A reprint from

American Scientist
the magazine of Sigma Xi, The Scientific Research Society

Imitation of Life

Brian Hayes

Almost 30 years ago, Harold J. Morowitz, who was then at Yale, set forth a bold plan for molecular biology. He outlined a campaign to study one of the smallest single-celled organisms, a bacterium of the genus *Mycoplasma*. The first step would be to decipher its complete genetic sequence, which in turn would reveal the amino acid sequences of all the proteins in the cell. In the 1980s reading an entire genome was not the routine task it is today, but Morowitz argued that the analysis should be possible if the genome was small enough. He calculated the information content of mycoplasma DNA to be about 160,000 bits, then added:

> Alternatively, this much DNA will code for about 600 proteins—which suggests that the logic of life can be written in 600 steps. Completely understanding the operations of a prokaryotic cell is a visualizable concept, one that is within the range of the possible.

There was one more intriguing element to Morowitz's plan:

> At 600 steps, a computer model is feasible, and every experiment that can be carried out in the laboratory can also be carried out on the computer. The extent to which these match measures the completeness of the paradigm of molecular biology.

Looking back on these proposals from the modern era of industrial-scale genomics and proteomics, there's no doubt that Morowitz was right about the feasibility of collecting sequence data. On the other hand, the challenges of writing down "the logic of life" in 600 steps and "completely understanding" a living cell still look fairly daunting. And what about the computer program that would simulate a living cell well enough to match experiments carried out on real organisms?

Can a computer program reproduce everything that happens inside a living cell?

As it happens, a computer program with exactly that goal was published last summer by Markus W. Covert of Stanford University and eight coworkers. The program, called the WholeCell simulation, describes the full life cycle of *Mycoplasma genitalium*, a bacterium from the genus that Morowitz had suggested. Included in the model are all the major processes of life: transcription of DNA into RNA, translation of RNA into protein, metabolism of nutrients to produce energy and structural constituents, replication of the genome, and ultimately reproduction by cell fission. The outputs of the simulation do seem to match experimental results. So the question has to be faced: Are we on the threshold of "completing" molecular biology?

The Smallest Life Forms

Bacteria of the genus *Mycoplasma* are attractive for experiments of this kind because they are the smallest and arguably the simplest self-replicating organisms. (Viruses are smaller, but they can reproduce only by hijacking the biochemical machinery of a host cell.)

When mycoplasmas were first observed in the 19th century, they were thought to be fungi (hence the prefix *myco-*, from the Greek root μύκης, meaning *fungus*). The organisms are now classified among the bacteria, but they are peculiar members of that kingdom. They lack the rigid cell wall that encases other bacteria, having only a lipid membrane. One consequence is that mycoplasmas are resistant to many antibiotics, notably those that work by interfering with the synthesis of cell wall components. Mycoplasmas cause a number of human ailments as well as diseases of other animals and also plants. Perhaps the best known of the human pathologies is a lung infection sometimes called "walking pneumonia."

M. genitalium, the organism chosen for the Covert group's computer model, has been known to science only since 1980, when it was isolated from a few patients with urethritis. Even among mycoplasmas, *M. genitalium* is a diminutive cell, with a diameter of roughly half a micrometer. The better-known bacterium *Escherichia coli*, by contrast, is two micrometers long, with a volume roughly 50 times as large. *M. genitalium* is also tiny in terms of its genetic complement. The single circular chromosome has 580,076 base pairs of DNA and just 525 identified genes (even fewer than Morowitz estimated). The *E. coli* genome is about 4.6 million base pairs with 4,300 genes.

The compact cells and concise genome of mycoplasmas make them a useful test bed not just for software but also for "wetware" explorations of the minimal apparatus needed to sustain life. One notable experiment of this kind was reported in 2010 by J. Craig Venter, Clyde A. Hutchison III, Hamilton O. Smith and others at the J. Craig Venter Institute. They sequenced the genome of a particular mycoplasma, storing the list of bases as a computer file; then they made a few edits that would serve as an identifiable "watermark"

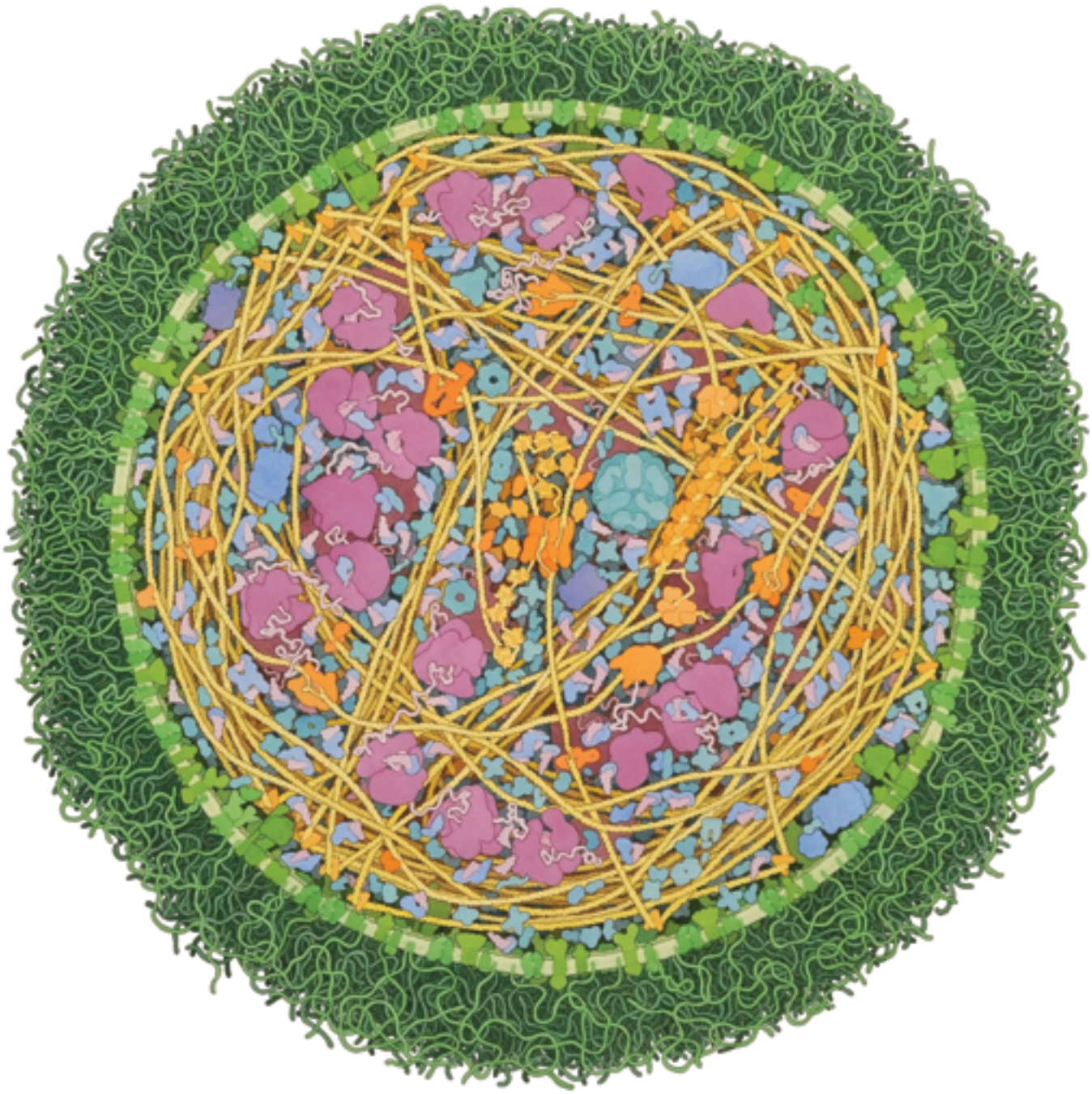
Brian Hayes is senior writer for American Scientist. *Additional material related to the* Computing Science *column appears at http://bit-player.org. Address: 11 Chandler St. #2, Somerville, MA 02144. E-mail: brian@bit-player.org*

and synthesized DNA corresponding to the altered sequence. Finally—and this was the hard part—they inserted the manufactured DNA into cells of another mycoplasma species, replacing the native genetic material. The cells grew and reproduced entirely under the direction of the artificial genome. The experiment can be viewed as a step toward creating a wholly synthetic life form.

In some respects, simulating life with a digital computer is even harder than synthesizing it from chemical components. Given the right ingredients, a biologist might be able to assemble a living cell without fully understanding all the details of how the parts interact. The computer programmer, however, must describe every molecular event.

Modes of Modeling

Building a computer model calls for a multitude of choices and compromises in finding the appropriate level of detail. Take the case of carbohydrate



***Mycoplasma mycoides*, one of the smallest and simplest of free-living organisms, is crammed full of macromolecules and organelles in a watercolor painting by David S. Goodsell of the Scripps Research Institute. The tan, twinelike substance is the DNA of the closed-loop bacterial chromosome. Near the center of the painting is a replication fork, where a DNA polymerase complex *(orange)* is duplicating the cell's genome in preparation for eventual cell division. Due north of the replication fork is one of several RNA polymerase molecules *(also orange)*, where the DNA is transcribed into messenger RNA. Magenta structures are ribosomes, which translate messenger RNA into protein. The dense green mane surrounding the cell consists of carbohydrate chains attached to the lipid membrane. The subject of the recent WholeCell computer model is an even smaller mycoplasma, *M. genitalium*.**

metabolism, in which sugars such as glucose break down to yield water and carbon dioxide. At the most abstract level, this process becomes a single chemical equation:

$$C_6H_{12}O_6 + 6O_2 \rightarrow 6CO_2 + 6H_2O,$$

which doesn't reveal much about what's actually happening inside the cell. A closer look would add dozens of intermediate steps. For example, the six-carbon glucose molecule is first split into two three-carbon pyruvate molecules, liberating energy that can be captured in the phosphate bonds of adenosine triphosphate (ATP).

Adding still more detail leads to a vast web of chemical reactions, as in the famous Metabolic Pathways poster devised by the late Donald E. Nicholson. And one needn't stop there. In principle a simulation could follow every individual molecule—or every atom, for that matter—as it passes through the cellular machinery. The Goldilocks strategy seeks a middle path between bland abstraction and pointless verisimilitude.

The authors of the WholeCell project chose to implement different parts of their model with different levels of detail. Certain key macromolecules are represented as distinct and identifiable entities. Smaller molecules are treated as aggregated quantities; the program keeps track of their numbers but not of their identity as individuals.

The distinction between these two modes of representation can be seen clearly in the sector of the model dealing with protein synthesis. Ribosomes, the large organelles where proteins are assembled, are represented as individuals; each ribosome has its own identity and history. Within the computer program, a separate block of memory is allocated to each ribosome. But the program has no representation for individual molecules of amino acids, the subunits that are linked together to form a protein. Instead the model merely keeps track of the quantity of each type of amino acid. There's a variable for counting all the alanine molecules, another variable for the lysines, and so on.

The WholeCell model is divided into 28 process modules, which correspond to major cellular activities such as replication of the genome, synthesis of protein and repair of damaged DNA. In addition, 16 data structures called state variables record the current status of various subsystems at every instant. The program begins by initializing the state variables to values appropriate to a "newborn" cell, just after cell division. Next, all 28 of the process modules are run for one second of simulated time. At the end of this interval the state variables are updated with the results of the calculations, and then the cycle repeats. The simulation continues until the cell completes its growth and divides. For *M. genitalium* this generation time is typically nine hours, or roughly 32,000 repetitions of the simulation loop. Running time for the program is about the same as the generation time.

The program is written in MATLAB. Source code is available on the project web page at http://wholecell.stanford.edu, along with a knowledge base of quantitative information that went into building the model.

Together with Covert, the principal authors of the software are Jonathan R. Karr and Jayodita C. Sanghvi, who are both graduate students in Covert's group. The model is described in a report published last July in *Cell*; the authors, in addition to Karr, Sanghvi and Covert, are Derek N. Macklin, Miriam V. Gutschow, Jared M. Jacobs and Benjamin Bolival Jr. of Stanford and Nacyra Assad-Garcia and John I. Glass of the Venter Institute.

The Engine Room

In trying to get a sense of how the WholeCell simulation works, I chose three modules for close examination. They are the modules for metabolism, for the transcription of genetic information and for the size and shape of the growing cell.

The metabolism module is where most of the classical biochemistry happens. Here we are in the blue-collar sector of the cell's economy, dealing with energy production, manufacturing and the handling of raw materials and wastes. (Most of the other modules are more concerned with white-collar chores of information processing.)

Even in a cell as tiny as *M. genitalium*, metabolism involves a bewildering maze of interlinked chemical processes. The metabolism module of the WholeCell model includes 104 enzymes, 585 substrates, 441 chemical reactions and 204 transport processes.

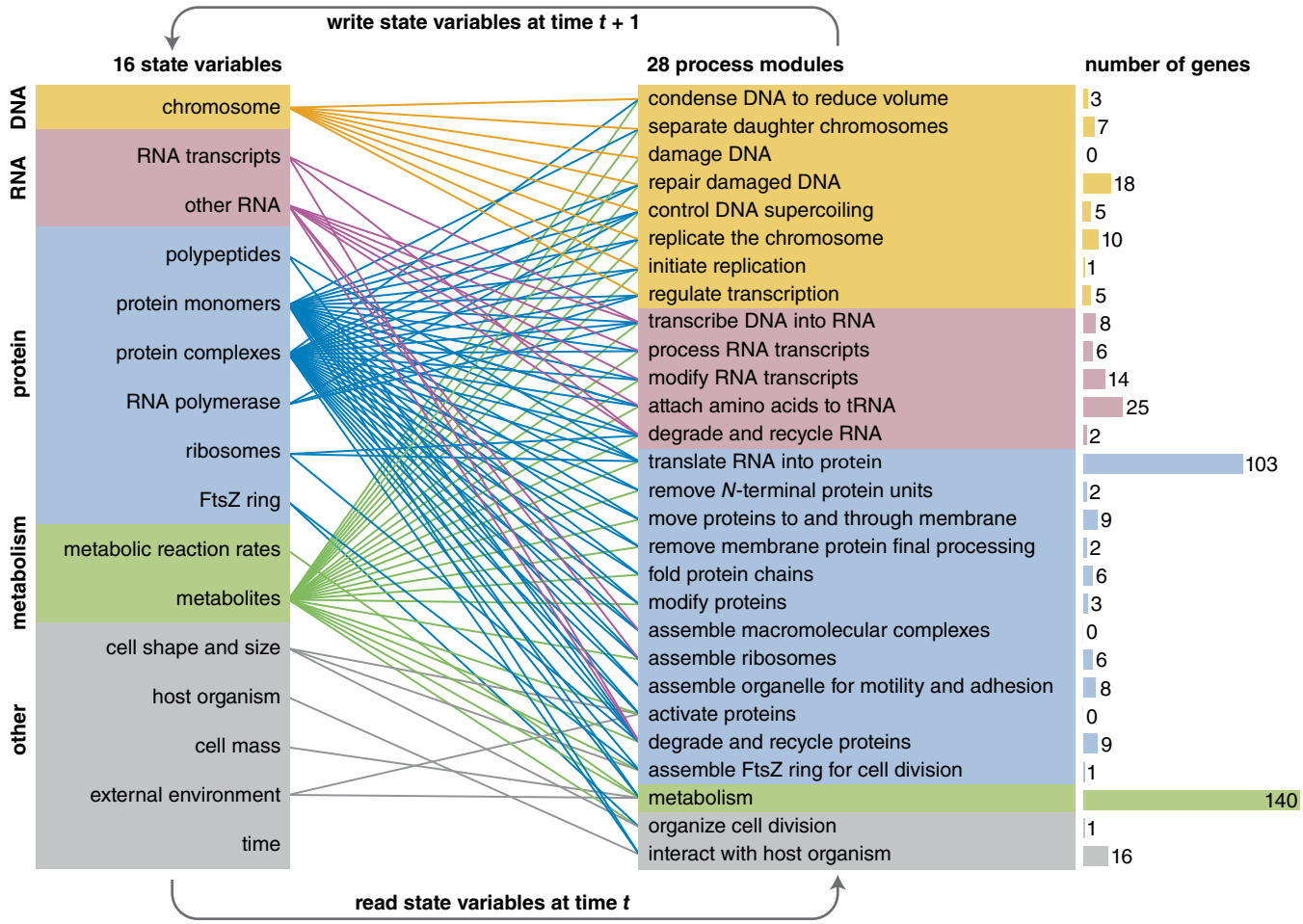
The size and complexity of this network foils many conventional methods of analysis. The rate of any given chemical reaction depends in part on the concentrations of the reactants and the products. But the products of one reaction are the inputs to another, so all the processes are closely coupled and cannot be solved independently. An added complication is that biological networks include cycles, such as the citric acid cycle of carbohydrate metabolism. With a cycle, the products of a given reaction may go all the way around the loop and reappear as inputs to the same reaction, so that the overall flux of material through the network is not uniquely defined.

To sidestep these difficulties, the WholeCell metabolic module relies on a methodology called flux-balance analysis. The underlying idea is that even if a reaction network does not have a *unique* solution, it may well have a *best* solution. The process for finding that solution is much like the algorithm used to optimize the operations of a chemical plant or an oil refinery. Suppose a refinery has a range of products (gasoline, diesel fuel and so on), which differ in manufacturing cost and market value. The mathematical technique of linear programming computes a mix of products that maximizes profit. Applying the same method to the living cell yields a set of reaction rates that make the most efficient use of available resources, such as nutrients. (It's not known with certainty that microorganisms optimize their growth in this way, but it's a plausible assumption in the context of Darwinian natural selection.)

The Scribe

The computations performed in the transcription module are quite different from those of the metabolic subunit. Instead of linear programming, we have discrete events governed by probabilities.

Transcription of a gene begins when a molecule of the enzyme RNA polymerase binds to a chromosomal site called a promoter. The enzyme then ratchets along the double helix, producing a strand of messenger RNA whose sequence is complementary to that of one DNA strand. When the transcript is complete, the polymerase drops off the double helix and releases the RNA. Each step in this process requires a variety of other molecules—initiation factors, elongation factors, termination factors, energy donors—as well as a supply of nucleotides to be incorporated into the growing RNA strand.



The WholeCell model is organized into 16 state variables and 28 process modules. The state variables keep track of the changing status of various aspects of the organism's physiology. A few of the variables, such as mass and time, are simple numerical quantities, but most of the variables are more elaborate data structures; for example, each RNA polymerase molecule and each ribosome has its own individual record. The process modules carry out the actual steps of the simulation, including such major activities as replicating the genome, transcribing DNA into RNA and translating RNA into protein. The metabolism module includes the large network of chemical reactions that supply energy and raw materials. Colored lines indicate which state variables communicate with which processes. The bars at far right show the number of genes that contribute to each module. (Not all of the bacterium's 525 genes are included in this tabulation.) The overall structure of the simulation program is a simple loop: The process modules read the current values of the state variables, calculate what happens during one second of simulated time, and then update the variables. This loop repeats until the life cycle of the bacterium is completed after about nine hours.

In the WholeCell system, each RNA polymerase molecule is an individual object with four possible states: actively transcribing, bound to a promoter region, bound to DNA elsewhere and unbound. Transitions between the states are random events with probabilities calculated to match the experimentally observed distribution. Also, various promoter sites differ in their affinity for RNA polymerase, so the probability of binding is higher in some places than others.

Because of probabilistic events like these, the WholeCell model has an element of nondeterminism. Every run can be expected to produce somewhat different results, even with the same initial conditions and environment. But of course fluctuations and chance events also have a role in real biology; even perfect clones will not follow exactly the same trajectory through life.

Reading the source code of the transcription module gives some vivid glimpses of the subtleties that a wary modeler must keep in mind. Suppose a roll of the digital dice dictates that a certain RNA polymerase molecule is to bind to a promoter site. What happens if all the promoter sites are already occupied? What happens if two polymerase molecules try to grab the same promoter site at the same time? What if two transcription enzymes collide as they move along the DNA? Nature seems to handle such conflicts without having to think about them, but the modeler has to think of everything.

Collisions between enzymes scuttling along the chromosome are not rare events. Results of the WholeCell simulation suggest they happen about once per second, or perhaps 30,000 times in the course of a full cell cycle. The model is therefore equipped with rules to decide who has the right of way.

On Growth and Form

The WholeCell model is not greatly concerned with details of spatial organization. The metabolic module treats the cell as if it were a well-stirred reactor vessel, where all molecules have the same chances of interacting, regardless of their location. Transcription and replication enzymes occupy specific positions along the bacterial

chromosome, but the coordinates are one-dimensional, measured with respect to the linear genetic sequence; they do not define position in three-dimensional space.

Nevertheless, the simulation does include a state variable for cell geometry, which describes the bacterium's shape and eventual fission. Curiously, the shape defined by the simulation is not in fact that of the biological cell. *M. genitalium* is usually described as having a flask or pear shape—a ball with a single asymmetrical appendage. Including this detail would complicate the model without revealing anything of biological significance, so the simulated cell is given a simpler geometry. It begins as a small sphere and elongates into a cylinder with hemispherical end caps. At the end of the life cycle, after the two copies of the genome have migrated to opposite poles of the cell, the middle of the cylinder begins to constrict and then pinches off to form two new cells.

The rules of cell growth are not hard to understand: As the volume of the cytoplasm increases, the enclosing membrane must grow in surface area by a commensurate amount. The mechanics of cell division are more mysterious, but the model nonetheless gives a tentative account. The key component is a protein called FtsZ, which forms a ring girdling the cell in the plane where the two daughter cells ultimately part company.

Fitting an Elephant

The WholeCell model is based on data collected from 900 publications. Some 1,900 numerical values were extracted from these sources to become parameters of the model. This is an impressive compendium, which anchors the simulation in real data.

However, a slate of 1,900 parameters also raises a red flag. If each parameter represents a control knob that can be turned to adjust the model's behavior, then by twiddling enough of the knobs, the output could be "fitted" to just about any desired result. When I asked Covert about this, he immediately cited John von Neumann's quip, "With four parameters I can fit an elephant, and with five I can make him wiggle his trunk." But Covert went on to say that the 1,900 WholeCell parameters have not been used for knob-twiddling or trunk-wiggling. Almost all of the values were taken directly from experimental measurements. They constrain the model rather than adapt it to a preconceived outcome.

Yet that's not quite the end of the story. The data come from many different experiments conducted by different workers over a period of decades. Quite a few parameters come from organisms other than *M. genitalium*, simply because not enough is known about mycoplasma physiology. Given these disparate sources, it's not surprising that the measured parameters are not always consistent. For example, an inventory of cell contents (published by Morowitz 50 years ago) suggested that mycoplasmas have only trace amounts of the amino acid cysteine, whereas analysis of the genome showed a notably greater need for cysteine in mycoplasma proteins. Such inconsistencies must be reconciled if the simulation is to succeed.

Covert and his colleagues tackled this problem by formulating a system of constraints, then searching for parameter values that satisfy the constraints while deviating as little as possible from the measured values. Initially they tried formal optimization algorithms, but these methods failed to converge on a feasible solution. They therefore adopted a heuristic approach, starting from the parameters that are deemed most reliable. Some such reconciliation procedure will remain necessary until more complete and accurate biochemical data become available.

In the meantime, the simulations reported in the *Cell* paper do give physiologically plausible results. The duration of the cell cycle, the rate of growth in biomass and the concentrations of various metabolites are all reasonably close to values measured in real cells. Further support for the model's robustness comes from a series of "knockout" experiments, in which single genes are deleted from the chromosome. After multiple model runs, a gene is classified as essential if losing it compromises viability. The simulation results agree with *in vivo* experiments on 79 percent of the genes.

Still another finding extends and explains known results. The mycoplasma cell cycle has an early phase of genome replication, in which the binding of enzymes initiates the process, and a later phase, in which the replication itself proceeds. Each of these phases varies in length, and yet their sum—the length of the overall cycle—shows comparatively little variation. Examination of the internal details of the model revealed the cause of this odd behavior. The nucleotides needed to synthesize the new chromosome are manufactured throughout the cell lifetime. If the early stage of replication is brief, the later stage is slowed by a shortage of nucleotides. If the early stage is prolonged, the stockpile of nucleotides is sufficient to support full-speed replication.

Reductionism Redux

The idea of building artificial life forms, whether in software or in synthetic cytoplasm, has always been controversial. Mary Shelley, almost 200 years ago, wrote a deep meditation on this theme: *Frankenstein, or the Modern Prometheus*. In Shelley's time the debate was framed in terms of vitalism versus mechanism. The vitalists argued that living things are distinguished from inorganic matter by some "spark of life" or animating principle. The opposing mechanist view had its greatest early champion in René Descartes, who compared animals to clockwork automata.

Within the world of science, the doctrine of vitalism is long dead, and yet there is still resistance to the idea that life is something we can fully comprehend by disassembling an organism and cataloging its component parts. In the brash early years of molecular biology, DNA was "the blueprint of life," a full set of instructions for building a cell. The core process of life was seen as symbol manipulation, a matter of pairing G with C and A with T, then mapping the 4-letter alphabet of nucleotides into the 20-letter alphabet of amino acids. If only we could learn to read the blueprints and decipher the genetic messages, we would know everything about how life works. Now that we read DNA sequences quite fluently, it seems clearer that there's more to life than the "central dogma" of molecular biology.

The idea of simulating a living cell with a computer program stands in the crossfire of this argument between reductionism and a more integrative vision of biology. On one hand, the WholeCell project makes abundantly clear that the DNA sequence by itself is not the master key to life. Even though the transfer of information from DNA to RNA to protein is a central element of the model, it is *not* handled as a simple

mapping between alphabets. The emphasis is on molecules, not symbols.

On the other hand, the very attempt to build such a model is a declaration that life is comprehensible, that there's nothing supernatural about it, that it can be reduced to an algorithm—a finite computational process. Everything that happens in the simulated cell arises from rules that we can enumerate and understand, for the simple reason that we wrote those rules.

I would love to believe that the success of simulation methods in biology might forge a new synthesis and put an end to philosophical bickering over these questions. I'm not holding my breath.

Bibliography

Barile, M. F., and S. Razin (eds). 1989. *The Mycoplasmas*. New York: Academic Press.

Bonarius, H. P. J., G. Schmid and J. Tramper. 1997. Flux analysis of underdetermined metabolic networks: The quest for the missing constraints. *Trends in Biotechnology* 15:308–314.

Brenner, S. 2010. Sequences and consequences. *Philosophical Transactions of the Royal Society of London* 365:207–212.

Covert, M. W., et al. 2001. Metabolic modeling of microbial strains *in silico*. *Trends in Biochemical Sciences* 26:179–186.

Covert, M. W., N. Xiao, T. J. Chen and J. R. Karr. 2008. Integrating metabolic, transcriptional regulatory and signal transduction models in *Escherichia coli*. *Bioinformatics* 24:2044–2050.

Crick, F. H. C. 1973. Project K: The complete solution of *E. coli*. *Perspectives in Biology and Medicine* 17:67–70.

Gibson, D. G., et al. 2010. Creation of a bacterial cell controlled by a chemically synthesized genome. *Science* 329:52–56.

Karr, J. R., et al. 2012. A whole-cell computational model predicts phenotype from genotype. *Cell* 150:389–401. Supplemental material: http://dx.doi.org/10.1016/j.cell.2012.05.044. Source code: http://wholecell.stanford.edu

Lerman, J. A., et al. 2012. In silico method for modelling metabolism and gene product expression at genome scale. *Nature Communications* 3:929.

Loeb, J. 1912. *The Mechanistic Conception of Life: Biological Essays*. Chicago: University of Chicago Press.

Morowitz, H. J. 1984. The completeness of molecular biology. *Israel Journal of Medical Sciences* 20:750–753.

Orth, J. D., I. Thiele and B. Ø. Palsson. 2010. What is flux balance analysis? *Nature Biotechnology* 28:245–248.

Suthers, P. F., et al. 2009. A genome-scale metabolic reconstruction of *Mycoplasma genitalium*, iPS189. *PLoS Computational Biology* 5:e1000285.

Tomita, M., et al. 1999. E-CELL: Software environment for whole-cell simulation. *Bioinformatics* 15:72–84.

