Energy & Environmental Science

Cite this: *Energy Environ. Sci.*, 2011, **4**, 4907

www.rsc.org/ees

PERSPECTIVE

Bacteria-based biocomputing with Cellular Computing Circuits to sense, decide, signal, and act

Michaela A. TerAvest, Zhongjian Li and Largus T. Angenent*

Received 23rd August 2011, Accepted 11th October 2011
DOI: 10.1039/c1ee02455h

Although the term 'Biocomputing' may bring to mind biological replacements of silicon processors; this type of application is far in the future. Use of bacteria-based Biocomputing for biosensors and industrial fermentation control, however, is presently attainable by using genetically-engineered bacterial cells that can process signals in a logical operation *via* one or a few pathways. Here, we refer to these systems as 'Cellular Computing Circuits' and focus on their possible future implementations. We also briefly discuss concepts from Synthetic Biology and enzyme-based Biocomputing because they will be important during future development. Our lab has already transformed an idea from enzyme-based Biocomputing into a bacteria-based Boolean logic gate with a digital output signal of direct electric current and we suggest future applications in this perspective. We predict useful functions for Cellular Computing Circuits in the near future.

Introduction

The term 'Biocomputing' may suggest a desktop computer with whole and live bacterial cells as the processors in lieu of silicon chips. And, indeed, a prelude of this concept has been explored within the area of 'Amorphous Cellular Computing' (shown as the red-lined interface within the yellow area of 'Cellular Computing' [Fig. 1]) to encompass Biocomputing, 'Synthetic Biology', and 'Amorphous Computing' – Fig. 1 visualizes the relationships among the 'Natural Computing' terminologies and gives their definitions. For instance, Baumgardner *et al.*[1] used genetically-modified *Escherichia coli* cells as massively parallel processors to solve the travelling-salesman problem (a Hamiltonian path solver [Table 1]), which is a problem that requires long processing times because of the serial nature (*i.e.*, check one possible solution at a time) of silicon-based computing. The authors designed and constructed a genetically-engineered bacterial strain that contained a carefully mixed-up DNA sequence for producing fluorescent protein, which could only become functional if rearranged in a known and specific order.

Upon growing this strain, the authors, indeed, found colonies that had successfully produced fluorescence as the output signal by recombination during growth; and sequencing of the rearranged sequence provided the solution to the problem. This proof-of-concept study worked with a known sequence and still required accurate modeling of the sequence information and sophisticated molecular biology skills. Although this approach points to the tremendous potential of using recombination in solving complex problems, a considerable hurdle must be overcome in terms of how to solve problems when the final sequence is unknown. In addition, computational problems that cannot be solved with silicon computing are rare and with the exponential increase in processing speeds it is unlikely that biocomputing will ever be competitive in traditional computational problems. We believe that for these reasons using genetically-engineered bacterial strains to solve real, sophisticated computational

Department of Biological and Environmental Engineering, Cornell University, Ithaca, NY, 14853. E-mail: la249@cornell.edu

Broader context

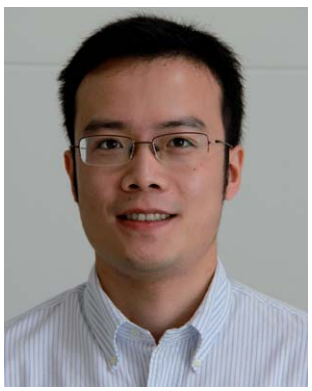
Traditional computing has clearly been essential in the development of current science and technology. However, there are important limitations to silicon computing, such as limited parallel processing and inability to interface directly with biological materials. As a result there is great interest in biocomputing, which may be able to address these issues. Bacterial cells can be conceptualized as individual, massively parallel processors with the ability to directly measure biological and chemical species. This field is in a very early stage of development, and a biocomputer with functionality similar to a silicon computer is not yet attainable. Nonetheless, the field is advancing quickly, and several useful applications of simple biological circuits are on the horizon. These biological circuits will be particularly useful for complex biosensing with customizable output signals.

problems by massively parallel processing (*i.e.*, *via* Amorphous Computing[2]) is far from being practical in computing systems.

There are, however, other functions, such as biosensing and controlling simple bacterial activity, in which Biocomputing may be advantageous because of the ability to process biological signals in a logical operation. Exclusion of the need for Amorphous Computing, creates a narrower research area that combines the Biocomputing and Synthetic Biology fields to focus on whole and live bacteria that process along one or several pathways rather than processing along many pathways in parallel. Here, we refer to this area as 'Cellular Computing Circuits" (shown as the green-lined interface within the yellow area of Cellular Computing in Fig. 1). In this perspective, we will focus mainly on this concept but incorporate ideas that stem from 'Biomolecular Computing' with, for example, enzymes (*i.e.*, 'Enzyme Computing'), and thus we will discuss previous research activities from within the entire purple circle (Fig. 1). However, this is not meant to be a comprehensive review of the Biocomputing area, and does not, for example, focus on Eukaryotic cells. For information on Cellular Computing with Eukaryotic cells we refer to four recent reviews, which also discuss fluorescent-protein-based and bioluminescence techniques, such as fluorescence resonance energy transfer (FRET).[3–6]

Michaela A. TerAvest

Michaela TerAvest is a Ph.D. candidate in Biological and Environmental Engineering at Cornell University in the research group of Dr. Lars Angenent. Her thesis topic focuses on understanding metabolic limitations of Shewanella oneidensis *MR-1 in bioelectrochemical systems by bringing biochemical techniques into the engineering realm. However, she also works on projects in other areas of bioelectrochemical systems, including bioremediation and biocomputing. She is expected to graduate in 2013 and received a B.S. in Biochemistry and Molecular Biology from Michigan State University in 2008.*



Zhongjian Li

Zhongjian Li is a Ph.D. student at Zhejiang University. He worked in Largus Angenent's lab at Cornell University as a visiting fellow during 2009–2011. His research area during the visit focused on the development of bacteria-based logic gates with Pseudomonas aeruginosa *and building a microfluidic bioelectrochemical system that can be used as a biosensor.*

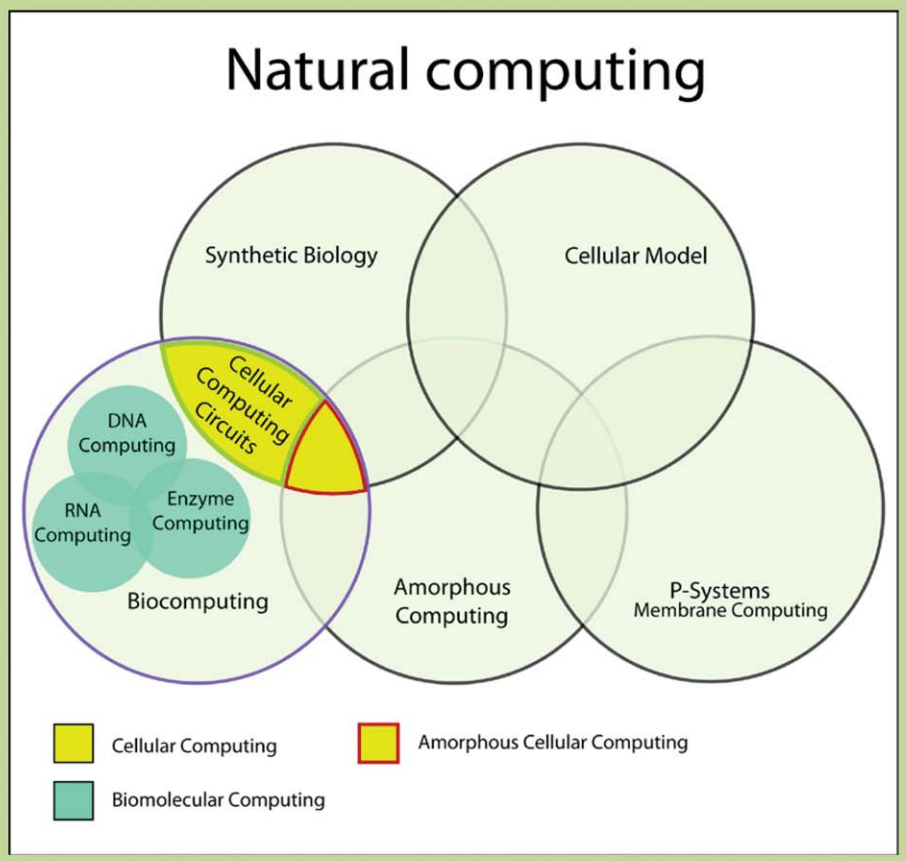
Genetic engineering has been used to generate Biocomputing programs for Cellular Computing Circuits with whole and live bacterial cells. These programs include counting, oscillation, and edge detection (*i.e.*, take a picture with bacteria) (Table 1). In addition, several studies have now developed versatile Biocomputing programs by using Boolean logic operations. In fact, Boolean logic operations are key to controlling complex biochemical reactions and to implement digital logic (11 Boolean logic gates are explained in Table 2). Logic operations are innate to bacterial cells, but often not useful to us or embedded in biochemical systems that are too complex to understand. Therefore, all published studies with Cellular Computing have used genetic engineering to simplify the Boolean logic gates with chemical signals as inputs and one detectable signal as the output.

In this perspective it becomes apparent that, except for our work, almost all other Cellular Computing studies have engineered fluorescent protein in *E. coli* to produce fluorescence as the output signal. Even though this is an advantage with regard to signal specificity and amplification, it requires additional fluorescence measurement technology, such as external excitation light and fluorometers, to communicate the information to the user interface. In addition, the fluorescence signal is prone to degradation while background autofluorescence reduces sensitivity.[6] Therefore, to circumvent problems with fluorescence, our work by Li *et al.*[7] utilized a mutant strain of *Pseudomonas aeruginosa* to generate a digital output signal of direct electric current by using the ability of wild-type *P. aeruginosa* to respire with a solid electrode by exocellular electron transfer (EET).[8]

Many studies have now shown that whole bacterial cells can generate a logical decision through Boolean logic operations despite the complex nature of the cellular mechanisms (Table 2). Biological components, such as enzymes, DNA, and RNA, are less complex and more specific compared to whole cells and have, therefore, been successfully used to generate an array of relatively complex cascades of logic gates (*i.e.*, Biomolecular

Largus T. Angenent

Lars Angenent is an Associate Professor of Biological and Environmental Engineering at Cornell University. His main research focus is in bioenergy, particularly conversion of organic wastes to energy carriers (such as electric current, medium-chain carboxylates, or butanol). He is also interested in unconventional uses of bioelectrochemical systems, such as biosensing and biocomputing. Other research areas include anaerobic digestion and syngas fermentation, with a particular focus on using a combination of engineering and molecular biology approaches to improve these systems. He received a Ph.D. in Environmental Engineering from Iowa State University and M.S. and B.S. degrees in Environmental Technology and Environmental Science from Wageningen University.



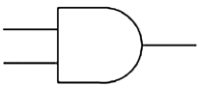
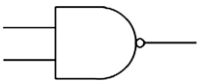
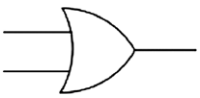
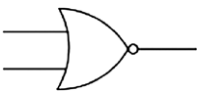
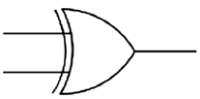
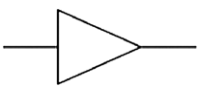
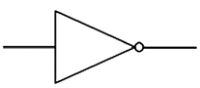
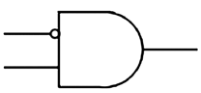
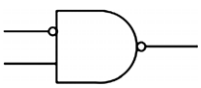
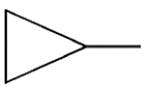
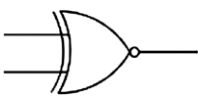
Amorphous Computing: Computation using many small, low power processors in parallel to perform complex functions via many pathways.
Amorphous Cellular Computing: Subset of Cellular Computing for which genetically-engineered whole cells are used to follow many parallel pathways.
Biocomputing: Computation performed using biological materials.
Biomolecular Computing: Subset of Biocomputing in which specific biological materials are used.
Cellular Computing: Subset of Biocomputing for which computation is performed by genetically-engineered whole cells.
Cellular Computing Circuits: Subset of Cellular Computing for which genetically-engineered bacterial cells process signals in a logical operation via one or a few pathways.
Cellular Model: Complete computational model of the cell, including algorithms, data structures, and visualization.
Membrane Computing: Algorithms and computational models based on cell membranes (part of the Cellular Model).
Natural Computing: Any research area combining computer science and biology.
P-Systems: Biologically-inspired silicon computing.
Synthetic Biology: Using engineering concepts to modify existing gene sequences or design novel sequences to improve engineered bacterial functions.

Fig. 1 Definition of Natural Computing terms and their relationships. For further reading on these subjects see Abelson *et al.*,[2] Purnick *et al.*,[27] Kari and Rozenberg,[45] and Păun.[46]

Table 1 Biocomputing programs

Component	Function	Implementation	Reference
Hamiltonian path solver	Solves implementations of the "traveling salesman" problem	Cell, DNA	1,47
Counter	True after predetermined number of true signals	Cell	19
Oscillator	Oscillates predictably between true and false	Cell	23
Flip-flop memory (latch)	Output is equal to most recent input, until next input	Enzyme	48

Table 2 Biocomputing logic gate operations

Component	Function	Implementation	Reference
AND gate 	True when A and B are true	Cell, Enzyme, DNA	7,20,24,49–54
NAND gate 	True except when A and B are true	Cell, Enzyme, DNA	20,52,55
OR gate 	True when A, B, or both are true	Cell, Enzyme, DNA	20,49,50,54,56,57
NOR gate 	True when neither A nor B is true	Cell, Enzyme, DNA	20,49,55
XOR gate 	True when A or B but not both are true	Cell, Enzyme, DNA	20,49,51
A gate 	True when A is true (filter)	Cell, Enzyme	20,49
NOT A gate 	True when A is not true (inverter)	Cell, Enzyme, DNA	20,49–51
A IMPLY B gate 	True when A is true and B is false	Cell, Enzyme, DNA	20,49,52
A NIMPLY B gate 	True except when A is true and B is false	Cell, DNA	20,58
TRUE gate 	Always true	Cell	20
EQUAL gate 	True when A and B are equal	Cell	20

Computing). Although Biomolecular Computing is possibly further in its development than Cellular Computing Circuits, it is important to realize that both are in their infancy.[9] Boolean logic gates with enzymatic reactions as the core components have been able to process biochemical input signals by defined chemical reactions (Table 2). In addition, by connecting redox-active enzymes to electrodes,[10] the output signal has been digitized (direct electric current). For example, Chuang *et al.*[11] constructed a cascade of enzyme-based reactions to create a **NOR** logic gate with a digital output signal. 'DNA Computing' is another example of Biomolecular Computing and uses simple DNA reaction mechanisms, such as reversible strand displacement. By using specially designed and synthesized DNA strands, sufficient sequence design space is available to perform many Boolean logic operations. For example, Qian and Winfree[12] constructed cascades of **AND**, **OR**, **A IMPLY B**, **NAND**, and **NOR** logic gates with DNA sequences as input signals (and fuel) and fluorescence as an output signal by using sequence-integrated

fluophores and quenchers. DNAzyme computing is a slightly different approach, wherein DNA molecules are used to construct logic gates, and have catalytic action on one another. A recent example by Elbaz *et al.*[13] demonstrated a modular biocomputing system based on a library of DNAzyme inputs and processors. Finally, hybrid systems based on DNA or RNA with enzymes have been developed to perform Biomolecular Computing tasks (*e.g.*, 'RNA Computing').[14,15] Of these Biomolecular Computing examples given here, we will further discuss enzyme computing *via* enzyme-based Boolean logic gates with the goal to incorporate new ideas for improvement of Cellular Computing Circuits with bacteria.

In this perspective we will focus on Cellular Computing Circuits that sense, decide, signal, and act to perform useful functions. Genetically-encoded biosensors[16] with bacteria are an important precursor of Cellular Computing Circuits because they respond to one intracellular input and produce one detectable output. This can be in itself an important function and the genetic circuitry used to accomplish biosensing is similar to the systems required to build cell-based logic gates. Thus far, genetically-encoded biosensors have mainly been used for research purposes, for example, to visualize intracellular concentrations and gradients of compounds in real time.[5] Here, we look beyond genetically-encoded biosensors and discuss the requirement for Boolean logic gates to generate useful applications outside of research. We will also discuss possible immediate implementations for Cellular Computing Circuits.

One important limitation of all Biocomputing platforms is slow computational speed due to diffusion-limited chemical processes.[2] For bacteria-based Biocomputing, the requirement to grow cells or generate functional proteins may further slow the computational speed, and thus Cellular Computing may not be practical for many applications. Some applications, such as biosensing, however, may not require immediate output signals. When more rapid responses are required, workers must utilize the innate ability of bacterial cells to reduce transport-related delays by localizing molecules in confined regions (*i.e.*, membranes or organelles).[17] One example of local DNA Computing with relatively high computing speeds is the use of polymerase reactions with single strands of DNA.[15]

Cellular Computing Circuits

Genetic toolbox

Engineered gene regulation. Development of Cellular Computing Circuits is reliant on techniques to genetically modify bacteria. The most common approach has been the control of transcription through inducible promoters (*i.e.*, promoters that can be turned on and off by a specific chemical or environmental signal). Genetic engineering has been used to develop genetic circuits that process information within the bacterial cell, resulting in a cellular response. Many different types of genetic circuits have been developed as comprehensively reviewed by Voigt.[18] By extracting useful inducible promoters from wild-type bacteria and recombining them in novel ways, researchers can create new regulatory networks and use them to control cell behaviors. Wild-type bacteria contain a wide variety of these promoters, which can respond to environmental signals, such as pH, temperature, sugar type, pollutants, and light.[18] For practical reasons, the resulting Cellular Computing Circuits can be utilized to control many functions in the bacterium, including the production of therapeutics or other biochemicals. However, for scientific reasons, Cellular Computing Circuits have mainly controlled the expression of fluorescent protein for ease of output signal detection and because it does not interfere with normal cellular processes. When different promoters are combined in creative ways to control not only cell functions but also each other, complex Boolean logic gate networks with multiple inputs can be developed.

For example, Friedland *et al.*[19] combined three inducible promoters to engineer an *E. coli* strain that was able to count to three. This strain counts three different inducer pulses (arabinose, tetracycline, and IPTG) and responds by producing fluorescent protein only after a pre-determined number of these pulses in the correct order. This counter was produced with single invertase memory modules that allowed fluorescent protein production only after DNA inversion steps, which were controlled by the inducible promoters. The same study also produced a three-counter system by using only one inducible promoter in combination with the invertase memory modules. Even though this system was able to count three pulses of one inducer (arabinose), the length of the pulses had to be strictly controlled because one long pulse also produced the output signal. The use of invertase memory modules shows that different types of regulation (in addition to inducible promoters) are an important part of the genetic toolbox to engineer bacteria. This counter could be useful for pulse detection in various biological systems. However, there are limitations of this system caused by the stochastic nature of biology and the different responses of individual bacterial cells. This is evident in the disagreement between these counter's true performance and model predictions for some conditions in the same study by Friedland *et al.*[19] To partially overcome this obstacle, it has been useful to exploit bacterial communication (*e.g.*, quorum sensing) to compel individual cells to act in unison.

Quorum sensing. The most recognized form of bacterial communication is quorum sensing and these regulatory systems have already been used in several Cellular Computing Circuits. Quorum sensing can be used in two capacities for Cellular Computing Circuits – synchronization of one population and communication among different populations. Synchronization is important because it allows many cells of one population to act simultaneously, which increases the quality of the output signal. Communication among different populations is important when several Cellular Computing Circuits are linked for more complex functions, analogous to wired connections between silicon circuit components.[20,21] Synchronization and communication occur because quorum-sensing pathways control extracellular levels of bacterial communication chemicals. In gram-negative bacteria the most common molecules to communicate are homoserine-lactones (HSL) of which some are species-specific, while others may be recognized among species of bacteria. A model gram-negative quorum-sensing pathway is the Lux system of *Vibrio fischeri*, which controls luciferase expression. Lux-type pathways consist of a LuxI (autoinducer) homolog and a LuxR (receptor) homolog.[22] LuxI is responsible for production of its own inducer

HSL. This HSL is secreted outside the cell, and above threshold concentrations binds to LuxR. The LuxR-HSL complex then induces expression of genes that are under control of this system, including *luxI*. Because all cells that are near each other will experience the same HSL concentration, they will have synchronous expression of any genes under quorum-sensing control. There is a diversity of Lux-type systems found in over 25 different bacteria.[22] Thus, using different combinations of these systems could not only facilitate synchronization and communication, but also the use of quorum-sensing molecules as input or output signals in bacteria-based Boolean logic gates.

Implementations

Oscillator. The concept of synchronization through quorum sensing was demonstrated by Danino *et al.*,[23] who engineered an *E. coli* strain to act as synchronized oscillators with a fluorescence output signal. The genetic circuits that code for: 1. the Lux pathway; 2. an HSL-degrading protein; and 3. a fluorescent protein were transformed into *E. coli*. The circuit was arranged to produce more HSL and fluorescent protein upon increasing concentrations of the autoinducer HSL. This would not have created an oscillator without the HSL-degrading protein, which is also induced by HSL, but only after a short delay. Because of a delay between HSL accumulation and degradation, periodic bursts of fluorescence occurred due to the cyclic nature of this regulatory operation. Diffusion limitations of HSL caused one group of *E. coli* cells to oscillate as a whole in small enclosures, while the group caused traveling waves of fluorescence in larger enclosures (see the original publication for videos). The frequency and amplitude of the oscillations were manipulated by changing the flow rate in microfluidic devices in which the cells were grown. This oscillation function is not only useful in scientific applications, but could be applicable in biomedical or industrial implementations because the oscillating strain could potentially be adapted for periodic drug or chemical delivery.

AND logic gates. A key component for application of Cellular Computing Circuits is to produce strains that can simultaneously process two or more inputs. This is important because the detection of several biomarkers simultaneously is required to make a conclusive decision (*e.g.*, disease state, security threat, environmental contamination). A basic example of this is the **AND** logic gate, which provides a true output only in the presence of two input signals (Table 2). The construction of **AND** logic gates in bacteria has been accomplished by, for example, Ramalingam *et al.*,[24] who constructed plasmid-encoded **AND** logic gates in *E. coli* with IPTG and tetracycline as the inputs and fluorescent protein as the output. This study is particularly interesting because the authors used six different combinations of inducible promoters to determine how design of the genetic circuit influences the performance of the logic operation. None of the logic gates were 100% accurate, but some logic gates performed better than others, indicating that the specific combination and order of inducible promoters was important. The work, therefore, suggests that modifying small details of the genetic circuit will be an important factor to optimizing Cellular Computing Circuits.

Edge detector (combination of three logic gates). While a single **AND** logic gate is useful for detection of two input signals, some desirable Cellular Computing functions require the combination of multiple logic gates into **one** strain. For instance, Tabor *et al.*[21] constructed one *E. coli* strain capable of detecting light/dark edges to produce a black pigment (output signal) at these edges, resulting in a black and white picture that outlines an applied light mask. They used an **AND** logic gate in combination with two **NOT** logic gates. This was accomplished by transforming genetic circuits into *E. coli*, which code for: 1. an inverted light sensor (dark sensor); 2. a modified Lux pathway that only allows expression of downstream genes with HSL; and 3. a black-pigment protein. The transformed *E. coli* can only produce HSL in the dark, while they can only produce black pigment in the light when HSL is present (they cannot produce HSL in the light). This causes the production of black pigment only at light/dark edges where lit and unlit *E. coli* cells are in close proximity, because HSL does not diffuse freely throughout solid medium in which the cells were grown. This approach was extended to engineer an *E. coli* strain that can distinguish between red and green light to make a black and white picture.[25]

Complex logic functions with multiple strains. To perform more complex functions, Tamsir *et al.*[20] engineered different genetic circuits into eight unique *E. coli* strains. Up to four of the eight strains were connected in different combinations *via* quorum-sensing signals. The authors used HSL signals with the Las and Rhl pathways instead of the Lux pathway.[22] The input signals were two different HSL molecules (3-oxododecanoyl homoserine lactone [3-oxo-C12-HSL] and N-butyryl homoserine lactone [C4-HSL]), tetracycline, and arabinose, while the same HSL molecules and fluorescent protein were output signals for the eight strains. By choosing up to four strains from the logic gate *E. coli* library (**A**, **NOT A**, **OR**, and **NOR** logic gates) in different combinations, the authors were able to construct more complex logic operations, such as **XOR**, **EQUAL**, **NAND**, and **A IMPLY B**. This system greatly improved the number of logic functions that were possible with whole cells, but the manual design of the logic gate library required sophisticated knowledge of molecular biology and ample time at the bench.[20]

One possible implementation of using several engineered bacterial strains together is to monitor and manage mixed communities of microbes. This implementation stipulates that these engineered strains would be able to sense, decide, communicate, and act as a group. For example, a set of engineered strains could be introduced to an industrial fermentor to monitor different populations. Brenner *et al.*[26] demonstrated this type of functional communication by designing two *E. coli* strains that could produce fluorescent protein only when cultured together. This Cellular Computing Circuit mimics an **AND** logic gate where the two input signals are the individual bacterial strains and the output signal is fluorescence. Each strain required one out of two HSL molecules (3-oxo-C12-HSL and C4-HSL) to be produced by only the other strain to generate the fluorescent protein. Their study demonstrates that engineered strains can detect members of the bacterial community and communicate with each other as part of a logic operation.

Transition to in-silico rational design

To overcome difficulties of individual strain engineering, the design and implementation of Cellular Computing Circuits can be aided by *in-silico* rational design. Rather than time-consuming trial and error in strain engineering, the worker would test configurations of genetic circuits and their outputs *in silico* prior to *in vitro* testing. This can yield valuable insight into which genetic circuits will perform as expected based on sophisticated mathematical models. Rational redesign is also an important part of this process, whereby parts of the genetic circuit may be mutated to tune binding and kinetic parameters of the individual genetic circuits to improve overall performance.[27] Batt *et al.*[28] have successfully improved the performance of biological devices by characterizing and compiling genetic circuits, which can be combined and tested *in silico*. This technology will allow Biocomputing researchers to assess many different configurations of a Cellular Computing Circuit in a short time. After this analysis, the most promising strains can be created and tested *in vitro*, which will allow faster tailoring of Cellular Computing Circuits towards useful implementations. In addition, the level of complexity of Cellular Computing Circuits may be increased. Such superior complexity is already achieved for enzyme computing, because the selectivity and lower complexity of isolated enzymes has, thus far, not required *in-silico* rational design.

Enzyme computing

Implementations

Biomedical sensor. Various choices of enzymes make enzyme-based logic gates suitable for many potential applications in different areas. One possible application for enzyme-based logic gates is in the biomedical field. Since many pathological conditions (*e.g.*, injuries) can cause abnormalities in concentrations and activities of biochemicals, including enzymes (*i.e.*, biomarkers) in human bodies, enzymes that can create a biochemical reaction with these biomarkers have been used as clinically relevant indictors. However, single biomarker detection cannot differentiate between closely related pathological conditions. The most reliable assessment method is, therefore, simultaneous detection of multiple enzymatic biomarkers. Fortunately, a logic gate can process multiple enzymatic biomarker input signals and generate an output signal as a rapid YES/NO decision depending on the simultaneous pathological or physiological concentration of multiple enzymatic biomarkers. Windmiller *et al.*[29] have reported detection of soft tissue injury with a **NAND** logic gate using creatine kinase and lactate dehydrogenase biomarkers as input signals with NADH as the output signal. Based on a similar principle, enzyme-based logic gates have also been used to detect traumatic brain injury and hemorrhagic shock,[30] and four other injuries.[31] These enzyme-based logic gates would function as a diagnostic tool to detect injuries (*i.e.*, body sensor)[29] that can cooperate with drug delivery systems for rapid individual treatment.[32]

Security sensor. Besides applications in the biomedical area, enzyme-based logic gates have also proven to be a promising tool in the security area. A biomolecular keypad lock was designed with a network of three concatenated **AND** logic gates.[33] The most important feature of this network is the dependence of the output signal on both the input signal combination and the correct input order. Compared to conventional security keypad locks, the enzyme-based keypad lock has a greater potential to incorporate biometric information into security systems. Another interesting application of enzyme-based logic gates is high-fidelity determination of security threats, including explosive compounds (2,4,6-trinitrotoluene and 2,4-dinitrotoluene) and nerve agents (paraoxon and methyl parathion).[11] This technology is a rapid, easy to use, and field-deployable threat detection system.

Transition from enzyme computing to bacteria-based biocomputing

As discussed above, researchers have developed many enzyme-based Boolean logic gates in elaborate cascades to add complexity and function.[34,35] Although enzymes are more selective and specific, the transition from enzyme-based to bacteria-based logic gates may be desirable for certain applications, especially when the computation speed is not critical. Possible advantages of bacteria-based logic gates are that they are self renewing and more robust than enzyme-based logic gates. In addition, bacteria can be integrated into a Biocomputing platform to perform more complex tasks than what enzyme cascades may be able to do (*e.g.*, they can produce complex biochemicals). Therefore, we translated an idea that was used by enzyme computing for use in Cellular Computing Circuits. As mentioned in the introduction, one of the main advances of enzyme-based Boolean logic gates is the use of electrochemically-active enzymes to generate a digital output signal as direct electric current in a biofuel cell (a type of bioelectrochemical system [BES]).[10] In this way, fluorescence measurement can be omitted from the user interface, simplifying the Biocomputing system to communicate directly.

Tam *et al.*[36] used an enzyme-based logic gate with a pH-sensitive electrode (*i.e.*, a polymer brush-modified electrode). This electrode is active only when the pH is lowered to 4.5 by enzymatic activity (triggered by the input signal). With a similar electrode setup in a microbial BES, we initiated studies to design an **AND** logic gate with a single-mutant *Pseudomonas aeruginosa* PA14 (Δ*lasI*). Unfortunately, our initial studies with this setup were unsuccessful. Regardless, we describe it here to illustrate that a more complex system with bacteria compared to enzymes creates unforeseen problems. The wild type and also this mutant of *P. aeruginosa* PA14 can generate an electric current in BESs by producing redox mediators (phenazines).[8,37,38] We chose the quorum-sensing signaling chemical 3-oxo-C12-HSL as the first input signal (signal A). Our previous work has shown that quorum sensing regulates production of phenazines, and thus electric current generation in BESs,[8] through quorum-sensing transcriptional regulation of phenazine production in *P. aeruginosa* PA14.[39] 3-oxo-C12-HSL is an input signal for our system because we used a mutant strain without the self-secretion autoinducer LasI, while the LasR receptor remained functional in this mutant. Thus, this mutant cannot produce the signal but can still detect it and trigger the quorum-sensing cascade to induce phenazine production.

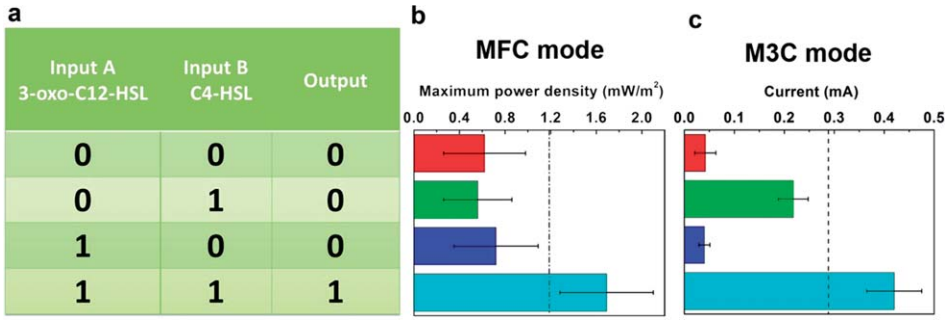


Fig. 2 Bacteria-based **AND** logic gate: **a.** Truth table for the **AND** logic gate in both microbial fuel cell (MFC) and microbial three-electrode cell (M3C) mode; **b.** maximum power density produced in MFC mode with four different input combinations; and **c.** current produced in M3C mode with four different input combinations. The output signal was above a threshold only when both input signal were present.

The pH drop for our **AND** logic gate with this single mutant and a polymer-modified, pH sensitive electrode was activated by adding an input signal (signal B) that consisted of an ester. Because of the presence of alginate in the growth medium, *P. aeruginosa* PA14 produces esterases that cleave the ester into hydrolysate acids. If this setup had worked, we would have created an **AND** logic gate because both input signals A and B would need to be present to create a functional electrode in combination with an electrochemically-active bacterium to generate one output signal (direct electric current). Instead, signal B did not function as expected; the pH drop did not activate the function of the electrode because the pH sensitive polymer and alginate aggregates were oppositely charged, resulting in an insulating coat around the electrode. Therefore, we had to omit the pH-sensitive electrode in our Cellular Computing Circuit.

As a replacement for signal B (ester with the pH-sensitive electrode), we sought another biological control of the current production and focused on quorum-sensing regulation for both input signals A and B (signal A remained 3-oxo-C12-HSL). Certain global signals (*e.g.*, glucose, temperature) were considered; however, they were not tested because these signals affect the entire regulatory system and not only the system that controls phenazine production, likely resulting in unpredictable output signals. Next, we tried taking advantage of high ferric iron concentrations to repress phenazine production by *P. aerugonosa* PA14 through quorum-sensing regulation. We operated the BES with a high concentration of ferric iron in the medium (to inhibit phenazine production) and chose EDTA, which is an iron chelator, as the second input signal (signal B) to remove the free ferric iron inhibitor with the objective to generate phenazine (current). Again, the system did not operate as planned due to the added complexity of whole bacterial cells – ferric iron had unpredictable effects on the phenazine production. Fortunately, a double mutant of *P. aeruginosa* PA14 was able to process a more specific signal B, which is described in the next paragraph.

Direct electric current with bacteria – our recent work

To obtain a more specific signal for regulating phenazine production, we chose a *P. aeruginosa* PA14 Δ*lasI*/Δ*rhlI* double mutant as the core part of the bacteria-based logic gate.[7] This made it possible to use C4-HSL as the second input signal (signal B) because, similar to 3-oxo-C12-HSL, it cannot be produced but can still be detected by the modified quorum-sensing pathway. Our results show that phenazine production, and therefore electric current generation, was dramatically upregulated in the presence of both quorum-sensing chemical signals (signals A and B). This made the system a functional bacteria-based **AND** logic gate (**1** in truth table [Fig. 2a]) because the output signal for each of the separate input signals was below a set threshold level (**0** in truth table [Fig. 2a]). This approach was successful because we used a double mutant to reduce the complexity of signal processing, and the signals chosen were highly specific to regulation of phenazine production.

The **AND** logic gate was developed in two different types of BESs: microbial fuel cells (MFCs) and microbial three-electrode cells (M3Cs). In MFCs, we use a natural difference in potential between the anode and cathode to generate an electric current, which can generate electric power when an external resistor is utilized.[40] MFCs, therefore, have a useful function to generate power and with our bacteria-based **AND** logic gate, we, in essence, developed a self-powered biosensor (Fig. 2b).[7] It is important to note here that to improve the definition of the output signal we recorded the current 115 h after adding the two input signals, which is a very slow computing speed. Thus, this speed must be further increased for the next generation of our Cellular Computing Circuits. In addition, the robustness of the output signal had to be improved. We circumvented problems with a robust output signal, which is caused by both anode and cathode performance noises that are propagated through the entire signal-processing pathway (Fig. 2b), by using a M3C. In M3Cs, we create a better-defined electrochemical environment by using a three-electrode system with potentiostatic control to set the working electrode potential. As anticipated, this resulted in improved output signal definition (Fig. 2c).

The direct digital output signal is one of the major improvements in this system compared to other Cellular Computing Circuits with analogous output signals (*e.g.*, fluorescence), which need to be detected by external equipment. We used an electrochemically-active bacterium and used its native regulatory system to control the output signal. Other electrochemically-active bacteria than *P. aeruginosa* PA14, such as *Shewanella oneidensis*, could possibly be utilized. This bacterium uses the electron shuttle riboflavin to mediate electrons between cells and electrodes,[41] and therefore controlling the production of riboflavin in *S. oneidensis* could be an alternative mechanism. Another possibility is by using genetically-modified strains of

non-electrochemically-active bacteria, such as *E. coli*, to produce a digital output signal of direct electric current.[42,43]

One possible implementation of our work is in the biomedical field, and includes monitoring of microbial communities on endotracheal tube surfaces. Endotracheal tubes are used for mechanical ventilation of patients, which is a life-saving medical strategy, but not without risks, because of the colonization of pathogens, including *P. aeruginosa* in biofilms that grow on the tubes when they are used for extended periods of time.[44] These pathogens can lead to increased rates of ventilator-associated pneumonia, which can be lethal. To combat pathogen colonization, endotracheal tubes can be modified to include the bacterial **AND** logic gate that we have developed, resulting in a smart medical apparatus with the ability to sense and report an oncoming infection. A modified endotracheal tube coupled with an engineered HSL-detection strain that is separated from the human environment could detect the onset of *P. aeruginosa* virulence. The resulting direct electrical output signal would trigger an immediate review by the doctor of the advantages and disadvantages of the mechanical ventilation strategy.

Outlook

Although Cellular Computing Circuits are still in an early stage of development, the time for application of these devices is nearing. We have discussed the useful functions of several strains and look forward to their implementation in biomedical, industrial, and environmental applications. These may include bacterial strains with quorum-sensing outputs that can regulate the activity of other microbes in a fermentor or in the human gut. Alternatively, these may also include strains with direct electric current generation (digital output signal) that sense complex chemical signals in diseased tissue, the environment, or in industrial fermentors and relay this information directly to us *via* digital communication. For example, Cellular Computing Circuits may be useful to detect bacterial contamination in fermentors for the biofuel industry. Then, these microbial processes can be converted into decision-making and even self-regulating "smart" microbial systems. The interface between biosensors and electronics will be particularly advantageous in creation of Cellular Computing Circuits that are able to both sense and act in a reasonable time frame. Through the development of Synthetic Biology, we expect to see rapid development of many functional and useful Cellular Computing Circuits as biosensors and biocontrollers.

Acknowledgements

This work was supported through NSF Career grant # 0939882 to L.T.A.. We thank Drs. John March and Hanno Richter (both at Cornell University) for their helpful comments.

References

1 J. Baumgardner, K. Acker, O. Adefuye, S. T. Crowley, W. Deloache, J. O. Dickson, L. Heard, A. T. Martens, N. Morton, M. Ritter, A. Shoecraft, J. Treece, M. Unzicker, A. Valencia, M. Waters, A. M. Campbell, L. J. Heyer, J. L. Poet and T. T. Eckdahl, *J. Biol. Eng.*, 2009, **3**, 11.
2 H. Abelson, D. Allen, D. Coore, C. Hanson, G. Homsy, J. Thomas, F. Knight, R. Nagpal, E. Rauch, G. J. Sussman and R. Weiss, *Commun. ACM*, 2000, **43**, 74–82.
3 A. Ibraheem and R. E. Campbell, *Curr. Opin. Chem. Biol.*, 2010, **14**, 30–36.
4 K. J. Herbst, Q. Ni and J. Zhang, *IUBMB Life*, 2009, **61**, 902–908.
5 S. Okumoto, *Curr. Opin. Biotechnol.*, 2010, **21**, 45–54.
6 A. E. Palmer, Y. Qin, J. G. Park and J. E. McCombs, *Trends Biotechnol.*, 2011, **29**, 144–152.
7 Z. Li, M. A. Rosenbaum, A. Venkataraman, T. K. Tam, E. Katz and L. T. Angenent, *Chem. Commun.*, 2011, **47**, 3060–3062.
8 A. Venkataraman, M. Rosenbaum, J. B. A. Arends, R. Halitschke and L. T. Angenent, *Electrochem. Commun.*, 2010, **12**, 459–462.
9 E. Katz and V. Privman, *Chem. Soc. Rev.*, 2010, **39**, 1835–1857.
10 E. Katz and I. Willner, *J. Am. Chem. Soc.*, 2003, **125**, 6803–6813.
11 M.-C. Chuang, J. R. Windmiller, P. Santhosh, G. V. Ramirez, E. Katz and J. Wang, *Chem. Commun.*, 2011, **47**, 3087–3089.
12 L. Qian and E. Winfree, *Science*, 2011, **332**, 1196–1201.
13 J. Elbaz, O. Lioubashevski, F. Wang, F. Remacle, R. D. Levine and I. Willner, *Nat. Nanotechnol.*, 2010, **5**, 417–422.
14 A. R. Cukras, D. Faulhammer, R. J. Lipton and L. F. Landweber, *BioSystems*, 1999, **52**, 35–45.
15 K. Sakamoto, D. Kiga, K. Komiya, H. Gouzu, S. Yokoyama, S. Ikeda, H. Sugiyama and M. Hagiya, *BioSystems*, 1999, **52**, 81–91.
16 J. R. van der Meer, *Bacterial sensors synthetic design and application principles*, http://www.morganclaypool.com/doi/abs/10.2200/S00312ED1V01Y201011SBI002.
17 J. H. Reif, *Science*, 2011, **332**, 1156–1157.
18 C. A. Voigt, *Curr. Opin. Biotechnol.*, 2006, **17**, 548–557.
19 A. E. Friedland, T. K. Lu, X. Wang, D. Shi, G. Church and J. J. Collins, *Science*, 2009, **324**, 1199–1202.
20 A. Tamsir, J. J. Tabor and C. A. Voigt, *Nature*, 2010, **469**, 212–215.
21 J. J. Tabor, H. M. Salis, Z. B. Simpson, A. A. Chevalier, A. Levskaya, E. M. Marcotte, C. A. Voigt and A. D. Ellington, *Cell*, 2009, **137**, 1272–1281.
22 M. B. Miller and B. L. Bassler, *Annu. Rev. Microbiol.*, 2001, **55**, 165–199.
23 T. Danino, O. Mondragon-Palomino, L. Tsimring and J. Hasty, *Nature*, 2010, **463**, 326–330.
24 K. I. Ramalingam, J. R. Tomshine, J. A. Maynard and Y. N. Kaznessis, *Biochem. Eng. J.*, 2009, **47**, 38–47.
25 J. J. Tabor, A. Levskaya and C. A. Voigt, *J. Mol. Biol.*, 2011, **405**, 315–324.
26 K. Brenner, D. K. Karig, R. Weiss and F. H. Arnold, *Proc. Natl. Acad. Sci. U. S. A.*, 2007, **104**, 17300–17304.
27 P. E. M. Purnick and R. Weiss, *Nat. Rev. Mol. Cell Biol.*, 2009, **10**, 410–422.
28 G. Batt, B. Yordanov, R. Weiss and C. Belta, *Bioinformatics*, 2007, **23**, 2415–2422.
29 J. R. Windmiller, G. Strack, M.-C. Chuang, J. Halámek, P. Santhosh, V. Bocharova, J. Zhou, E. Katz and J. Wang, *Sens. Actuators, B*, 2010, **150**, 285–290.
30 K. M. Manesh, J. Halámek, M. Pita, J. Zhou, T. K. Tam, P. Santhosh, M.-C. Chuang, J. R. Windmiller, D. Abidin, E. Katz and J. Wang, *Biosens. Bioelectron.*, 2009, **24**, 3569–3574.
31 J. Halámek, J. R. Windmiller, J. Zhou, M.-C. Chuang, P. Santhosh, G. Strack, M. A. Arugula, S. Chinnapareddy, V. Bocharova, J. Wang and E. Katz, *Analyst*, 2010, **135**, 2249–2259.
32 I. Tokarev, V. Gopishetty, J. Zhou, M. Pita, M. Motornov, E. Katz and S. Minko, *ACS Appl. Mater. Interfaces*, 2009, **1**, 532–536.
33 G. Strack, M. Ornatska, M. Pita and E. Katz, *J. Am. Chem. Soc.*, 2008, **130**, 4234–4235.
34 L. Amir, T. K. Tam, M. Pita, M. M. Meijler, L. Alfonta and E. Katz, *J. Am. Chem. Soc.*, 2009, **131**, 826–832.
35 E. Katz and M. Pita, *Chem.–Eur. J.*, 2009, **15**, 12554–12564.
36 T. K. Tam, M. Ornatska, M. Pita, S. Minko and E. Katz, *J. Phys. Chem. C*, 2008, **112**, 8438–8445.
37 K. Rabaey, N. Boon, S. D. Siciliano, M. Verhaege and W. Verstraete, *Appl. Environ. Microbiol.*, 2004, **70**, 5373–5382.
38 A. Venkataraman, M. A. Rosenbaum, S. D. Perkins, J. J. Werner and L. T. Angenent, *Energy Environ. Sci.*, 2011, DOI: 10.1039/C1EE01377G.
39 M. Whiteley, K. M. Lee and E. P. Greenberg, *Proc. Natl. Acad. Sci. U. S. A.*, 1999, **96**, 13904–13909.

40 J. J. Fornero, M. Rosenbaum and L. T. Angenent, *Electroanalysis*, 2010, **22**, 832–843.
41 E. Marsili, D. B. Baron, I. D. Shikhare, D. Coursolle, J. A. Gralnick and D. R. Bond, *Proc. Natl. Acad. Sci. U. S. A.*, 2008, **105**, 3968–3973.
42 M. Rosenbaum and L. T. Angenent, in *Bioelectrochemical systems: from extracellular electron transfer to biotechnological application*, ed. K. Rabaey, L. T. Angenent, U. Schröder and J. Keller, IWA Publishing, London/New York, 2010, ch. 6, pp. 101–117.
43 H. M. Jensen, A. E. Albers, K. R. Malley, Y. Y. Londer, B. E. Cohen, B. A. Helms, P. Weigele, J. T. Groves and C. M. Ajo-Franklin, *Proc. Natl. Acad. Sci. U. S. A.*, 2010, **107**, 19213–19218.
44 S. D. Perkins, K. F. Woeltje and L. T. Angenent, *Int. J. Med. Microbiol.*, 2010, **300**, 503–511.
45 L. Kari and G. Rozenberg, *Commun. ACM*, 2008, **51**, 72–83.
46 G. Păun, in *Fundamentals of Computation Theory*, ed. A. Lingas and B. Nilsson, Springer Berlin/Heidelberg, 2003, vol. 2751, pp. 177–220.
47 L. Adleman, *Science*, 1994, **266**, 1021–1024.
48 M. Pita, G. Strack, K. MacVittie, J. Zhou and E. Katz, *J. Phys. Chem. B*, 2009, **113**, 16071–16076.
49 R. Baron, O. Lioubashevski, E. Katz, T. Niazov and I. Willner, *J. Phys. Chem. A*, 2006, **110**, 8548–8553.
50 A. Okamoto, K. Tanaka and I. Saito, *J. Am. Chem. Soc.*, 2004, **126**, 9458–9463.
51 M. N. Stojanovic, T. E. Mitchell and D. Stefanovic, *J. Am. Chem. Soc.*, 2002, **124**, 3555–3561.
52 A. Saghatelian, N. H. Völcker, K. M. Guckian, V. S. Y. Lin and M. R. Ghadiri, *J. Am. Chem. Soc.*, 2003, **125**, 346–347.
53 J. C. Anderson, C. A. Voigt and A. P. Arkin, *Mol. Syst. Biol.*, 2007, **3**.
54 O. Rackham and J. W. Chin, *J. Am. Chem. Soc.*, 2005, **127**, 17584–17585.
55 J. Zhou, M. A. Arugula, J. Halaámek, M. Pita and E. Katz, *J. Phys. Chem. B*, 2009, **113**, 16065–16070.
56 M. Pita and E. Katz, *J. Am. Chem. Soc.*, 2008, **130**, 36.
57 X. Su and L. M. Smith, *Nucleic Acids Res.*, 2004, **32**, 3115–3123.
58 J.-H. Guo, D.-M. Kong and H.-X. Shen, *Biosens. Bioelectron.*, 2010, **26**, 327–332.

