[Journal Publication of International Research for Engineering and Management](http://www.joirem.com) (JOIREM)
Volume: 04 Issue: 03 | Mar-2026
ISSN (O) 3107-6696

A LIVING ALTERNATIVE TO MATHEMATICAL HEURISTICS

Reshma Banu S[1], Prannesh U R[2],Mahenthiran R[3], Sudha S S[4]

[1]Post graduate student, Department of microbiology, [3]Assistant professor, Department of microbiology ,[4]Professor, Department of microbiology, Dr. NGP Arts and Science College, Coimbatore

---***---

Abstract- Microbial computation uses the abilities of microorganisms, mainly bacteria and slime molds, to do calculations that would normally be done by machines. Microorganisms have been genetically changed or created by synthetic biology to be able to interpret input and create output through logical circuits. In real life examples of microbial computation, quorum sensing and chemotaxis are being used by microorganisms as a form of computational ability. Physarum polycephalum, for example, has shown how its network of protoplasmic tubes can be used to physically represent solutions to optimization problems . In addition, it has been shown that synthetic biochips engineered microbial colonies can count events, execute Boolean logic, and adjust to their surroundings. By using physiological processes for these types of computations, they end up having low energy requirements, are highly parallel in nature, and self-replicate. Examples of how microbial computations mimic algorithms used in programming include chemical gradient signaling, nutrient driven spatial growth, and stigmergy where microorganisms provide environmental signals to other microorganisms to help in future actions upon that same environment. Using integrated microbial units that consist of shape, genetic control, and metabolic feedback, they serve as an encoder to encode data and can process via morphological computation and give analytical output. Advancements in synthetic biology have provided the foundation for bio circuits with more accurate and scalable capabilities, through the uses of DNA computing and logic gates based on CRISPR technology. Smart therapies, biosensing and diagnostics are just a few of the potential uses for these bio-circuits. Despite these advancements, several challenges persist including slow processing times, inconsistent results and difficulty scaling complexity. In the future bio-hybrid computing will fuse electronic and microbiological systems to create flexible, low power platforms to solve real world problems in logistics, robotics and medicine. In essence, microbial computation redefines intelligence as an emergent quality of life; therefore, microbial computation presents a physical and sustainable approach to traditional computing systems.

Keywords: Logic gates, gene circuits, synthetic biology, microbial computation, and genetic engineering

Introduction:

The basis of a new area of research called microbial computation is the ability of engineered microbe (i.e. bacterium) to perform logical operations similar to electronic circuits. Researchers use synthetic biology to introduce the logic gates of their choice (AND, NOR, etc.) into bacteria such as Escherichia coli or yeast and use those bacteria to process chemical signal inputs and generate outputs (e.g. production of a fluorescently labeled protein). Initial studies have focused on constructing NOT gates and by extension, basic Boolean functions; recent studies have used multiple genetic inputs to create logic circuits that can perform modular processing [1]. As a result of the complexity of assembling logic gates into logic circuits and therefore conducting logical operations, consortiums of two or more bacteria were developed as multicellular computing systems. Each bacterial strain contains one of the logic gates and due to quorum sensing functionality (i.e. quorum sensing molecules act as chemical wires), the bacterial strains communicate with each other. The result is that spatially separated colonies of bacteria on agar plates can perform equivalent and/or logical XOR operations. Coordinating colony patterning with intercellular chemical communication leads to computation of higher order than either could produce separately. The Asha Institute in India is presently conducting research that illustrates this principle, with the development of an artificial neural network-configured bacterial system comprised of 14 different strains of bacteria, each referred to as a "bactoneuron." Bacteria are able to perform many yes/no decisions making problems and optimization problems (e.g., determining how many slices can be made from a pie using straight cuts or determining if a number is prime) based on chemical signals received (as binary digits) and then returned in the form of colored fluorescence (as a response). Examples of how microbial computing can also solve maze like challenges include distributing the solution among the different bacterial types involved by taking advantage of the ability of microorganisms to grow in parallel. An example would be to apply this distributed solution method to solve the 2 x 2 maze mentioned previously. Biocomputers have shown to be extremely energy efficient per mass unit and possess large scale processing power for parallel processing; however, they execute at a slower rate than silicon based systems.

Microbial computation uses the abilities of microorganisms, mainly bacteria and slime molds, to do calculation Using E. coli cells as the basis for a single cell system, scientists were able to create a functional half-adder by layering different types of genetic circuits together and utilizing three different types of logic gates (AND, OR and NOT). By being able to respond to chemical inducers like arabinose and rhamnose, the bacteria were able to perform addition and subtraction on a molecular level (in vivo). Integration issues included promoter interference and mRNA regulation, but reliability of the half-adder was improved through the use of engineering techniques such as ribozymes and CRISPR-based insulators. Another way to solve problems is through the use of DNA strand displacement logic, which is a molecular-level way of doing calculations. Example problems include finding the square root of a decimal number when given as binary. These types of DNA circuits demonstrate the flexibility of biomolecular computation as an alternative to cellular-based biocomputing, but they can take a long time (a few hours) and be complicated to set up. Slime molds (i.e., Physarum polycephalum) have also been used as living computers by scientists, much like bacteria. They will spontaneously reorganize their networks to maximize their connectivity and find the shortest path between points in a maze. Research has also included developing computer models of road networks and using oscillation frequency shifts to create logic gates and have tackled computational problems such as the Traveling Salesman Problem. The basic theory behind Turing-complete capability has been realized through the implementation of Universal Computing Systems using only a minimal number of bacteria and plasmids. For example, using two different bacteria that both have 34 plasmids, one can build a register machine analogous to a computer utilized in logic at either the microbial level or at the plasmid level that will compute virtually any recursive function. Therefore, although a physical implementation is still very challenging, this demonstrates much potential for the use of biological computation. Additionally, bio computation has been utilized in conjunction with the concept of "wetware" or "living" computing where one can build software incorporated into a single organism or system of living. Similar to biological systems, cells provide natural solutions for memory, processing and flexibility through nature. Cells function at very large-scale parallel processing and are capable of self-replicating as well as responding to their environment. Therefore, this can lead to making biological and silicon composite platforms in the future, allowing for greater opportunities to combine both biological and silicon systems to create biocomplexity.

Microorganisms as computational devices

Unconventional computing, in both its physical and biological manifestations, represents a concept in which a non-traditional physical, chemical, or biological system performs computation (e.g., calculation, processing) outside the normal silicon architecture. In this conceptualization, computation derives from the intrinsic dynamics and flows of matter, energy, and living cells rather than from transistors or logic circuits alone. Spontaneous emergence of computation occurs naturally within biological systems, specifically microorganisms, through metabolism, signals and signaling processes, growth, and organization of space. Microorganisms are therefore attractive as biological substrates for this form of computation. Microorganisms possess many advantageous properties as substrates for unconventional computation. First, microorganisms use energy-efficiently and exhibit self-replication: every microbial cell can be grown from a very small biomass (energy) and a population of microbes can scale compute capability by dividing cells. Additionally, when using microbial systems as information processing systems, there is vast evidence of energy efficiencies in that microbial-based distributed compute systems achieve the same level of energy efficiency per-operation as complex electronic computing systems with the additional trait of requiring less than 10-14 to 10-16 of energy per logical operation. While computations are not the primary role of microorganisms, the metabolic expense associated with executing calculations imposes an extraordinarily low (~5-15%) overhead to the baseline metabolic activity of the microorganisms. Second, microorganisms respond to stimuli from their environment in a high degree of sensitivity. Bacteria are able to perceive chemical gradients as well as nutrients, toxins, light, and even temperature via well-studied sensory pathways including quorum sensing and two-component regulatory systems. Once bacteria receive inputs, they can transform those inputs into gene expression, moving through space, and changing their metabolism in such a way that will activate decision-making and logic processing pathways.

Third, bacteria exhibit emergent distributed systems with complex characteristics that develop out of simple rules [5]. The behavior of bacteria can be compared to network optimization and graph theory methods which are represented using physical processes instead of logical computation. Additionally, amorphous computing within microbial communities exhibits the successful combination of a large number of simple, locally interacting agents (cells) into one coherent computer system through the use of gradient signaling and emergent patterns. Microbes offer an attractive alternative to the design of living computers because they can replicate themselves, require little energy, adapt to their environment easily, and offer emergent capabilities. An example of this was engineered bacteria that can generate multilayered Boolean circuits capable of processing multiple input signals and producing output signals that were predefined. Electronic logic circuits also operate much the same as programmable circuitry would function using timed circuits (timers) and memories. Recently, engineered bacterial colonies have been placed on chips utilizing microfluidic arrays. These microfluidic devices work like logic devices by chemically responding to input signals through changes in pH or oxygen. This shift of pH or oxygen indicates the circuit's logic state. Therefore, this technique is the beginning of developing bacterial biochips to use for sensing and diagnostic functions. Additionally, utilizing the natural and inherent computation possibilities of a microorganism's physiology and creating artificial circuit-based microbial computing are equally critical. Gene expression, motility, and biofilm formation result from biological signaling networks such as one and two-component regulatory systems (in bacteria). Therefore, natural circuits can be looked at as being a type of metabolic and transcriptional hardware that processes the information provided by the environment to provide optimal means of survival and an ability to adapt to the changing environment. Simply stated, biology has evolved computation for living organisms rather than just transferring of information. Given the large number of cells (i.e., billions) that colonies of microorganisms have, there is a great potential for a high degree of parallelism in computation because a large number of microorganisms can collectively search through potential solution spaces. With the use of localized interactions (e.g., through diffusion/adsorption of small molecules) and distributed computing across large numbers of microorganisms, microorganisms can utilize this inherent parallelism to approximate solutions to complex optimization problems, such as the Hamiltonian path problem and traveling salesman problem (3).As an additional demonstration of non-computational processing, slime molds have been used experimentally to model emergent networks of minimal-distance travel, to solve mazes and to simulate real-world transportation networks. However, there are limits to the computational potential of microorganisms: compared to electrical gates, their processing times are generally quite slow; output can fluctuate due to metabolic changes or noise; and the complexity of developing large-scale, sophisticated algorithms for microbial systems remains a challenge. In spite of their great ability for parallel exploration and adaptability, bacteria are not as accurate, reliable, or timely as electronic systems when operating in situations other than specialized applications. Nevertheless, their low energy and massive scaling capabilities make them attractive candidates for applications in medical diagnostics, low-power distributed decision-making, environmental sensing, and materials exhibiting intrinsic responsive and adaptive abilities that are not present with conventional methods of computation.

Optimization Challenges Solved by Microorganisms

Microorganisms have shown exceptional capability for solving numerous optimization problems, including routing issues (networks) and the Traveling Salesman Problem (TSP). As recently as 2025, researchers demonstrated optical feedback using a modified Hopfield Network to direct plasmodium to travel through micro-fabricated channels that represented 8 towns. Of all the various microbial organisms studied, Physarum polycephalum (a slime mold) has served as the primary living simulator (or approximator) for TSP. When Physarum achieved a non-equilibrium steady state, it retracted/expanded several branches of its network until only one branch remained in each of the solution lanes, closely following what is considered an approximate optimal tour. In some small cases, this method has performed better than conventional quadratic time heuristics and has shown a nearly linear scaling of results with respect to N for values of N up to 8.

To develop solutions for more complex combinatorial problems, TSP solution algorithms based on slime mold behavior have been combined with ant colony or genetic algorithms, as well as with biological solutions to TSP. To improve the accuracy of path selection and speed the convergence of solutions, hybrid approaches combine slime mold flow/distance techniques with pheromone dynamics from ant systems [4].In addition, there are slime mold models used to approximate something similar to network routing and the Steiner tree formulation, which includes telecom and sensor networks, to develop effective spanning/routing topologies by using the emergent protoplasmic networks formed by the slime mold. Artificial gene circuits added to the microbial cell make it possible to count, add and subtract in a mathematical sense within microorganisms. The first intracellular counters were developed using DNA-invertase and rib regulated transcriptional cascades that functioned together in early synthetic biology research at MIT and Boston University's early colonization of the field of synthetic biology. In essence, these counters represent how a cell could be trained to behave like a biological clock or like a signal processor by producing protein or fluorescence outputs in response to a predetermined number of "stimuli."

While the circuit does not yet enable the complete performance of addition and subtraction in any one cell, these circuits can serve as the basis of engineered biological systems for timing, memory, and logic operations. A second example of microbial computation is combinatorial problem-solving, particularly in terms of maze-solving and decision-making. In maze navigation research, the slime mold plasmodium can be guided to an objective by placing a chemo-attractant at that objective. When a slime mold plasmodium is injected near a maze's edge, it will move toward the objective along one of several available maze paths using chemical gradients. Eventually, it will retract all but the path that leads directly to the food, which demonstrates that this slime mold has executed a one-shot shortest-path algorithm.

In a 2021 study, six distinct bacterial populations that have been engineered with different genetic logic gates were able to "read" four different chemical inputs that represented all possible combinations of a 2x2 maze (i.e., 2x2=4 possible configurations). Again, engineered bacteria have demonstrated how they can solve chemical mazes by only producing fluorescence of specific populations of cells based on the chemical inputs, indicating whether or not there was a valid solution path. This study has shown that bacteria can use combinatorial truth-table logic circuits to assess the solvability of maze instances and provide an observable output in the presence of complex mixtures.

Moreover, microbes utilize their inherent sensory circuits to make decisions about their environment and select between areas that are hazardous versus areas that are nutrient-rich. Physarum, for example, has shown the capacity to "anticipate" reoccurrences of negative environments and modify its branching or movement pattern accordingly, which provides a sense of learning or prediction where there are safety and exploration trade-offs. In computational systems with many microbial organisms involved, almost every member within each colony or group will independently be able to search for multiple options (pathways or configurations of state) at once, based on relatively uncomplicated local rules (e.g., chemical attraction, dynamic branching in growth, genetic regulation, and quorum sensing). They are all applying decentralized reasoning, low-cost energy to produce large emergent solutions to combinatorial and graph problems. Unfortunately, microbes are slow to operate (hours or days), have limited ability to become more complex, and suffer from biological noise and variation that make precision in repeating the same results difficult to achieve. On the other hand, microbiological computing has many valuable features that distinguish them from standard computational systems, including the ability to self-replicate, react to stimuli, explore options in parallel, and adapt to changes. They represent a model of living computation that is embodied, morphological, and realized through physical reactions to events in real time, which is a key distinction from digital electronic computational systems.

Microbial computation:

Chemical Gradients

Chemical gradients are an essential feature of how bacteria perceive, process, and respond to their environment. Essentially, chemical gradients provide bacteria with a spatial input for their computation. An example is the use of methyl-accepting chemotaxis proteins (MCPs), which are transmembrane receptors used by bacteria like Escherichia coli to detect changes in the abundance of attractants (e.g., sugars and amino acids) and/or repellents in the environment through chemoreception. MCPs allow for the modulation of flagellum rotation by binding to chemical signals that then initiate a series of phosphorylation in the cytoplasm via a signaling cascade involving CheA and CheY. As a result of the biased random walk behavior of E. coli, they can "move away" from larger concentrations of toxic/beneficial chemicals. Critically, E. coli also have memory to be able to determine their environment relative to their previous experiences, which is accomplished because E. coli detects temporal changes in concentration rather than absolute concentrations. In a computational sense, the motility of the cell is the output (i.e., the choice/ decision made by the cell), the chemical gradient is the input variable, and the receptors/signaling cascade act as the analog processor. Bacteria use chemical gradients to convert environmental signals into gene expression responses, spatial patterns, and directed movement. This ability to sense and respond to chemical gradients provides a foundation for more complex calculations that enable optimization, spatial decision-making, and emergent network formation within multicellular microbial systems.

Nutrient placement and growth patterns

The relationship between microbial growth patterns and the allocation of nutrients constitutes an important aspect of mathematical problem-solving, particularly in disciplines such as network architecture, graph theory, and spatial optimization. For example, when developing an efficient network of protoplasmic tubes and migrating toward nutrient sources, the slime mold Physarum polycephalum is essentially solving traditional computation problems such as finding a minimum spanning tree, the shortest route between multiple points, or even a simplified version of the Traveling Salesman Problem. The common goal of these classic computation problems is to find the most economical way to connect multiple locations while minimizing the distance to do so. The slime mold appears to generate almost perfect solutions solely based on its internal feedback mechanisms and external chemical signals without using a digital computer or any other type of symbolic reasoning. By modifying their physical form, organisms are able to encode a solution to an "inherent problem" in the spatial layout of the nutrients around them through some form of "morphological computing". This has been shown to have parallels with "heuristic search type algorithms" where bacteria grown in structured environments with non-uniform distribution of nutrients tend to exhibit "branching" or "fractal" growth patterns, which are representative of their adaptive methods of exploring an area [16]. These types of solutions based on growth provide biological equivalents to mathematical methods that are important for determining optimal routes and the efficient allocation of resources in case of urban planning, logistics, and operations research. Therefore, microbes turn environmental information (i.e., where nutrients are located) into physical embodiments of "mathematical solutions", which demonstrates how nature can perform "computations" in the "real world" through dispersed and embodied methods.

Logic Gates through Genetic Circuits

Bacterial genetic circuits behave like logic gates that allow for the processing of information and the making of decisions based on specific exposures to chemicals or environmental stimuli. These circuits are constructed using DNA sequences encoding regulatory elements (riboswitches, repressors, promoters, etc.) that regulate the expression of genes due to changes in the environment (internal or external). For example, an AND gate in E. coli could be constructed that produces a fluorescent protein only when both inducers (chemical signals) are present, which will only show a "true" output when both inputs are active [9]. Bacteria can therefore perform classical computing operations through these artificial genetic constructs that are based on Boolean logic operations. Furthermore, bacteria have the ability to perform more complex functions such as multiplexing, counting, storing memory, and simple arithmetic by combining multiple gates into single living cells. These logical reasoning capabilities enable bacteria to perform mathematical problems such as pattern classification, signal integration, and conditional response behaviors, similar to how a simple algorithm reacts based on rules. As an example, from a biological perspective, this means that a bacterium would activate a survival gene in the presence of both a stressor and a quorum signal to determine its optimum response to its environment.

Stigmergy

The way bacteria use their environment to help one another by leaving traces or markings that other bacteria will use to find and follow leads. This method known as "Stigmergy" is an important behavior for bacteria to solve problems cooperatively. When bacteria use stigmergy to solve cooperative problems they do so without centralized control; rather, they use a system of collaboration to help them complete complex computer-like calculations without needing a central processor. The use of distributed iterated processes to solve mathematic problems through a specific process is similar to the method of solving graph and optimization problems by bacteria through stigmergy.

A clear example of how bacteria use their environment to collate and share their collaborative efforts is demonstrated in the process of EPS production in Pseudomonas aeruginosa. An EPS-producing character leaves behind a trace of this outside their respective cells which can subsequently be detected and followed by later arriving unassociated cells. The combination of the marked path leads subsequently arriving cells to a particular area of EPS production, or other resources, in an organized and efficient manner, thereby leaving an environmental memory for self-organized path creation that guides subsequent movement as well as structural computations to be made.

This process of using an environment to self-organize a path and create an organized structure is directly comparable to the method in which one uses an A* search algorithm or pathfinding techniques by marking visited points to inform subsequent points whether to visit again. Slime molds, such as Physarum polycephalum, leave a trail of protoplasm behind them as they navigate the environment. The mold is able to travel the shortest routes and build a network of paths that resembles (in both their properties and in how they build these networks) the concepts of minimal spanning trees and Steiner trees from graph theory by tracking where it has been previously and utilizing that information to influence how it will travel in the future.

From a computational point of view, geometrical environments can be transformed into distributed data structures using stimuli. The surroundings represent and employ variables and memory in the same manner as formal algorithms do, allowing them to collect and modify information regarding the state of the system. Through the use of stimuli, an organism builds upon previous successes and levels of success to arrive at a solution to a combinatorial optimization problem, such as a maze or routing problem. This is analogous to techniques used in ant colony optimization (ACO) to solve problems using heuristics in operations research and artificial intelligence. Stigmergy can serve as a biological model of decentralized computation with memory enhancement, as well as an inspiration for swarm robotics and distributed algorithms. Furthermore, microbial systems' use of their environment to engage in algorithm-like behaviors in both a physical, embodied form and through modifications to their surroundings provides insight into how life computes without a brain or centralized processor, thus lending itself to mathematical implications.

Integration Model (Microbial Computational Unit)

The information encoding, signal processing, physical adaptation, and feedback coordination functions of microbial computational units consist of a dynamically integrated computational loop that is created as a distributed, self-organizing (rather than centrally-controlled) living system. The integrated model is based on the morphological computation paradigm that uses the internal cytoskeletal structure, network topology, and shape of the organism as a substrate for storing and processing information. Behaviors of slime molds have been mapped into a graph-based model of computation using the Kolmogorov-Uspensky machine paradigm. In this graph, the protein source nodes represent the edges, and the protoplasmic tubes are connected to the edges. Additionally, the plasmodium rewrites the edges dynamically according to the local gradient and flow rules, producing graph transformations and representing the morphological form of algorithmic logic. The first component of the integrated model is the external data encoding. For example, when nutrients, repellents, or chemical signals are available in the environment, a microbial cellular or plasmodial spatial arrangement creates gradient fields that define the location of protein nodes and influence how and where they can place or engage with other organisms (e.g., grow towards or express a gene) (7).Through receptors and signaling pathways, internalization of these domains into gene regulatory states and patterns of cytoskeletal networks form information streams that extend throughout the body. The cytoskeletal networks of slime molds are living hardware that supports data processing and dissemination of signals across the slime mold. The mechanism for computing operations is the propagation of signals and the reconfiguration of networks. Morphological adaptation and metabolic perception circuits are the two primary modes of signal processing in this embedded substrate. In bacterial systems, metabolic flux is influenced by environmental input (nutrients or inducers) levels; then, gene expression and activity of transcription factors occur via a series of interrelated feedback loops within the network. As with constructed genetic circuits, logic operations and state transitions (AND/AND gates, toggle states, counters) are performed by these processes. Like minimum routing or spanning tree estimations, slime mold networks reduce transport distance and energy costs by retracting branches that are the least utilized, while also using flow-based feedback to drive protoplasmic tubes that carry high flow rates towards nutrient sources.

Feedback loops play a critical role in coordinating and adapting to memory needs, coordination, and memory. In flow dynamics, the negative nature of feedback allows for exploratory movement and eliminates the possibility of premature fixation, while the positive nature of feedback strengthens pathways towards nutrient-dense areas. Bacteria coordinate their gene activity at the colony level via a process called quorum sensing, which is based upon the interpretation of local chemical signals and allows bacterial colonies to make collective decisions. A part of the process involves using materials from slime mold aggregation - such as depositing protoplasm on the surface and creating extracellular trails - so that past states (for instance, previous growth patterns) will influence what occurred next (in other words, recording intermediate computation states) through a means known as stigmergy.

Thus, the biochemical computation system for microbiological unit will develop and operate, according to the following phases:

1. Aggregating nutrients and utilizing outside spatial gradients for recording information.

2. An integrating/recording process for local sensory information being processed by receptor systems/cytoskeletal systems.

3. A distributed processing system utilizing adaptable networks, large gene circuits, and metabolic feedback loops.

4. A morphological output from the system in relation to connectivity to the unit, form of the unit, and states of gene expression.

5. Interaction with the outside environment has modified the field to enable further calculations.

The above process is like a biological model of digital computation where the loop has: Low Energy Metabolic Basis; Parallel Physical Execution; a High Tolerance for Errors due to Noise and Uncertainty. Mathematically, this model is conceptually related to agent-based simulation models, reaction-diffusion simulation models, and graph rewriting methods. The development of formal models for slime mold computing has taken place using multi-agent systems which model emergent pattern generation based on local rules and diffusible inputs and also through using differential equations to model cytoplasmic flow and network adaptability. These models correspond closely to optimization algorithms produced by humans but exist in biological material instead; this is evidenced by their ability to model Voronoi or Steiner graphs, to find the shortest path between two points, and to adapt to changing inputs without any central authority or control over the operation of the system.

Microfluidic platform technology provides an infrastructure that enables the integration of design cellular populations containing different logic modules for developing integrated microbial computational units in engineered bacterial circuits. Bacterial gene circuits use the chemical inducers diffusing into channels as inputs to process signaling; therefore, the sensors which read the outputs (e.g., pH changes or fluorescent signals) can be utilized to provide feedback rounds that allow for stable logic processing, even when there may be molecular noise or signal delays. Overall, the microbial computational unit functions as a living, growing machine: its growth and gene expression represent solutions to computational problems; the cyto-skeleton and metabolic networks serve as information processing systems; environmental memories store prior states of the system; and nutrients and chemical gradients encode problems within the system. This unit illustrates an example of natural algorithms, where computational processes are embedded within biological structures, movement and processes. Through these natural processes, bacteria can solve logical, spatial and optimization challenges using robust and embodied rules of action that challenge our notions of computation.

Synthetic biology and engineering solutions

Synthetic biology has begun to create living cells that behave as computable entities that can be programmed to carry out specific tasks and actions. The original programming of living cells occurred when scientists first created the genetic "toggle switch" in Escherichia coli (E. coli) through the work of Collins, Elowitz and their contemporaries, who developed oscillators and bistable genetic "toggles" within that same organism. These developments have created possibilities for scientists to program cells to function similarly to digital bits and clocks based on the same principles of computation. It is now possible for researchers to construct a variety of types of circuits from allosteric transcription factors (aTFs), orthogonal promoter systems and CRISPR-dCas9 based logic gates, thus enabling modular design of gene circuits. Each of these designs has the potential to create layered computations through the interconnection of complex Boolean logic AND, OR, NOT and will ultimately lead to the development of sophisticated gene circuits embedded within living cells. Microbial computing has also been scaled down to operate more effectively on small devices through the use of microfluidics. An example of scaling down lies within the biochip, which uses microcolonies, or groups of microorganisms, in small channels to operate as logic modules. A specific biochip design that illustrates this type of circuitry is one that contains altered E. coli AND gates; when integrated electrochemical sensors detect the presence of certain chemicals, the AND gates will alter the pH or dissolved oxygen levels of their surroundings. Both of these biochips represent programmed microbial circuitry on a chip and have been modeled with respect to the time it takes for signals to fluctuate and for molecules to delay when executed by the biochip's processing system. "Living bio-sensing platforms" bacteria engineered to contain promoter-reporter circuits, or to use these circuits to identify pollutants such as: heavy metals, TNT, or residues from landmines are created using similar concepts. Whenever the sensor molecule attaches to its target, luminescence or fluorescence is produced in a cascading manner, indicating the presence of contaminating substances. Therefore, because these systems are designed circuits for converting information received from the environment into a value that can be read from a unit (output), they calculate the "logistic" process of the target based upon what was sensed. While DNA computing and molecular logic networks can exist completely independently of living cell systems, they utilize nanostructures for implementing AND, OR, NOT, and even multi-bit computations along with DNA strand displacement, DNA enzymes, and switches based on DNA aptamers [10]. Scientists have developed non-enzymatic molecular circuits using toehold-mediated strand displacement that can play Tic Tac Toe, compute square roots and cube roots, and reset automatically [11]. These molecular circuits rely on chemical processes to represent computation; the input to the system produces hybridization events, while the system's output is manifested via the production of a catalytic product or fluorescence. As evidenced by this work, DNA molecules can serve both as hardware and software for computers. The rise of theragnostic applications is illustrating the potential for biomarker identifying DNA logic circuits to amplify diagnostic signals or release therapeutics using logic gates for high-fidelity-based cancer treatment with control and specificity. Finally, as we move towards development of reprogrammable molecular systems, it will become possible to use one set of microbial chassis or tile designs to host multiple algorithms. As such, the change towards dynamic, re-usable molecular programming was exemplified by one DNA tile based system that displayed 21 algorithms, including leader election, counting and pattern formation, derived from a common hardware base that differed only by seed strands or tile selection [14].

Synthetic biology combines:

- Genetic logic gates using CRISPR in genetically modified cells for programmable response and sensing.

- Microfluidic (or biochip) systems that allow biological logic with electrical output to be integrated.

- DNA computing devices that execute chemical logic and do not require living cells.

Collectively, these elements create a layered toolkit of biological computing architecture, illustrated by synthetic constructs, DNA and living organisms functioning as either living computers or molecular computers; both of which can solve mathematical puzzles, perform diagnostic tests and enable biosensing in environmental and medical applications [15].

Limitations and challenges

The field of Microbiological and Molecular Computing is fast-expanding; however, the ability to apply these technologies to large scale and practical applications is faced with many challenges. One major issue is Scalability: while organisms and DNA-based computers demonstrate good capabilities for simple tasks (e.g., logic operations), these biological systems find it more challenging to scale when tasks become more complex. Microorganisms aren't able to multiply in a similar manner to digital computers that can easily add additional processing units. Increasing complexity in a biological system causes difficulties maintaining the accuracy of larger systems as they may fail to operate properly due to circuit failures or gene mis regulations.

Another key challenge faced by biological computing is Speed. Biological computing occurs on a slower timescale compared to electronic computing. For instance, a silicon chip can perform computations at the microsecond level of granularity whereas a DNA-based computing chip may require hours to complete. The delay from the process of Gene Expression and Molecular Interactions explains this slow processing compared to electronic computing. Thus, biological computation does not lend itself to applications that require either time-sensitive results (i.e., robotics) or real time results (i.e., environmental sensing).The control and predictability, caused by a multitude of variables including changes in temperature and nutrition will determine the success of an organism. As a result, it is difficult to ensure a consistent end product because of the continuous change of inputs and outputs, including within a complex network and in a dynamic environment. Even if the system has been well designed, it will still experience unexpected results due to many internal/external forces. Unpredictability will increase the complexity of performing reliable tasks such as secure encryption and accurate calculations.

Researchers are working on new solutions, such as genetic components designed to tolerate noise and hybrid systems that combine electronic and biological components. Digital systems will most likely not be replaced by biological computing; however, it may perform better in areas where biocompatibility, flexibility, and adaptability are of greater importance than speed. Microbial systems may also be well suited to applications where computation involves direct interaction with the environment, such as environmental remediation, biosensors, and diagnostics. However, in order for the microbial industry to have a broader market, it must overcome its current scalability, performance, and control limitations.

Future Directions

The combination of microbial and biological computing with traditional electronic devices is creating bio-hybrid computing systems that merge the speed and efficiency of silicon-based computers with the flexibility and energy efficiency of living organisms. Within these systems, biological components such as genetic circuits, DNA strands, or engineered microorganisms can be used as flexible, responsive connections to sense, learn, and respond to stimuli in ways that standard computer systems cannot do [13]. Current research is exploring these hybrid systems in applications such as microfluidic devices that use digital encoding to operate bacterial logic gates or biosensors that detect chemical stimuli and generate electrical signals. One promising area of research is the development of biological computing as a solution for solving real-world optimization problems in the areas of robotics, medicine, and logistics. For example, researchers are conducting research on microbial-inspired systems for routing supplies, navigating autonomous robots, and optimizing traffic networks; previously these applications were developed using only heuristic or evolutionary algorithms. Synthetic gene circuits provide a future way to make smart therapeutics cells that will recognize sickness pathways and determine when to dispense a medication. The formation of engineered tissues and microbial biofilms that will consume surrounding stimuli and execute logical functions is an area of research that has not yet been achieved. In this sense, these engineered biological materials will emulate certain aspects of thought, decision-making and self-repair through preprogrammed stimulation. Such applications of engineered biological materials can be utilized in soft robotics, bio-adaptive interfaces or responsive infrastructure. Continuing to foster connections through the multidisciplinary efforts to unite biology, computing, and engineering will allow for engineered biological computers to evolve from a pursuit of imagination and into an established technology for the delivery of intelligent, responsive and sustainable solutions.

Conclusion:

Microbial systems are providing a very exciting and revolutionary challenge to traditional definitions and classification systems of intelligence, processing, and algorithmic design, and these new ways of looking at the world are completely changing the way researchers view computing in general. For example, unlike traditional digital computing, which typically relies on binary representations of information and operates through electronic circuit design, microbial computing utilizes the inherent abilities of living organisms, such as chemotaxis, genetic regulation, growth patterns, and environmental adaptation to perform logic gate computations that are often performed by artificial intelligence or mathematical reasoning. There are numerous examples of how biological systems function as computing devices; e.g., engineered bacteria can perform logic gate functions (using logic gates as inputs for logic circuitry) and slime molds are able to map out the most efficient paths. These organisms are examples of how computing ability is not necessarily restricted to silicon architecture; rather, there are different types of computational capabilities that originate from the physical and chemical processes of organisms. Furthermore, it is important to note that microbial computing has several critical advantages compared to traditional computing, including the self-replicating, energy-efficient, and highly adaptable nature of microbes that allow them to perform multiple types of tasks simultaneously and continue to operate under dynamic, unstructured environments (qualities that are becoming increasingly important in addressing real-world problems, especially in the areas of environmental monitoring, logistics, and healthcare).Additionally, researchers can alter the behavior of microbes and construct biological computers that can solve specific problems by incorporating synthetic biology techniques (such as CRISPR) with programmable genetic circuits. The exploration and application of microbial computation will not only extend the boundaries of what is possible in computing, but also inspire new forms of distributed, embodied, and emergent intelligence [18]. Therefore, this integration of biology and computation may lead to the development of computer systems that are more resilient and sustainable, and that can operate in a specific context, and potentially replace or augment traditional technologies in the future.

References

[1] - Bagh, S., Sarkar, K., Bonnerjee, D., & Bagh, S. (2024). Genetically engineered bacteria can do complex math: prime-number identification, vowel recognition, pizza-cut optimization. Nature News, 24 October 2024.

[2] - Sarkar, K., Bonnerjee, D., & Bagh, S. (2021). Engineered bacteria computationally solve chemically generated 2×2 maze problems. arXiv preprint. June 2021.

[3] - Baumgardner, J., Acker, K., Adefuye, O., Crowley, S. T., DeLoache, W., Dickson, J. O., … Eckdahl, T. T. (2009). Solving a Hamiltonian path problem with a bacterial computer. Journal of Biological Engineering, in press [2009].

[4] - Poet, J. L., Amos, M., Heyer, L., & Eckdahl, T. T. (2010). Bacterial computing: genetic algorithms and bio-inspired techniques using living cells. Bacterial Computing, in Natural Computing chapter. (See review).

[5] - Bonifaci, V., Mehlhorn, K., & Varma, G. (2011). Physarum can compute shortest paths. arXiv preprint. June 2011.

[6] - Painter, K. J. (2018). Mathematical models for chemotaxis and their applications in self-organisation phenomena. arXiv preprint, 2018.

[7] - Dubinkina, V., Fridman, Y., Pandey, P. P., & Maslov, S. (2019). Multi stability and regime shifts in microbial communities explained by competition for essential nutrients. arXiv preprint, June 2019.

[8] - Cordero, O. X., & Poet, J. L. (2010). From bacteria to biocomputers: programming cells to solve math problems. In Bacterial computing chapter in Natural Computing.

[9] - Bui, H., Shah, S., Mokhtar, R., Song, T., Garg, S., & Seelig, G. (2018). Localized DNA hybridization chain reactions on DNA origami. ACS Nano.

[10] - Chatterjee, G., Dalchau, N., Muscat, R. A.,

Phillips, A., & Seelig, G. (2017). A spatially localized architecture for fast and modular DNA computing. Nature Nanotechnology.

[11] - Qian, L., & Winfree, E. (2011). Scaling up digital circuit computation with DNA strand displacement cascades. Science, June 2011.

[12] - Winfree, E. (2010). Biological computation: emergent information processing in active matter. Biological Computation, Computer Science Faculty Publications.

[13] - Mitchell, M. (2010). Biological computation. Faculty Publications.

[14] - Ben-Jacob, E., Cohen, I., & Levine, H. (1998). Cooperative strategies in bacterial colony pattern formation. Fractals, 3, 1859–1862.

[15] - Ben-Jacob, E., Cohen, I., & Levine, H. (2000). Cooperative self-organization of microorganisms under stress. Physica A, 282, 247–282.

[16] - Ben-Jacob, E., Shochet, O., Sochet, O., Avidan, O., & Tenenbaum, A. (1994). Holotransformation of bacterial colonies and genome cybernetics. Physica A, 202.

[17] - Bonifaci, V., Mehlhorn, K., & Varma, G. (2011). Natural algorithms in slime mold solving shortest-path. Theoretical Computer Science.

[18] - Rosenberg, G., Back, T., & Kok, J. (2012). Handbook of Natural Computing. Springer. Chapters include microbial computation.

[19] - Royal Society Interface modelling group. (2017). Mathematical modelling of microbes: metabolism, gene expression and system dynamics. Journal of the Royal Society Interface.

[20] - Cordero, O. X., Poet, J. L., Di Paola, M., & Amos, M. (2004). Learning and evolution in bacterial intelligence and computing. Natural Computing chapter.