one vessel to the inputs of others. This framework connected many different kinds of chemical operations into an orderly, predictable system.

It occurred to me that this approach, with some modification, might work for our cell simulator if I was willing to make an important, simplifying assumption: that even though all these biological processes occur simultaneously in a living cell, their actions are effectively independent over periods of less than a second. If that assumption was sound, we could divide the life of the cell into one-second ticks of the clock and run each of the 28 modules, in order, for one tick before updating the pool of cell variables. The model would capture all the interconnectedness of biochemistry—the reliance of gene transcription and DNA synthesis on the energy and nucleotides produced by metabolism, for example—but only on timescales greater than one second.

We had no theoretical proof that this would work. It was a leap of faith.

While constructing our virtual cell, we put in software sensors to measure what was going on inside. Every run of the simulator, covering the entire life cycle of a single cell, churned out 500 megabytes of data. The numerical output flowed into a kind of instrument panel—a collection of dozens of charts and visualizations that, when printed, completely filled a binder.

The results were frustrating at first. For months, as we debugged the code, refined the math, and added more and better lab-derived constraints for the parameters, the cell refused to divide or behaved erratically. For a while it produced huge amounts of the amino acid alanine and very little else.

Then, one day, our cybernetic germ reached the end of its cell cycle and divided successfully. Even more exciting, the doubling time was around nine hours, just like that of living *M. genitalium*. Many other readings were still way off, but we felt then that success was within reach.

Months later I was at a two-day conference in Bethesda, Md., when I was called to the hotel's front desk between sessions.

"Dr. Covert? This package came for you."

Back in my room, I peeled open the box and pulled out a binder. As I spent the next hours flipping through hundreds of pages of plots and complex visualizations, my heart began to race. The great majority of the data looked just like one would expect from an actual growing cell. And the remainder was intriguing—unexpected but biologically plausible. That is when I knew we had reached the summit of that mountain that loomed so large years ago. The first computer model of an entire living organism was up and running. What would it teach us?

A WINDOW INTO THE LIFE OF A CELL

AFTER ABOUT A YEAR of applying our new tool, we still see fascinating things every time we peer inside the workings of the virtual microorganism as it handles the millions of details involved in living and reproducing. We found, to our surprise, that proteins knock one another off the DNA shockingly often—about 30,000 times during every nine-hour life cycle. We also discovered that the microbe's remarkably stable doubling period is actually an emergent property that arises from the complex interplay between two distinct phases of replication, each of which independently varies wildly in duration. And the second-by-second records of the cell's behavior have allowed us to explain why it is that the cell stops dividing immediately when certain genes are disabled but reproduces another 10 times before dying when other essential genes are turned off. Those additional rounds of division can happen whenever the cell stockpiles more copies of the protein made from the gene than it needs in one lifetime—the extra is passed on to its descendants, which perish only when the store at last runs out. These initial results are exciting, but we may need years to understand everything that these simulations are telling us about how these microbes, and cells in general, function.

Our work with *M. genitalium* is only the first of many steps on the way to modeling human cells or tissues at the level of genes and molecules. The model that we have today is far from perfect, and mycoplasmas are about as simple as self-sustaining life-forms get. We have made all our simulations, source code, knowledge base, visualization code and experimental data freely available online, and we and other investigators are already working to improve the simulator and extend it to a variety of organisms, such as *E. coli* and the yeast *Saccharomyces cerevisiae,* both of which are ubiquitous in academic and industrial labs.

In these species, the regulation of genes is much more complex, and the location within the cell at which events occur is far more important. When those issues have been addressed, I anticipate that the next target will be a mouse or human cell: most likely a cell, such as a macrophage (an attack cell in the immune system), that can be readily cultured and employed as a source of measurements to both tune and validate the model.

I cannot guess how far we are today from such technology. Compared with bacteria, human cells have many more compartments and exhibit far greater genetic control, much of which remains mysterious. Moreover, as team players within multicellular tissues, human cells interact more intimately with other cell types than bacteria do.

On February 13, 2008, I would have said that we were at least a decade away from the goal of modeling the simplest cell, and I would not have even considered attempting to model anything more complex. Now we can at least conceive of trying to simulate a human cell—if only to see how the software fails, which will illuminate the many things we still need to learn about our own cells. Even that would be a pretty big step. SA

MORE TO EXPLORE

The Dawn of Virtual Cell Biology. Peter L. Freddolino and Saeed Tavazoie in *Cell,* Vol. 150, No. 2, pages 248–250; July 20, 2012.

A Whole-Cell Computational Model Predicts Phenotype from Genotype. Jonathan R. Karr et al. in *Cell,* Vol. 150, No. 2, pages 389–401; July 20, 2012.

Bridging the Layers: Toward Integration of Signal Transduction, Regulation and Metabolism into Mathematical Models. Emanuel Gonçalves et al. in *Molecular Biosystems,* Vol. 9, No. 7, pages 1576–1583; July 2013.

FROM OUR ARCHIVES

Cybernetic Cells. W. Wayt Gibbs; August 2001.