

Biology software promises easier way to program living cells

'Cello' automates the fast, reliable design of DNA-based logic circuits.

Erika Check Hayden

31 March 2016

Synthetic biologists have created software that automates the design of DNA circuits for living cells.

The aim is to help people who are not skilled biologists to quickly design working biological systems, says synthetic biologist Christopher Voigt at the Massachusetts Institute of Technology in Cambridge, who led the work. "This is the first example where we've literally created a programming language for cells," he says.

In the new software — called [Cello](#) — a user first specifies the kind of cell they are using and what they want it to do: for example, sense metabolic conditions in the gut and produce a drug in response. They type in commands to explain how these inputs and outputs should be logically connected, using a computing language called Verilog that electrical engineers have long relied on to design silicon circuits. Finally, Cello translates this information to design a DNA sequence that, when put into a cell, will execute the demands.

Voigt says his team is writing user interfaces that would allow biologists to write a single program and be returned different DNA sequences for different organisms. Anyone can access Cello through a [Web-based interface](#), or by downloading its [open-source code from the online repository GitHub](#).

"This paper solves the problem of the automated design, construction and testing of logic circuits in living cells," says bioengineer Herbert Sauro at the University of Washington in Seattle, who was not involved in the study. The work is published in *Science*.¹

Working together

Creating Cello required a decade of labour, says Voigt. The hard part, he says, was not writing the software itself but working out how to make biological parts — logic gates, by analogy with electronic circuits — that reliably worked together to carry out the functions programmed into the circuit by Verilog. For instance, the team had to develop a combination of genetic components that work together as an 'insulator' — ensuring that each biological part works no matter where in the DNA sequence it is placed.

How to design reliable, complex biological computing systems is a [central problem of synthetic biology](#), Voigt says. Researchers found that DNA-based analogues of electronic switches and transistors would work in simple cases but would often fail when organized into more complex circuits. But inexpensive gene-synthesis technologies and the use of fast, cheap genetic sequencing to look at what goes wrong have enabled [huge advances](#) in understanding.

"What we're finding over time is that biology isn't this kind of mysterious unpredictable substrate; it just felt that way because we didn't really have the tools to see what was going on," Voigt says.

Voigt's team tested 60 designs made using Cello; 45 worked correctly the first time. He estimates that it would take about a week to design 60 biological circuits with Cello; by contrast, it took a postdoc three years to design, test and build one successful biological circuit for a paper published in 2012, he says².

Synthetic biologist Adam Arkin at the University of California, Berkeley, who was not involved in the work, says that Cello is one of [a series of steps](#) that aim to push synthetic biology closer to meeting its founding goals of using the principles of engineering to enable the design of new biological circuits.

"What is wonderful is seeing the original conception of synthetic biology — to build the infrastructure to make the engineering of new



KTS Design/Getty Images

Synthetic biologists program cells with DNA-based logic circuits.



Living factories of the future

biological function vastly more efficient, predictable, transparent and safe — come to fruition in such powerful computational tools and biological reagents,” Arkin says.

Nature | doi:10.1038/nature.2016.19671

References

1. Nielsen, A. A. K. *et al.* *Science* <http://dx.doi.org/10.1126/science.aac7341> (2016).
2. Moon, T. S. *et al.* *Nature* **491**, 249–253 (2012).