Bacterial Computing

MARTYN AMOS

Manchester Metropolitan University, United Kingdom

Invited article for the Encyclopedia of Complexity and System Science, Springer, 2008

Article Outline

Glossary

DNA	Deoxyribonucleic acid. Molecule that encodes the genetic information of cellular organisms.
Operon	Set of functionally related genes with a common promoter ("on switch").
Plasmid	Small circular DNA molecule used to transfer genes from one organism to another.
RNA	Ribonucleic acid. Molecule similar to DNA, which helps in the conversion of genetic information to proteins.
Transcription	Conversion of a genetic sequence into RNA.
Translation	Conversion of an RNA sequence into an amino acid sequence (and, ultimately, a protein).

I Definition of the Subject and Its Importance

Bacterial computing is a conceptual subset of *synthetic biology*, which is itself an emerging scientific discipline largely concerned with the *engineering* of biological systems. The goals of synthetic biology may be loosely partioned into four sets: (1) To better understand the fundamental operation of the biological system being engineered, (2) To extend synthetic chemistry, and create improved systems for the synthesis of molecules, (3) To investigate the "optimization" of existing biological systems for human purposes, (4) To develop and apply rational *engineering* principles to the design and construction of biological systems. It is on these last two goals that we focus in the current article.

The main benefits that may accrue from these studies are both theoretical and practical; the construction and study of synthetic biosystems could improve our quantitative understanding of the fundamental underlying processes, as well as suggesting plausible applications in fields as diverse as pharmaceutical synthesis and delivery, biosensing, tissue engineering, bionanotechnology, biomaterials, energy production and environmental remediation.

II Introduction

Complex natural processes may often be described in terms of networks of computational components, such as Boolean logic gates or artificial neurons. The interaction of biological molecules and the flow of information controlling the development and behavior of organisms is particularly amenable to this approach, and these models are well-established in the biological community. However, only relatively recently have papers appeared proposing the use of such systems to perform useful, *human-defined* tasks. For example, rather than merely using the network analogy as a convenient technique for clarifying our understanding of complex systems, it may now be possible to harness the power of such systems for the purposes of computation.

Despite the relatively recent emergence of biological computing as a distinct research area, the link between biology and computer science is not a new one. Of course, for years biologists have used computers to store and analyze experimental data. Indeed, it is widely accepted that the huge advances of the Human Genome Project (as well as other genome projects) were only made possible by the powerful computational tools available. Bioinformatics has emerged as "the science of the 21st century", requiring the contributions of truly interdisciplinary scientists who are equally at home at the lab bench or writing software at the computer.

However, the seeds of the relationship between biology and computer science were sown over fifty years ago, when the latter discipline did not even exist. When, in the 17th century, the French mathematician and philosopher René Descartes declared to Queen Christina of Sweden that animals could be considered a class of machines, she challenged him to demonstrate how a clock could reproduce. Three centuries later in 1951, with the publication of *"The General and Logical Theory of Automata"* [38] John von Neumann showed how a machine could indeed construct a copy of itself. Von Neumann believed that the behavior of *natural* organisms, although orders of magnitude more complex, was similar to that of the most intricate machines of the day. He believed that life was based on *logic*. We now begin to look at how this view of life may be used, not simply as a useful analogy, but as the practical foundation of a whole new engineering discipline.

III Motivation for Bacterial Computing

Here we consider the main motivations behind recent work on bacterial computing (and, more broadly, synthetic biology). Before recombinant DNA technology made it possible to construct new genetic sequences, biologists were restricted to crudely "knocking out" individual genes from an organism's genome, and then assessing the damage caused (or otherwise). Such knock-outs gradually allowed them to piece together fragments of causality, but the process was very time-consuming and error-prone.

Since the dawn of genetic engineering – with the ability to synthesise novel gene segments – biologists have been in a position to make much more finely-tuned modifications to their organism of choice, this generating much more refined data. Other advances in biochemistry have also contributed, allowing scientists to – for example – investigate new types of genetic systems with, for example, *twelve* bases, rather than the traditional four [14]. Such creations have yielded valuable insights into the mechanics of mutation, adaptation and evolution. Researchers in synthetic biology are now extending their work beyond the synthesis of single genes, and are now introducing whole new gene *complexes* into organisms.

The objectives behind this work are both theoretical and practical. As Benner and Seymour argue [5], "...a synthetic goal forces scientists to cross uncharted ground to encounter and solve problems that are not easily encountered through [top-down] analysis. This drives the emergence of new paradigms ["world views"] in ways that analysis cannot easily do." Drew Endy agrees. "Worst-case scenario, it's a complementary approach to traditional discovery science" [18]. "Best-case scenario, we get systems that are simpler and easier to understand..." Or, to put it bluntly, "Let's build new biological systems – systems that are easier to understand because we made them that way" [31]. As well as shedding new light on the underlying biology, these novel systems may well have significant paractical utility. Such new creations, according to Endy's "personal wish list" might include "generating biological machines that could clean up toxic waste, detect chemical weapons, perform simple computations, stalk cancer cells, lay down electronic circuits, synthesize complex compounds and even produce hydrogen from sunlight" [18]. In the next Section we begin to consider how this might be achieved, by first describing the underlying logic of genetic circuitry.

IV The Logic of Life

Twenty years after von Neumann's seminal paper, Francois Jacob and Jacques Monod identified *specific* natural processes that could be viewed as behaving according to logical principles:

"The logic of biological regulatory systems abides not by Hegelian

> laws but, like the workings of computers, by the propositional algebra of George Boole." [29]

This conclusion was drawn from earlier work of Jacob and Monod [30]. In addition, Jacob and Monod described the "lactose system" [20], which is one of the archetypal examples of a Boolean biosystem. We describe this system shortly, but first give a brief introduction to the operation of genes in general terms.

DNA as the Carrier of Genetic Information

The *central dogma* of molecular biology [9] is that DNA produces RNA, which in turn produces proteins. The basic "building blocks" of genetic information are known as *genes.* Each gene codes for one specific *protein* and may be turned on (*expressed*) or off (*repressed*) when required.

Transcription and translation

We now describe the processes that determine the structure of a protein, and hence its function. Note that in what follows we assume the processes described occur in bacteria, rather than in higher organisms such as humans. For a full description of the structure of the DNA molecule, see the chapter on *DNA computing.* In order for a DNA sequence to be converted into a protein molecule, it must be read (*transcribed*) and the transcript converted (*translated*) into a protein. Transcription of a gene produces a *messenger RNA* (mRNA) copy, which can then be translated into a protein.

Transcription proceeds as follows. The mRNA copy is synthesized by an enzyme known as *RNA polymerase.* In order to do this, the RNA polymerase must be able to recognize the specific region to be transcribed. This specificity requirement facilitates the regulation of genetic expression, thus preventing the production of unwanted proteins. Transcription begins at specific sites within the DNA sequence, known as *promoters.* These promoters may be thought of as "markers", or "signs", in that they are not transcribed into RNA. The regions that *are* transcribed into RNA (and eventually translated into protein) are referred to as *structural* genes. The RNA polymerase recognizes the promoter, and transcription begins. In order for the RNA polymerase to begin transcription, the double helix must be opened so that the sequence of bases may be read. This opening involves the breaking of the hydrogen bonds between bases. The RNA polymerase then moves along the DNA *template* strand in the $3 \rightarrow 5$' direction. As it does so, the polymerase creates an *antiparallel* mRNA chain (that is, the mRNA strand is the equivalent of the Watson-Crick complement of the template). However, there is one significant difference, in that RNA contains uracil instead of thymine. Thus, in mRNA terms, "U binds with A."

The RNA polymerase moves along the DNA, the DNA re-coiling into its double-helix structure behind it, until it reaches the end of the region to be transcribed. The end of this region is marked by a *terminator* which, like the promoter, is not transcribed.

Genetic regulation

Each step of the conversion, from stored information (DNA), through mRNA (messenger), to protein synthesis (effector), is itself catalyzed by other effector molecules. These may be enzymes or other factors that are required for a process to continue (for example, sugars). Consequently, a loop is formed, where products of one gene are required to produce further gene products, and may even influence that gene's own expression. This process was first described by Jacob and Monod in 1961 [20], a discovery that earned them a share of the 1965 Nobel Prize in Physiology or Medicine.

Genes are composed of a number of distinct regions, which control and encode the desired product. These regions are generally of the form promoter–gene–terminator. Transcription may be regulated by effector molecules known as *inducers* and *repressors*, which interact with the promoter and increase or decrease the level of transcription. This allows effective control over the expression of proteins, avoiding the production of unnecessary compounds. It is important to note at this stage that, in reality, genetic regulation does not conform to the digital "on-off" model that is popularly portrayed; rather, it is continuous or analog in nature.

The *Lac* operon

One of the most well-studied genetic systems is the *lac operon*. An *operon* is a set of functionally related genes with a common promoter. An example of this is the *lac* operon, which contains three structural genes that allow *E.coli* to utilize the sugar lactose.

When *E.coli* is grown on the sugar glucose, the product of the (separate, and unrelated to the *lac* operon) *lacI* gene *represses* the transcription of the *lacZYA* operon (i.e., the operon is turned off). However, if lactose is supplied *together with* glucose, a lactose by-product is produced which interacts with the repressor molecule, preventing it from repressing the *lacZYA* operon. This derepression does not itself initiate transcription, since it would be inefficient to utilize lactose if the more common sugar glucose were still available. The operon is *positively* regulated (i.e., "encouraged") by a different molecule, whose level increases as the amount of available glucose decreases. Therefore, if lactose were present as the sole carbon source, the *lacI* repression would be relaxed and the high "encouraging" levels would activate transcription, leading to the synthesis of the *lacZYA* gene products. Thus, the promoter is under the control of two sugars, and the *lacZYA* operon is only transcribed when lactose is *present* and glucose is *absent.*

In essence, Jacob and Monod showed how a gene may be thought of (in very abstract terms) as a binary switch, and how the state of that switch might be affected by the presence or absence of certain molecules. Monod's point, made in his classic book *Chance and Necessity* and quoted above, was that the workings of biological systems operate not by Hegel's philosophical or metaphysical logic of understanding, but according to the formal, mathematically-grounded logical system of George Boole.

What Jacob and Monod found was that the transcription of a gene may be regulated by molecules known as *inducers* and *repressors*, which either increase or decrease the "volume" of a gene (corresponding to its level of transcription, which isn't always as clear cut and binary as Monod's quote might suggest). These molecules interact with the promoter region of a gene, allowing the gene's level to be finely "tuned". The *lac* genes are so-called because, in the *E. coli* bacterium, they combine to produce a variety of proteins that allow the cell to metabolise the sugar lactose (which is most commonly found in milk, hence the derivation from the Latin, *lact*, meaning milk).

For reasons of efficiency, these proteins should only be produced (i.e., the genes be turned on) when lactose is present in the cell's environment. Making these proteins when lactose is *absent* would be a waste of the cell's resources, after all. However, a different sugar – glucose – will *always* be preferable to lactose, if the cell can get it, since glucose is an "easier" form of sugar to metabolise. So, the input to and output from the *lac* operon may be expressed as a truth table, with G and L standing for glucose and lactose (1 if present, 0 if absent), and O standing for the output of the operon (1 if on, 0 if off):

G	L	O
0	0	0
0	1	1
1	0	0
1	1	0

The Boolean function that the *lac* operon therefore *physically* computes is (L AND (NOT G)), since it only outputs 1 if L=1 (lactose present) *and* G=0 (glucose is absent). By showing how one gene could affect the expression of another – just like a transistor feeds into the input of another and affects its state – Jacob and Monod laid the foundations for a new way of thinking about genes; not simply in terms of protein blueprints, but of *circuits* of interacting parts, or dynamic networks of interconnected switches and logic gates. This view of the genome is now well-established [22, 23, 27], but in the next Section we show how it might be used to guide the *engineering* of biological systems.

V Rewiring Genetic Circuitry

A key difference between the wiring of a computer chip and the circuitry of the cell is that "electronics engineers know exactly how resistors and capacitors are wired to each other because they installed the wiring. But biologists often don't have a complete picture. They may not know which of thousands of genes and proteins are interacting at a given moment, making it hard to predict how circuits will behave inside cells" [12]. This makes the task of reengineering vastly more complex.

Rather than trying assimilate the huge amounts of data currently being generated by the various genome projects, synthetic biologists are taking a novel route – simplify and build. "They create models of genetic circuits, build the circuits, see if they work, and adjust them if they don't – learning about biology in the process. 'I view it as a reductionist approach to systems biology,' says biomedical engineer Jim Collins of Boston University" [12].

The field of *systems biology* has emerged in recent years as an alternative to the traditional reductionist way of doing science. Rather than simply focussing on a single level of description (such as individual proteins), researchers are now seeking to *integrate* information from many different layers of complexity. By studing how different biological components *interact*, rather than simply looking at their structure, systems biologists are attempting to build models of systems from the bottom up. A model is simply an abstract description of how a system operates – for example, a set of equations that describe how a disease spreads throughout a population. The point of a model is to capture the *essence* of a system's operation, and it should therefore be as simple as possible. Crucially, a model should also be capable of making *predictions*, which can then be tested against reality using real data (for example, if infected people are placed in quarantine for a week, how does this affect the progress of the disease in question, or what happens if I feed this signal into the chip?). The results obtained from these tests may then feed back in to further refinements of the model, in a continuous cycle of improvement.

When a model suggests a plausible structure for a synthetic genetic circuit, the next stage is to engineer it into the chosen organism, such as a bacterium. Once the structure of the DNA molecule was elucidated and the processes of transcription and translation were understood, molecular biologists were frustrated by the lack of suitable experimental techniques that would facilitate more detailed examination of the genetic material. However, in the early 1970s, several techniques were developed that allowed previously impossible experiments to be carried out (see [8, 33]). These techniques quickly led to the first ever successful cloning experiments [19, 25]. Cloning is generally defined as "... the production of multiple identical copies of a single gene, cell, virus, or organism." [35]. This is achieved as follows: a specific sequence (corresponding, perhaps, to a novel gene) is inserted in a circular DNA molecule, known as a *plasmid*, or *vector*, producing a *recombinant DNA molecule*. The vector acts as a *vehicle*, transporting the sequence into a *host* cell (usually a bacterium, such as *E.coli*). Cloning single genes is well-established, but is often done on an *ad hoc* basis. If biological computing is to succeed, it requires some degree of *standardisation*, in the same way that computer manufacturers build different computers, but using a standard library of components. "Biobricks are the first example of standard biological parts," explains Drew Endy [21]. "You will be able to use biobricks to program systems that do whatever biological systems do." He continues. "That way, if in the future, soemone asks me to make an organism that, say, counts to 3,000 and then turns left, I can grab the parts I need off the shelf, hook them together and predict how they will perform" [15].

Each biobrick is a simple component, such as an AND gate, or an inverter (NOT). Put them together one after the other, and you have a NAND (NOT-AND) gate, which is all that is needed to build any Boolean circuit (an arbitrary circuit can be translated to an equivalent circuit that uses only NAND gates. It will be much bigger than the original, but it will compute the same function. Such considerations were important in the early stages of integrated circuits, when building *different* logic gates was difficult and expensive). Just as transistors can be used together to build logic gates, and these gates then combined into circuits, there exists a hierarchy of complexity with biobricks. At the bottom are the "parts", which generally correspond to coding regions for proteins. Then, one level up, we have "devices", which are built from parts – the oscillator of Elowitz and Leibler, for example, could be constructed from three inverter devices chained together, since all an inverter does is "flip" its signal from 1 to 0 (or vice versa). This circuit would be an example of the biobricks at the top of the conceptual tree – "systems", which are collections of parts to do a significant task (like oscillating or counting).

Tom Knight at MIT made the first 6 biobricks, each held in a plasmid ready for use. As we have stated, plasmids can be used to insert novel DNA sequences in the genomes of bacteria, which act as the "testbed" for the biobrick circuits. "Just pour the contents of one of these vials into a standard reagent solution, and the DNA will transform itself into a functional component of the bacteria," he explains [6]. Drew Endy was instrumental in developing this work further, one invaluable resource being the Registry of Standard Biological Parts [32], the definitive catalogue of new biological component. At the start of 2006, it contained 224 "basic" parts and 459 "composite" parts, with 270 parts "under construction". Biobricks are still at a relatively early stage, but "Eventually we'll be able to design and build *in silico* and go out and have things synthesized," says Jay Keasling, head of Lawrence Berkeley National Laboratory's new synthetic biology department [12].

VI Successful Implementations

Although important foundational work had been performed by Arkin and Ross as early as 1994 [2], the year 2000 was a particularly significant one for synthetic biology. In January two foundational papers appeared back-to-back in the same issue of *Nature*. "Networks of interacting biomolecules carry out many essential functions in living cells, but the 'design principles' underlying the functioning of such intracellular networks remain poorly understood, despite intensive efforts including quantitative analysis of relatively simple systems. Here we present a complementary approach to this problem: the design and construction of a synthetic network to implement a particular function" [11]. That was the introduction to a paper that Drew Endy would call "the high-water mark of a synthetic genetic circuit that does something" [12]. In the first of the two articles, Michael Elowitz and Stanislau Leibler (then both at Princeton) showed how to build microscopic "Christmas tree lights" using bacteria.

Synthetic Oscillator

In physics, an *oscillator* is a system that produces a regular, periodic "output". Familiar examples include a pendulum, a vibrating string, or a lighthouse. Linking several oscillators together in some way gives rise to *synchrony* – for example, heart cells repeatedly firing in unison, or millions of fireflies blinking on and off, seemingly as one [36].

Leibler actually had *two* articles published in the same high-impact issue of *Nature*. The other was a short communication, co-authored with Naama Barkai – also at Princeton, but in the department of Physics [3]. In their paper, titled "Circadian clocks limited by noise", Leibler and Barkai showed how a simple model of biochemical networks could oscillate reliably, even in the presence of noise. They argued that such oscillations (which might, for example, control the internal circadian clock that tells us when to wake up and when to be tired) are based on networks of *genetic regulation*. They built a simulation of a simple regulatory network using a *Monte Carlo* algorithm. They found that, however they perturbed the system, it still oscillated reliably, although, at the time, their results existed only *in silico.* The other paper by Leibler was much more applied, in the sense that they had constructed a biological circuit [11]. Elowitz and Leibler had succeeded in constructing an artificial genetic oscillator in cells, using a synthetic network of repressors. They called this construction the *repressilator.*

Rather than investigating *existing* oscillator networks, Elowitz and Leibler decided to build one entirely from first principles. They chose three repressor-promoter pairs that had already been sequenced and characterised, and first built a mathematical model in software. By running the sets of equations, they identified from their simulation results certain molecular characteristics of the components that gave rise to so-called *limit cycle* oscillations; those that are robust to perturbations. This information from the model's results lead Elowitz and Leibler to select strong promoter molecules and repressor molecules that would rapidly decay. In order to implement the oscillation, they chose three genes, each of which affected one of the others by repressing it, or turning it off. For the sake of illustration, we call the genes A, B and C. The product of gene A turns off (represses) gene B. The absence of B (which represses C) allows C to turn on. C is chosen such that it turns gene A *off* again, and the three genes loop continuously in a "daisy chain" effect, turning on and off in a repetitive cycle. However, some form of *reporting* is necessary in order to confirm that the oscillation is occurring as planned.

Green fluorescent protein (GFP) is a molecule found occurring naturally in the jellyfish *Aequorea victoria.* Biologists find it invaluable because it has one interesting property – when placed under ultraviolet light, it *glows.* Biologists quickly sequenced the gene responsible for producing this protein, as they realised that it could have many applications as a *reporter.* By inserting the gene into an organism, you have a ready-made "status light" – when placed into bacteria, they glow brightly if the gene is turned on, and look normal if it's turned off. We can think of it in terms of a Boolean circuit – if the circuit outputs the value 1, the GFP promoter is produced to turn on the light. If the value is 0, the promoter isn't produced, the GFP gene isn't expressed, and the light stays off.

Elowitz and Leibler set up their gene network so that the GFP gene would be expressed whenever gene C was turned *off* – when it was turned on, the GFP would gradually decay and fade away. They synthesised the appropriate DNA sequences and inserted them into a *plasmid*, eventually yielding a population of bacteria that blinked on and off in a repetitive cycle, like miniature lighthouses. Moreover, and perhaps most significantly, the period between flashes was longer than the time taken for the cells to divide, showing that the state of the system had been passed on during reproduction.

Synthetic Toggle Switch

Rather than model an existing circuit and then altering it, Elowitz and Leibler had taken a "bottom up" approach to learning about how gene circuits operate. The other notable paper to appear in that issue was written by Timothy Gardner, Charles Cantor and Jim Collins, all of Boston University in the US. In 2000, Gardner, Collins and Cantor observed that genetic switching (such as that observed in the lambda phage [34]) had not yet been "demonstrated in networks of non-specialised regulatory components" [13]. That is to say, at that point nobody had been able to construct a switch out of genes that hadn't already been "designed" by evolution to perform that specific task. The team had a similar philosophy to that of Elowitz and Leibler, in that their main motivation was being able to test theories about the fundamental behaviour of gene regulatory networks. "Owing to the difficulty of testing their predictions," they explained, "these theories have not, in general, been verified experimentally. Here we have integrated theory and experiment by constructing and testing a synthetic, bistable [two-state] gene circuit based on the predictions of a simple mathematical model."

"We were looking for the equivalent of a light switch to flip processes on or off in the cell," explained Gardner [10]. "Then I realized a way to do this with genes instead of with electric circuits." The team chose two genes that were mutually inhibitory – that is, each produced a molecule that would turn the other off. One important thing to bear in mind is that the system didn't have a single input. Although the team acknowledged that bistability might be possible – in theory – using only a single promoter that regulated *itself*, they anticipated possible problems with robustness and experimental tunability if they used that approach. Instead, they decided to use a system whereby each "side" of the switch could be "pressed" by a different stimulus – the addition of a chemical on one, and a change in temperature on the other. Things were set up so that if the system was in the state induced by the chemical, it would stay in that state until the temperature was changed, and would only change *back* again if the chemical was reintroduced. Importantly, these stimuli did not have to be applied continuously – a "short sharp" burst was enough to cause the switch to flip over. As with the other experiment, Gardner and his colleagues used GFP as the system state reporter, so that the cells glowed in one state, and looked "normal" in the other.

In line with the bottom-up approach, they first created a mathematical model of the system and made some predictions about how it would behave inside a cell. Within a year, Gardner had spliced the appropriate genes into the bacteria, and he was able to flip them –at will – from one state the the other. As McAdams and Arkin observed, synthetic "one way" switches had been created in the mid-1980s, but "this is perhaps the first engineered design exploiting bistability to produce a switch with capability of *reversibly* switching between two...stable states" [28]. The potential applications of such a bacterial switch were clear. As they state in the conclusion to their article, "As a practical device, the toggle switch...may find applications in gene therapy and biotechnology." They also borrowed from the language of computer programming, using an analogy between their construction and the short "applets" written in the Java language, which now allow us to download and run programs in our web browser. "Finally, as a cellular memory unit, the toggle forms the basis for 'genetic applets' – self-contained, programmable, synthetic gene circuits for the control of cell function."

Engineered Communication

Towards the end of his life, Alan Turing did some foundational work on pattern formation in nature, in an attempt to explain how zebras get their striped coats or leopards their spots. The study of *morphogenesis* (from the Greek, *morphe* – shape, and *genesis* – creation. "Amorphous" therefore means "without shape or structure") is concerned with how cells split to assume new roles and communicate with another to form very precise shapes, such as tissues and organs. Turing postulated that the diffusion of chemical signals both within and between cells is the main driving force behind such complex pattern formation [37].

Although Turing's work was mainly concerned with the processes occurring amongst cells inside a developing embryo, it is clear that chemical signalling also goes on between bacteria. Ron Weiss of Princeton University was particularly interested in *Vibrio fischeri*, a bacterium that has a symbiotic relationship with a variety of acquatic creatures, including the Hawaiian squid. This relationship is due mainly to the fact that the bacteria exhibit *bioluminescence* – they generate a chemical known as a *Luciferase* (coded by the *Lux* gene), a version of which is also found in fireflies, and which causes them to glow when gathered together in numbers. Cells within the primitive light organs of the squid draw in bacteria from the seawater and encourage them to grow. Crucially, once enough bacteria are packed into the light organ they produce a signal to tell the squid cells to stop attracting their colleagues, and only then do they begin to glow. The cells get a safe environment in which to grow, protected from competition, and the squid has a light source by which to navigate and catch prey. The mechanism by which the *Vibrio* "know" when to start glowing is known as *quorum sensing*, since there have to be sufficient "members" present for luminiscence to occur.

The bacteria secrete an *autoinducer* molecule, known as VAI (*Vibrio* Auto Inducer), which diffuses through the cell wall. The *Lux* gene (which generates the glowing chemical) needs to be activated (turned on) by a particular protein – which attracts the attention of the polymerase – but the protein can only do this with help from the VAI. Its particular 3D structure is such that it can't fit tightly onto the gene unless it's been slightly bent out of shape. Where there's enough VAI present, it locks onto the protein and alters its conformation, so that it can turn on the gene. Thus, the concentration of VAI is absolutely crucial; once a critical threshold has been passed, the bacteria "know" that there are enough of them present, and they begin to glow.

Weiss realised that this quorum-based cell-to-cell communication mechanism could provide a powerful framework for the construction of bacterial devices – imagine, for example, a tube of solution containing engineered bacteria that can be added to a sample of seawater, causing it to glow only if the concentration of a particular pollutant exceeds a certain threshold. Crucially, as we will see shortly, it also allows the possibility of generating precise "complex patterned" development.

Weiss set up two colonies of *E. coli*, one containing "sender", and the other "receivers". The idea was that the senders would generate a chemical signal made up of VAI, which could diffuse across a gap and then be picked up by the receivers. Once a strong enough signal was being communicated, the receivers would glow using GFP to say that it had been picked up. Weiss cloned the appropriate gene sequences (corresponding to a type of biobrick) into his bacteria, placed colonies of receiver cells on a plate, and the receivers started to glow in acknowledgment.

Synthetic Circuit Evolution

In late 2002, Weiss and his colleagues published another paper, this time describing how rigourous engineering principles may be brought to bear on the problem of designing and building entirely new genetic circuitry. The motivation was clear – "biological circuit engineers will have to confront their inability to predict the precise behavior of even the most simple synthetic networks, a serious shortcoming and challenge for the design and construction of more sophisticated genetic circuitry in the future" [39].

Together with colleagues Yohei Yokobayashi and Frances Arnold, Weiss proposed a two stage stategy: first, design a circuit from the bottom up, as Elowitz and others had before, and clone it into bacteria. Such circuits are highly unlikely to work first time, "because the behavior of biological components inside living cells is highly context-dependent, the actual circuit performance will likely differ from the design predictions, often resulting in a poorly performing or nonfunctional circuit." Rather than simply abandoning their design, Weiss and his team decided to then *tune* the circuit inside the cell itself, by applying the principles of evolution. By inducing mutations in the DNA that they had just introduced, they were able to slightly modify the behaviour of the circuit that it represented. Of course, many of these changes would be catastrophic, giving even worse performance than before, but, occasionally, they observed a minor imporovement. In that case, they kept the "winning" bacteria, and subjected it to another round of mutation, in a repeated cycle. In a microcosmic version of Darwinian evolution, mutation followed by selection of the fittest took an initially unpromising pool of broken circuits and transformed them into winners. "Ron is utilizing the power of evolution to design networks in ways so that they perform exactly the way you want them to," observed Jim Collins [16]. In a commentary article in the same issue of the journal, Jeff Hasty called this approach "design then mutate" [17]. The team showed how a circuit made up of three genetic gates could be fine-tuned *in vivo* to give the correct performance, and they concluded that "the approach we have outlined should serve as a robust and widely applicable route to obtaining circuits, as well as new genetic devices, that function inside living cells."

Pattern Formation

The next topic studied by Weiss and his team was the problem of *space* – specifically, how to get a population of bacteria to cover a surface with a specific density. This facility could be useful when designing bacterial bio-sensors – devices that detect chemicals in the environment and produce a response. By controlling the density of the microbial components, it might be possible to tune the sensitivity of the overall device. More importantly, the ability for cells to control their own density would provide a useful "self-destruct" mechanism were these genetically-modified bugs ever to be released into the environment for "real world" applications.

In "Programmed population control by cell-cell communication and regulated killing" [40], Weiss and his team built on their previous results to demonstrate the ability to keep the density of an *E. coli* population artificially low – that is, below the "natural" density that could be supported by the available nutrients. They designed a genetic circuit that caused the bacteria to generate a different *Vibrio* signalling molecule, only this time, instead of making the cells glow, a sufficient concentration would flip a switch inside the cell, turning on a *killer gene*, encoding a protein that was toxic in sufficient quantities. The system behaved exactly as predicted by their mathematical model. The culture grew at an exponential rate (that is, doubling every time step) for seven hours, before hitting the defined density threshold. At that point the population dropped sharply, as countless cells expired, until the population settled at a steady density significantly (ten times) lower than an unmodified "control" colony. The team concluded that "The population-control circuit lays the foundations for using cell-cell communication to programme interactions among bacterial colonies, allowing the concept of communication -regulated growth and death to be extended to engineering synthetic ecosystems" [40].

The next stage was to programme cells to form *specific* spatial patterns in the dish. As we have already mentioned briefly, pattern formation is one characteristic of multicellular systems. This is generally achieved using some form of chemical signalling, combined with a differential response – that is, different cells, although genetically identical, may "read" the environmental signals and react in different ways, depending on their internal state. For example, one cell might be "hungry" and choose to move towards a food source, while an identical cell might choose to remain in the same spot, since it has adequate levels of energy.

The team used a variant of the sender-receiver model, only this time adding a "distance detection" component to the receiver circuit. The senders were placed in the centre of the dish, and the receivers distributed uniformly across the surface. The receivers constructed so that they could measure the *strength* of the signal being "beamed" from the senders, a signal which decayed over distance (a little like a radio station gradually breaking up as you move out of the reception area). The cells were engineered so that only those that were either "near" to the senders or "far" from the senders would generate a response (those in the middle region were instructed to keep quiet). These cells are genetically identical, and are uniformly distributed over the surface – the differential response comes in the way that they assess the strength of the signal, and make a decision on whether or not to respond. The power of the system was increased further by making the near cells glow green, and those far away glow red (using a different fluorescent protein).

When the team set the system running, they observed the formation of a "dartboard" pattern, with the "bullseye" being the colony of senders (instructed to glow cyan, or light blue), which was surrounded by a green ring, which in turn was surrounded by a red ring. By placing three sender colonies in a triangle, they were also able to obtain a green heart-shaped pattern, formed by the intersection of three green circles, as well as other patterns, determined solely by the initial configuration of senders [4].

Bacterial Camera

Rather than generating light, a different team decided to use bacteria to *detect* light – in the process, building the world's first microbial camera. By engineering a dense bed of *E. coli*, a team of students led by Chris Voight at Berkeley developed light-sensitive "film" capable of storing images at a resolution of 100 megapixels per square inch. *E. coli* are not normally sensitive to light, so the group took genes coding for *photoreceptors* from blue-green algae, and spliced them into their bugs [24]. When light was shone on the cells, it turned on a genetic switch that cause a chemical inside them to permanently darken, thus generating a black “pixel”. By projecting an image onto a plate of bacteria, the team were able to obtain several monochrome images, including the *Nature* logo and the face of team member Andrew Ellington. Nobel Laureate Sir Harry Kroto, discoverer of “buckballs”, called the team’s camera an “extremely exciting advance” [26], going on to say that “I have always thought that the first major nanotechnology advances would involve some sort of chemical modification of biology.”

VII Future Directions

Weiss and his team team suggest that “the integration of such systems into higher-level organisms and with different cell functions will have practical applications in three-dimensional tissue engineering, biosensing, and biomaterial fabrication.” [4] One possible use for such a system might lie in the detection of bio-weapons – spread a culture of bacteria over a surface, and, with the appropriate control circuit, they will be able to accurately pinpoint the location of any pathogens. Programmed cells could eventually replace artificial tissues, or even organs – current attempts to build such constructions in the laboratory rely on cells arranging themselves around an artificial scaffold. Controlled cellular structure formation could do away with the need for such support – “The way we’re doing tissue engineering, right now, ... is very unnatural,” argues Weiss. “Clearly cells make scaffolds themselves. If we’re able to program them to do that, we might be able to embed them in the site of injury and have them figure out for themselves what the pattern should be” [7]. In addition to building structures, others are considering engineering cells to act as miniature drug delivery systems – fighting disease or infection from the *inside*. Adam Arkin and Chris Voight are currently investigating the use of modified it E. coli to battle against cancer tumours, while Jay Keasling and co-workers at Berkeley are looking at engineering circuits into the same bacteria to persuade them to generate a potent antimalarial drug that is normally found in small amounts in wormwood plants.

Clearly, bacterial computing/synthetic biology is still at a relatively early stage in its development, although the field is growing at a tremendous pace. It could be argued, with some justification, that the dominant science of the new millennium may well prove to be at the intersection of biology and computing. As biologist Roger Brent argues, “I think that synthetic biology...will be as important to the 21st century as [the] ability to manipulate bits was to the 20th.” [1]

Primary Literature

[1] Anon. Roger Brent and the Alpha project. *ACM Ubiquity*, 5(3), 2004.

[2] A. Arkin and J. Ross. Computational functions in biochemical reaction networks. *Biophysical Journal*, 67:560–578, 1994.

[3] Naama Barkai and Stanislas Leibler. Circadian clocks limited by noise. *Nature*, 403:267–268, 2000.

[4] Subhayu Basu, Yoram Gerchman, Cynthia H. Collins, Frances H. Arnold, and Ron Weiss. A synthetic multicellular system for programmed pattern formation. *Nature*, 434:1130–1134, 2005.

[5] Steven A. Benner and Michael Sismour. Synthetic biology. *Nature Reviews Genetics*, 6:533–543, 2005.

[6] Chappell Brown. BioBricks to help reverse-engineer life. *EE Times*, June 11, 2004.

[7] Susan Brown. Command performances. *San Diego Union-Tribune*, December 14, 2005.

[8] T.A. Brown. *Gene Cloning: an Introduction.* Chapman and Hall, second edition, 1990.

[9] Francis Crick. Central dogma of molecular biology. *Nature*, 227:561–563, 1970.

[10] Anne Eisenberg. Unlike viruses, bacteria find a welcome in the world of computing. *New York Times*, June 1, 2000.

[11] M. Elowitz and S. Leibler. A synthetic oscillatory network of transcriptional regulators. *Nature*, 403:335–338, January 2000.

[12] Dan Ferber. Synthetic biology: Microbes made to order. *Science*, 303(5655):158–161, 2004.

[13] T. Gardner, R. Cantor, and J. Collins. Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 403:339–342, January 2000.

[14] C.R. Geyer, T.R. Battersby, and S.A. Benner. Nucleobase pairing in expanded Watson-Crick-like genetic information systems. *Structure*, 11:1485–1498, 2003.

[15] W. Wayt Gibbs. Synthetic life. *Scientific American*, April 26, 2004.

[16] Lauren Gravitz. 10 emerging technologies that will change your world. *MIT Technology Review*, February 2004.

[17] Jeff Hasty. Design then mutate. *Proc. Natl. Acad. Sci. USA (PNAS)*, 99(26):16516–16518, 2002.

[18] Karen Hopkin. Life: The next generation. *The Scientist*, 18(19):56, October 11, 2004.

[19] D.A. Jackson, R.H. Symons, and P. Berg. Biochemical method for inserting new genetic information into DNA of simian virus 40: circular SV40 DNA molecules containing lambda phage genes and the galactose operon of *Escherichia coli*. *Proc. Natl. Acad. Sci. USA*, 69:2904–2909, 1972.

[20] F. Jacob and J. Monod. Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 3:318–356, 1961.

[21] Alok Jha. From the cells up. *The Guardian*, March 10, 2005.

[22] Stuart Kauffman. Gene regulation networks: a theory for their global structure and behaviors. *Current topics in developmental biology*, 6:145–182, 1971.

[23] Stuart A. Kauffman. *The origins of order: Self-organization and selection in evolution.* Oxford University Press, 1993.

[24] Anselm Levskaya, Aaron A. Chevalier, Jeffrey J. Tabor, Zachary Booth Simpson, Laura A. Lavery, Matthew Levy, Eric A. Davidson, Alexander Scouras, Andrew D. Ellington, Edward M. Marcotte, and Christopher A. Voight. Engineering *Escherichia coli* to see light. *Nature*, 438:441–442, 2005.

[25] P.E. Lobban and C.A. Sutton. Enzymatic end-to-end joining of DNA molecules. *J. Mol. Biol.*, pages 453–471, 1973.

[26] Paul Marks. Living camera uses bacteria to capture image. *New Scientist*, November 23, 2005.

[27] Harley H. McAdams and Lucy Shapiro. Circuit simulation of genetic networks. *Science*, 269(5224):650–656, 1995.

[28] H.H. McAdams and A. Arkin. Genetic regulatory circuits: Advances toward a genetic circuit engineering discipline. *Current Biology*, 10:318–320, 2000.

[29] Jacques Monod. *Chance and Necessity.* Penguin, 1970.

[30] Jacques Monod, Jean-Pierre Changeux, and Francois Jacob. Allosteric proteins and cellular control systems. *Journal of Molecular Biology*, 6:306–329, 1963.

[31] Oliver Morton. Life, Reinvented. *Wired*, 13.01, January 2005.

[32] Registry of Standard Biological Parts. http://parts.mit.edu/.

[33] R. Old and S. Primrose. *Principles of Gene Manipulation, an Introduction to Genetic Engineering.* Blackwell, fifth edition, 1994.

[34] Mark Ptashne. *A Genetic Switch.* Cell Press and Blackwell Scientific, 1987.

[35] Lynne Roberts and Colin Murrell (Eds.). An introduction to genetic engineering. Department of Biological Sciences, University of Warwick, 1998.

[36] Steven Strogatz. *Sync: The Emerging Science of Spontaneous Order.* Penguin, 2003.

[37] A.M. Turing. The chemical basis of morphogenesis. *Philosophical Transactions of the Royal Society B (London)*, 237:37–72, 1952.

[38] John von Neumann. The general and logical theory of automata. In *Cerebral Mechanisms in Behavior*, pages 1–41. Wiley, New York, 1941.

[39] Yohei Yokobayashi, Ron Weiss, and Frances H. Arnold. Directed evolution of a genetic circuit. *Proc. Natl. Acad. Sci. USA (PNAS)*, 99(26):16587–16591, 2002.

[40] Lingchong You, Robert Sidney Cox III, Ron Weiss, and Frances H. Arnold. Programmed population control by cell-cell communication and regulated killing. *Nature*, 428:868–871, 2004.

Books and Reviews

[41] Uri Alon. *An Introduction to Systems Biology: Design Principles of Biological Circuits.* Chapman and Hall/CRC, 2006.

[42] Martyn Amos, editor. *Cellular Computing.* Series in Systems Biology. Oxford University Press, USA, 2004.

[43] Martyn Amos. *Genesis Machines: The New Science of Biocomputing.* Atlantic Books, 2006.

[44] Steven A. Benner. Synthetic biology: Act natural. *Nature*, 421:118, 2003.

[45] Drew Endy. Foundations for engineering biology. *Nature*, 436:449–453, 2005.

[46] Hideki Kobayashi, Mads Kaern, Michihiro Araki, Kristy Chung, Timothy S. Gardner, Charles R. Cantor, and James J. Collins. Programmable cells: interfacing natural and engineered gene networks. *Proc. Nat. Acad. Sci.*, 101(22):8414–8419, 2004.

[47] Gary S. Sayler, Michael L. Simpson, and Chris D. Cox. Emerging foundations: nano-engineering and bio-microelectronics for environmental biotechnology. *Current Opinion in Microbiology*, 7:267–273, 2004.