Int. Journ. of Unconventional Computing, Vol. 8, pp. 509–515
Reprints available directly from the publisher
Photocopying permitted by license only

Engineering Bacterial Biocomputers Using A Top-Down Approach and Dynamic Genetic Circuits

MICHAELA A. TERAVEST AND LARGUS T. ANGENENT*

**Department of Biological and Environmental Engineering,*
Cornell University, Ithaca, NY 14853, USA
Email: la249@cornell.edu

Received: September 25, 2012. Accepted: October 22, 2012.

Biocomputing is a broad field of research encompassing any area at the intersection of computing and biology. Bacterial biocomputing is a branch of biocomputing focused on generating bacterial strains with useful logical and sensing functions, using chemical, electrical, or physical input or output signals. Significant advances have been made in the last decade; however, there are major obstacles, including a complete lack of programmability and generalizability of all strains created to date. Two reasons for this problem are: i) the limited variety and function of standard genetic parts; and ii) the static nature of current design strategies. We propose a top-down genetic engineering approach and *in situ* genetic remodeling mechanisms as possible steps to advance bacterial biocomputer designs.

Keywords: Biocomputing, bacteria, genetic circuit, genetic engineering, quorum sensing, microbial electrochemistry

INTRODUCTION

Biocomputing is a new and quickly expanding field of research. While biocomputing speeds are not expected to rival those of conventional computing, there are other benefits of computing in biological systems, namely massively parallel processing and biocompatibility. Because every cell can be considered an independent processor and bacterial cultures contain millions or billions of cells, there may be significant opportunities in biocomputing for

solving specialized problems requiring highly parallel processing (1). Biocompatibility offers broad benefits to biocomputing, allowing biocomputers to: i) contain integrated sensors for complex biomolecules; and ii) exist and function under physiological conditions and possibly even within the human body as part of a medical device (2). These advantages make biocomputing attractive for data collection, processing, and decision making within medical devices and in environmental sensing applications.

Several platforms are being explored for biocomputing, including DNA, RNA, enzymes, and whole cells. Each platform has its own advantages and disadvantages; here we will discuss the challenges and benefits of sensing and computing using whole bacterial cells. The major disadvantages of using whole bacterial cells include slow response times, necessity to feed and maintain the cells, longer design times, and increased difficulty of implementation. In addition, engineered bacterial cells may not be compatible with implantable devices or release to the environment because of the risk of pathogenicity or contamination of wild microbial consortia with engineered DNA. However, there are also unique benefits of whole bacterial cells as biocomputers. Unlike nucleic acids and enzymes, bacterial cells are capable of self-maintenance and replication, a distinct benefit for continuity of the system, especially under conditions which may be damaging to purified DNA or enzymes. Further, bacterial hosts not only maintain biocomputing function over time, but may also improve system function through directed evolution (3). Finally, whole cells exhibit coordinated behaviors more complex than anything possible for single molecules, and these behaviors can be co-opted for use in biocomputing systems, rather than engineered *de novo*.

Thus far, however, the majority of bacterial biocomputers have not utilized native bacterial functions or pathways, but rather used a "bottom-up" engineering approach to build genetic circuits from standardized genetic parts. This approach has been successful in producing a wide variety of logic gates with lactose or antibiotic inputs and fluorescent protein outputs (4). Other inputs, such as white and colored light have also been used, but to date the variety of inputs and outputs has been limited due to the small number of well-characterized genetic parts. Although the majority of bacterial biocomputing systems produced to date use this bottom-up approach (5-10), some rely on the native behavior of bacteria, providing a proof-of concept for use of native pathways in biocomputing (11-13). See our recent perspective for a thorough review of current bacterial biocomputing systems (4).

CHALLENGES TO APPLICATION

Because of the complexity of whole cell systems, design times are long and there is much trial and error involved in producing functional genetic circuits for bacterial biocomputing. The difficulty of implementation is made clear by

the disparity between proposed genetic circuits and successfully developed biocomputing strains. For example, genetic circuits for associative memory have been proposed multiple times, but this behavior has not yet been produced in engineered bacterial strains (14-15). Libraries of genetic parts and cellular modeling are improving, but these efforts have not yet closed the gap between desired biocomputing functions and synthetic biology capabilities.

Even as libraries of genetic parts expand, the difficulty of design and implementation of genetic circuits has encouraged most researchers to work with the best characterized genetic parts that respond to lactose or antibiotics. While this is important for development of "proof of concept" strains, it is a hindrance to practical application of bacterial biocomputers because sensing of these particular input signals is not useful for most medical or environmental applications. Although bacteria are capable of sensing many other chemical and physical parameters, genetic parts for doing so are not yet characterized well enough to be integrated in current genetic design strategies. Lu et al. (15) suggested improving libraries of standard genetic parts; however, there are major disadvantages of these libraries, such as the registry of standard genetic parts. First, these libraries are geared toward engineering in *Escherichia coli*, and many of the parts will not be functional in other bacterial hosts. Second, the libraries encourage a very specific strategy for construction of plasmids and genetic circuits, which can limit the variety of genetic network designs. Because of these limitations it is also important to consider other design strategies, including utilization of native bacterial pathways with a "top-down" engineering approach.

Bacterial pathways evolve over long periods of time and become optimized for function in their native host and evolutionary fine-tuning creates complex systems that current genetic design methods cannot recreate. Rather than extracting useful bacterial behaviors from their optimized environment, it may be more practical to use them within their native host, and minimize genetic engineering changes. This top-down approach could allow production of more versatile biocomputers with less time spent on genetic engineering.

TOP-DOWN ENGINEERING USING NATIVE BACTERIAL BEHAVIORS

An excellent example of a microbial trait conducive to this approach is the production of electrical current by electrochemically active bacteria (EAB). This is a complex behavior, and detailed mechanisms are still only understood for a few bacterial species. Recently, efforts have been made to introduce the current production pathway of *Shewanella oneidensis* into *E. coli,* and while the necessary pathway was successfully transformed, expressed, and assembled, current production was orders of magnitude lower than for

wild-type *S. oneidensis* (16). In contrast, a microbial AND gate using *S. oneidensis* to produce an electrical output signal has already been reported by Arugula et al. (13). In addition, our group had already shown that native electricity production pathways in *Pseudomonas aeruginosa* can be logically controlled by external chemical outputs with minimal genetic manipulation (11). This suggests that complex behaviors like current production may be easier to manipulate *in situ* than to extract and use in heterologous host organisms.

Our bacterial biocomputer was a successful proof of concept for an AND logic gate with chemical input signals and an electrical output signal; however, the adaptability of this system is severely constrained. Because the output was controlled using native genetic pathways of *P. aeruginosa* with only two genes knocked out, the options for changing the input signals or logic behavior of the system are limited to the native pathways that control current production and are understood well enough to be successfully manipulated. There is no general rule for changing the programming of this biocomputer, and while using *E. coli* would expand the range of options somewhat, the library of parts is still limited. Because both bottom-up and top-down approaches have benefits and disadvantages it will be important to consider all possible methods for engineering genetic circuits as bacterial biocomputing progresses.

Summary of a top-down engineering approach

Exploiting native behavior can be a hindrance because of limited functionality existing in non-engineered strains. However, it is a major benefit in terms of time and effort required to produce a new biocomputing system, which has been comprehensively reviewed previously (15). Genetic engineering can be a long and difficult process, and in the end, only a single, static behavior is introduced into the organism and "programming" to change input and output signals or behavior is not possible (15). Here, we propose a strategy to take one step closer toward generating programmable strains by incorporating genetic remodeling mechanisms into the genetic circuits.

GENETIC REMODELING MECHANISMS

While most current biocomputing circuits have had a static architecture, bacterial genomes can be remodeled, and one group has taken advantage of this to create a dynamic genetic circuit, using DNA segments that can be reoriented within the genome (10). Broader application of this dynamic circuit design would allow production of simple "programmable" bacterial biocomputers that could express different logic or sensing functions without additional genetic modification. The major genetic remodeling pathways are: i) inversion; ii) movement of transposons; and iii) homologous recombination.

These pathways allow the genetic rearrangement to occur *in situ* rather than requiring tedious plasmid preparation, transformation, and selection for each biocomputing behavior.

Inversion

Inversion is the simplest mechanism of genetic remodeling and constitutes the flipping of a DNA sequence between two inverted signal sequences (Figure 1) (17). This activity is accomplished by an invertase enzyme, which recognizes, and acts on specific DNA signal sequences. The signal sequences are typically inverted repeats (a recurring motif in DNA remodeling) flanking the sequence to be inverted. Inversion is a useful mechanism, because it can create permanent switching between functional and nonfunctional, or two different functional states. For programming-like function in genetic circuits, the inversion step can be controlled through transient expression of the invertase, and protease tags can be used to help create discrete switching steps (10).

Movement of transposons

Transposition is another form of DNA remodeling, which involves movement of small DNA fragments (transposons) from one position to another in the genome. Some transposons insert themselves at specific target sequences

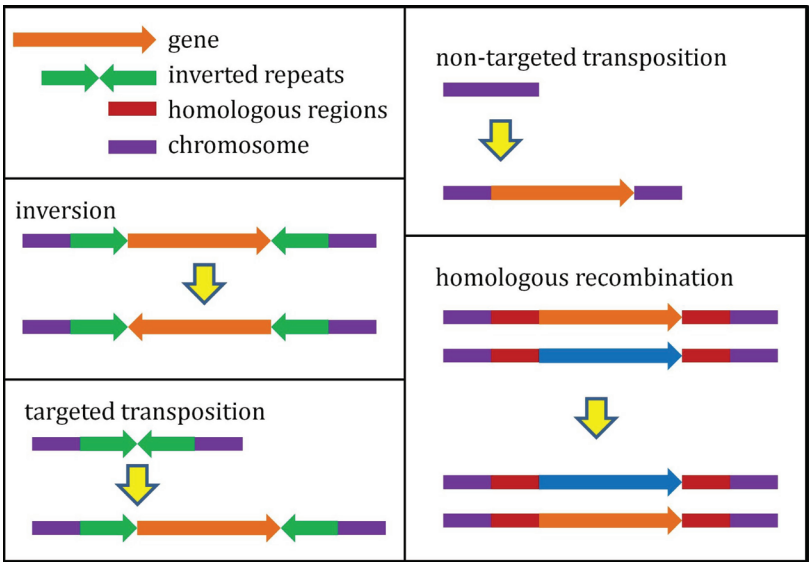

FIGURE 1

Graphical representations of common genetic remodeling mechanisms. Here, we show these mechanisms as three different genetic remodeling pathways: i) inversion; ii) movement of transposons, which we further divide in targeted transposition and non-targeted transposition; and iii) homologous recombination.

(often inverted repeats), while others may insert randomly in the genome (Figure 1) (17). Transposons could be used to inactivate genetic circuits in response to specific signals, or to insert functional genes into a genetic circuit when they are needed. Targeted transposons could also potentially be used to disrupt specific native cellular pathways and co-opt them into the synthetic genetic network *in vivo*. Engineering transposon specificity to recombine at specific functional sites (such as promoters) could create an extremely useful tool for controlling native bacterial behaviors. Combining such semi-specific transposons with clever selection could potentially yield a high-throughput method for production of a wide range of sensor-output strains.

Homologous recombination

The third useful DNA remodeling mechanism is homologous DNA recombination. Homologous recombination is a process by which two double stranded DNA molecules with homologous regions may swap a segment of the sequence (Figure 1) (17). This mechanism is already an essential tool for genetic engineering, but it has not been used to regulate functional behavior of genetic circuits. Homologous recombination can be much more specific and also move larger regions of DNA than either of the other two mechanisms. With complex genetic networks it may be necessary to move large portions of the DNA rather than just the transposition or inversion machinery.

Summary of a DNA remodeling approach

DNA remodeling mechanisms are advantageous in that they allow genetic remodeling to occur only under tightly controlled circumstances. By including inducible promoters or protein sequestered DNA sequences in the homologous regions, recombination could be repressed until controlled conditions allow the desired genetic rearrangement. This would allow for additional layers of complexity in the design of genetic circuits and open up possibilities for more applicable functions and simple programming. Integrated genetic remodeling could make strains with multiple different functions under different conditions possible, thus, relieving some of the need to produce unique strains for each desired biocomputing function.

OUTLOOK

Even though there has been considerable advancement in the field of bacterial biocomputing, there is still no approach for the creation of a generalizable bacterial biocomputing platform. Currently, it is unclear whether this is more likely to occur through improved synthetic biology technologies or through better exploitation of existing biological mechanisms. We propose an increased focus on use of native bacterial pathways, and further, applications of *in situ* genetic remodeling mechanisms to alter existing genetic networks.

Both of these approaches have the potential to improve design, implementation, and performance of bacterial biocomputers using existing technology.

REFERENCES

[1] Baumgardner, J., Acker, K., Adefuye, O., Crowley, S. T., Deloache, W., Dickson, J. O., Heard, L., Martens, A. T., Morton, N., Ritter, M., Shoecraft, A., Treece, J., Unzicker, M., Valencia, A., Waters, M., Campbell, A. M., Heyer, L. J., Poet, J. L., and Eckdahl, T. T. (2009) Solving a Hamiltonian Path Problem with a bacterial computer. *J. Biol. Eng.* **3**, 11.

[2] Manesh, K. M., Halámek, J., Pita, M., Zhou, J., Tam, T. K., Santhosh, P., Chuang, M.-C., Windmiller, J. R., Abidin, D., Katz, E., and Wang, J. (2009) Enzyme logic gates for the digital analysis of physiological level upon injury. *Biosensors Bioelectron.* **24**, 3569–3574.

[3] Yokobayashi, Y., Weiss, R., and Arnold, F. H. (2002) Directed evolution of a genetic circuit. *Proc. Natl. Acad. Sci. U.S.A.* **99**, 16587–16591.

[4] TerAvest, M. A., Li, Z., and Angenent, L. T. (2011) Bacteria-based biocomputing with Cellular Computing Circuits to sense, decide, signal, and act. *Energy Environ. Sci.* **4**, 4907–4916.

[5] Danino, T., Mondragon-Palomino, O., Tsimring, L., and Hasty, J. (2010) A synchronized quorum of genetic clocks. *Nature* **463**, 326–330.

[6] Voigt, C. A. (2006) Genetic parts to program bacteria. *Curr. Opin. Biotechnol.* **17**, 548-557.

[7] Tabor, J. J., Levskaya, A., and Voigt, C. A. (2011) Multichromatic Control of Gene Expression in Escherichia coli. *J. Mol. Biol.* **405**, 315–324.

[8] Tamsir, A., Tabor, J. J., and Voigt, C. A. (2011) Robust multicellular computing using genetically encoded NOR gates and chemical 'wires'. *Nature* **469**, 212–215.

[9] You, L., Cox, R. S., Weiss, R., and Arnold, F. H. (2004) Programmed population control by cell-cell communication and regulated killing. *Nature* **428**, 868–871.

[10] Friedland, A. E., Lu, T. K., Wang, X., Shi, D., Church, G., and Collins, J. J. (2009) Synthetic Gene Networks That Count. *Science* **324**, 1199–1202.

[11] Li, Z. J., Rosenbaum, M. A., Venkataraman, A., Tam, T. K., Katz, E., and Angenent, L. T. (2011) Bacteria-based AND logic gate: a decision-making and self-powered biosensor. *Chem. Comm.* **47**, 3060–3062.

[12] Yuan, Y., Zhou, S., Zhang, J., Zhuang, L., Yang, G., and Kim, S. (2012) Multiple logic gates based on reversible electron transfer of self-organized bacterial biofilm. *Electrochem. Commun.* **18**, 62–65.

[13] Arugula, M. A., Shroff, N., Katz, E., and He, Z. (2012) Molecular AND logic gate based on bacterial anaerobic respiration. *Chem. Comm.* **48,** 10174–10176.

[14] Fernando, C. T., Liekens, A. M. L., Bingle, L. E. H., Beck, C., Lenser, T., Stekel, D. J., and Rowe, J. E. (2009) Molecular circuits for associative learning in single-celled organisms. *J. R. Soc. Interface* **6**, 463–469.

[15] Lu, T. K., Khalil, A. S., and Collins, J. J. (2009) Next-generation synthetic gene networks. *Nat. Biotechnol.* **27**, 1139–1150.

[16] Jensen, H. M., Albers, A. E., Malley, K. R., Londer, Y. Y., Cohen, B. E., Helms, B. A., Weigele, P., Groves, J. T., and Ajo-Franklin, C. M. (2010) Engineering of a synthetic electron conduit in living cells. *Proc. Natl. Acad. Sci. U.S.A.* **107,** 19213–19218.

[17] Rocha, E. P. C. (2008) The Organization of the Bacterial Genome. in *Annu. Rev. Genet.*, Annual Reviews, Palo Alto. pp 211–233.