Biology enters 'The Matrix' through new computer language

July 23 2008

Ever since the human genome was sequenced less than 10 years ago, researchers have been able to access a dizzying plethora of genomic information with a simple click of a mouse. This digitizing of genomic data—and its public access—is something that would have been unthinkable a generation earlier.

But as molecules go, DNA is pretty straight forward. With its simple composition and linear structure, it easily lends itself to mathematical models. Not so with proteins. In fact, proteins are an order of magnitude more complex than DNA. It is proteins, not DNA, that carry out the cell's heavy lifting. However, with their intricately folded three-dimensional shapes determining a seemingly endless range of possible functions and their manifold interactions with other proteins and with DNA, the leg-work required to mathematically capture the protein universe seems absurd.

And it is.

That is why a team of Harvard Medical School researchers have decided to attack this issue from an entirely new angle. Rather than build a mountain range of proteomic data one grain of dirt at a time, they have developed a computer program that can take on the responsibility of assembling such a gargantuan model.

Enter Little b, a computational language that can penetrate the "mind" of a cell.

"Through incorporating principles of engineering, we've developed a language that can describe biology in the same way a biologist would," says Jeremy Gunawardena, director of the Virtual Cell Program in Harvard Medical School's department of systems biology. "The potential here is enormous. This opens the door to actually performing discovery science, to look at things like drug interactions, right on the computer."

These findings will be published in the July 23 issue of *Journal of the Royal Society Interface*.

Most current computational methods of modeling biological systems are not unlike writing a document with pen and paper. Each new project starts from scratch; there are no facilities for cutting and pasting, for linking to other texts, for including images, etc.—things that come so "naturally" to electronic documents.

Harvard Medical School researcher Jeremy Gunawardena, a mathematician by training, teamed up with Aneil Mallavarapu, a cell biologist and computer scientist, to lead a project that would bypass these limitations.

"We knew that the secret to doing this would be to assimilate fundamental concepts of engineering, concepts like modularity and abstraction, into the biological realm," says Mallavarapu, who was recently awarded the Merrimack prize by the Council for Systems Biology in Boston for developing this program.

Modularity involves breaking a problem down into separate modules and constructing each module so that it can interact with the others. Abstraction refers to extracting generic biological properties and incorporating them into the modules, so that they can use this abstract information in concrete contexts. Put another way, abstraction means that, unlike the old days of pen and paper, each new model does *not*

need to be built from scratch. Models can be built upon each other and their individual modules refined and re-used.

To do this, Mallavarapu used the programming language LISP, a language widely used in artificial intelligence research. LISP is famous among computer scientists due to its ability to write code that, in turn, can write code, enabling a programmer to derive new mini-languages.

"LISP isn't like typical programs, it's more like a conversation," says Gunawardena. "When we input data into Little b, Little b responds to it and reasons over the data."

For example, Gunawardena's lab works on kinases, a kind of protein that transfers phosphate chemicals to other proteins in order to regulate their activity. While this property is common to all kinases, there is a great deal of variety in how particular kinases carry this out. Little b, however, understands this basic property of kinases, this abstraction.

Here, the researchers demonstrated how they were able to interact with Little b to build complex models of kinase activity, using Little b as a kind of scientific collaborator, and not simply a passive tool.

On a larger scale, the researchers also used the program to query the development of fruit fly embryos. As a result, they discovered levels of complexity in these embryonic structures that previous research had missed.

"This language is stepping into an unknown universe, when your computer starts building things for you," says Gunawardena. "Your whole relationship with the computer becomes a different one. You've ceded some control to the machine. The machine is drawing inferences on your behalf and constructing things for you."

The researchers sometimes admit, half-joking, that Little b sometimes feels a little bit like "The Matrix"—referring, of course, to the film trilogy in which human beings lived in a computer-generated virtual world.

Mallavarapu and Gunawardena have a pretty clear vision for this project: they want every biologist in the world to use it.

But in order to bring the program out from the early adopter community, where it is currently being used by colleagues in the Harvard community, it needs to be more accessible.

"The next step is to create an interface that's easy to use," says Gunarwardena. "Think of web page development. Lots of people are creating web pages with little or no knowledge of HTML. They use simple interfaces like Dreamweaver. Once we've developed the equivalent, scientists will be able to use our system without having to learn Little b."

And the more people use it, the smarter it gets. As researchers around the world input their discoveries into Little b, the program will assimilate that information into its language.

The ultimate goal is to have an in silico, virtual cell—a dynamic biological system living in software.

"Sure, it's a long way off," says Gunawardena, "but we're getting there."

Citation: *Journal of the Royal Society Interface*, online publication, July 23, 2008; "Programming with models: modularity and abstraction provide powerful capabilities for systems biology"; Aneil Mallavarapu, Matthew Thomson, Benjamin Ullian, and Jeremy Gunawardena