International Journal of Pharmacy and Analytical Research (IJPAR)

ISSN: 2320-2831

IJPAR |Vol.12 | Issue 2 | Apr - Jun -2023
www.ijpar.com

Review article *Analytical research*

DNA programming in drug discovery and drug development

Dr. Ravi kumar maddali*, Naga kishore R

Principal & Professor, Geethanjali College of Pharmacy

***Corresponding author: Ravi kumar maddali**
Published on: 29.04.2023

ABSTRACT

Important traits that biological cells have that make them appealing for many applications are energy efficiency, self-reproduction, and small size. Examples include the incorporation of intelligence into smart materials, sensing, and nanoscale manufacturing. Simple "genetic circuits" that are encoded in DNA and carry out their functions in the cellular environment have been developed over the past 15 years. A gene circuit uses biological interactions to provide computation, similar to an electronic circuit. Simulators that replicate cell colonies running microbial programmes have been developed at the protein and language levels.

Keywords: DNA, genetic circuits

INTRODUCTION

Biological cells possess important characteristics, such as energy efficiency, self reproduction, and miniature scale that make them attractive for much application. Examples include embedded intelligence in smart medicine, materials, sensing and nanoscale fabrication. Huge numbers of programmed cells executing in parallel will enable computation. Living cells are the ultimate programming substrate. The last 15 years has seen the design of simple "genetic circuits" that are encoded in DNA and perform their function in the cellular milieu. An electronic circuit-like computation is produced by a gene circuit by harnessing biological interactions. For example, a simple NOT gate can be constructed by using a gene that turns off a second gene. When the first gene is on, the second is off, and vice versa. Using more complex strategies, a number of circuits have been built that function like logic gates, oscillators, and memory. The next phase of synthetic biology will be to understand how to connect many such circuits to build programs. DNA defines what enzymes, proteins, and molecules are made inside of a cell, and that DNA code can be programmed to grow new kinds of proteins and molecules that can be used in beauty.

Leonard Adleman is an American computer scientist. One of his contributions to the development of the RSA encryption algorithm earned him the 2002 Turing Award, also known as the Nobel Prize in Computer Science. He also founded the field of DNA computing, which is another accomplishment. Worked on DNA-based computation and the digital nature of DNA's function in processing information. Von Neumann pioneered game theory and, along with Alan Turing and Claude Shannon, was one of the conceptual inventors of the stored-program digital computer. The polymerase chain reaction (PCR), which allows for the rapid and massive duplication of a little amount of DNA, was developed by Kary Mullis in 1985. The two strands of the DNA molecule are split apart by heat, and the DNA building blocks that have been added are attached to each strand.
The significance of this programming did not lie in the sophistication of the problem, but in the fact that it showed that strands of DNA, mixed together in a vial, could be controlled such that their biochemistry could be viewed as a computation. Parallel power of DNA computing: An example of the Hamilton path problem. All physical systems are capable of doing computations; it is up to the investigator to make those computations useful. Since 1994, DNA computation has made significant advancements, and

programmable computers free of moving parts and using just DNA molecules have been created.

DNA-based computing, specifically synthetic biology and DNA self-assembly. As opposed to employing non-biological materials, DNA's stability and digital nature make it possible to design, or should we say programme, structures on a nanometer scale. Synthetic biology, on the other hand, concerns the creation of biological systems from scratch or from components that had other functions.

Organisms can actively restructure their genetic material, believes that such processes should be more widely incorporated into evolutionary thinking. Synthetic biology, on the other hand, concerns the creation of biological systems from scratch or from components that had other functions. The different paths to synthetic biology covers most of the recent advances that have made the headlines. These range from pieces of DNA called 'repressilator' and the 'biobricks' to a way of coaxing yeast to produce the precursor for a malaria drug—a feat for which Jay Keasling, a biochemical engineer at the University of California, Berkeley, won Discover magazine's first 'scientist of the year' award.

BioBrick parts are DNA sequences which conform to a restriction-enzyme assembly standard. These building blocks are used to assemble and design larger synthetic biological circuits from individual parts and combinations of parts with defined functions, which would then be incorporated into living cells such as *Escherichia coli* cells to construct new biological systems. Examples of BioBrick parts include ribosomal binding sites (RBS), coding sequences, promoters and terminators.

Future applications for this type of programming include designing bacterial cells that can generate a cancer drug when they detect a tumor, or creating yeast cells that can stop their own fermentation process if too many toxic byproducts build up. Over the past 15 years, biologists and engineers have designed many genetic parts, such as biological clocks, memory sensors, and switches, that can be combined to modify existing cell functions and add new ones. However, designing each circuit is a laborious process that requires great expertise and often a lot of trial and error. Users of the new programming language, however, need no special knowledge of genetic engineering.

"You could be completely naive as to how any of it works. That's what's really different about this"

You could be a student in high school and go onto the Web-based server and type out the program you want, and it spits back the DNA sequence. To create a version of the language that would work for cells, the researchers designed computing elements such as oxygen and logic gates that can be encoded in a bacterial cell's DNA. The sensors can detect different compounds, such as glucose or oxygen, as well as light, acidity, temperature, and other environmental conditions. Users can also add their own sensors.

Biological circuits

The researchers used this language to design 60 circuits with various purposes, and 45 of them performed as intended the first time they were tested. Many of the circuits were designed to measure one or more environmental conditions, such as oxygen level or glucose concentration, and respond accordingly. An additional circuit was designed to rank three different inputs and they responded accordingly.

Future Applications

Using this method, scientists intend to work on a number of different applications, including yeast that can be engineered to shut off when it is producing too many toxic byproducts in a fermentation reactor and bacteria that can live on plant roots and produce insecticide if they detect an attack on the plant. Bacteria that can be swallowed to help with lactose digestion is also one of the applications they plan to work on. One of the new circuits, which have seven logic gates and over 12,000 base pairs of DNA, is the biggest biological circuit yet constructed. The quickness of this method is an additional benefit. Until now, "it would take years to build these types of circuits. Now you just hit the button and immediately get a DNA sequence to test"

Cellular Gate

A fundamental chemical process in the cell is the production of proteins from genes encoded in the DNA. The cell performs important regulatory activities through DNA-binding proteins that repress the production of particular protein. Proposes using this regulatory mechanism to implement digital logic inverters. This concept can be extended to construct complex digital logic, making the cell a self-contained computational unit.

The Microbial Colony Language is a programming paradigm simple enough for biological cells, yet expressive enough to implement interesting applications. The language makes use of programming techniques that are dependably executed by biological cells. The programme for a single cell consists of chemical diffusion with a limited communication range, boolean state, boolean operations, and event-triggered rules. In computer science, a Boolean is a logical data type that can have only the values true or false. For example, in JavaScript, Boolean conditionals are often used to decide which sections of code to execute (such as in if statements) or repeat (such as in for loops).

Global Trend

The biological advancement is now coming to the forefront of beauty, bringing innovations and opportunities through its ability to program code to grow new kinds of beauty ingredients. Biology operates on a sort of code, which is one of its most amazing features. Biopharmaceutical firms like Biocon Limited and GSK have already started looking for Synthetic Biology-based remedies in India to quickly and cheaply produce innovative medicines. The Department of Biotechnology (DBT), India has recently started a Synthetic Biology training program for postgraduates, Ph.D. Students, Postdocs, and faculty members in 2018.

CONCLUSION

Scientists have implemented language-level and protein-level simulators that model cell colonies executing microbial programs. Simulations demonstrate that these algorithms are capable of producing large-scale pattern formation and coordinated group activity. The effort to develop programmable biological systems is merely a first step in extending the role and nature of computer beyond conventional applications.

REFERENCES

1. Anne Trafton. MIT News. A programming language for living cells. 2016 March 31. Available from: https://news.mit.edu/2016/programming-language-living-cells-bacteria-0331.
2. Biological programming: genesis Machines: the New Science of Biocomputing by Martyn Amos Atlantic Books; 2006. 320 p.
3. Synthetic Biology. Weebly. Available from: https://syntheticbio.weebly.com/biological-programing.html.