

# Artificial Life



CHRISTOPHER M. BISHOP  
Darwin College, Cambridge, U.K.  
[Christopher.Bishop@microsoft.com](mailto:Christopher.Bishop@microsoft.com)

To appear in *Life*, edited by William Brown and Andrew C. Fabian  
Cambridge University Press (2014)

Living organisms are extraordinary. They have capabilities which far exceed any present-day technology, and it is therefore inevitable that scientists and engineers should seek to emulate at least some of those capabilities in artificial systems. Such an endeavour not only offers the possibility of practical applications, but it also sheds light on the nature of biological systems.

The notion of artificial life can take many diverse forms, and in this article we will focus on three aspects: modelling the development of structure in living systems, the quest to create artificial intelligence, and the emerging field of synthetic biology. All three topics reveal surprising, and sometimes remarkably deep, connections between the apparently disparate disciplines of biology and computer science. There is something else which links these three strands: the Cambridge mathematician Alan Turing (see Figure 1) whose birth centennial we celebrate this year.



**Figure 1: Alan Turing (1912 – 1954).**

It is widely acknowledged that Turing laid many of the foundations for the field of computer science, although amongst the general public he is perhaps best known for his role in breaking the Enigma and other cyphers at Bletchley Park during the Second World War [Hodges, 1992]. What is perhaps less widely appreciated is

that Turing also made important contributions to biology. As we shall see, each of the three topics discussed in this paper builds on a different seminal contribution made by Turing.

## Morphogenesis

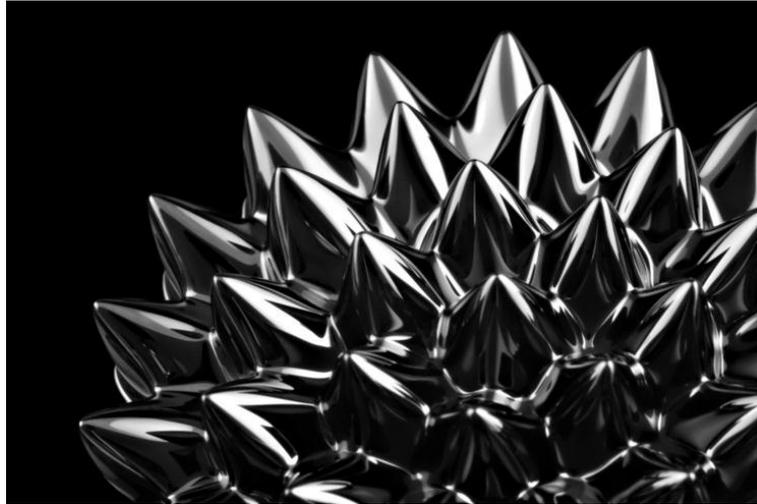
One of the most intriguing challenges in biology is to understand the mechanisms by which an organism acquires structure and form during its development. This is known as *morphogenesis* (the “creation of shape”). A visually striking, and very familiar, example of structure in a living organism is given by the spots on a leopard (Figure 2).



Figure 2: Leopard with spots.

The fact that a leopard has spots rather than, say, stripes, is determined genetically. However, the precise size, shape, and location of individual spots do not seem to be genetically controlled but, instead, emerge through a mechanism for creating “spottiness”. One of the earliest attempts to elucidate such a mechanism was given by Alan Turing. In 1952, just two years before his untimely death, he proposed a chemical basis for morphogenesis, together with a corresponding mathematical analysis [Turing, 1952]. His idea was that spatial structures, such as the spots on the leopard, arise from the interaction between specific chemical and physical processes in an example of spontaneous pattern formation, also known as *self-organisation*.

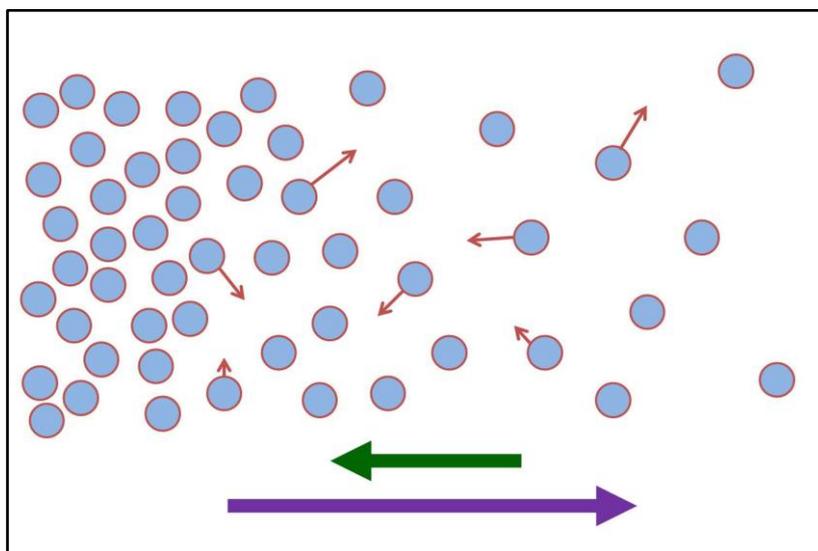
There are many examples of self-organising systems. For example, when a magnetic field is applied to a ferrofluid, a regular array of spikes is formed, as illustrated in Figure 3.



**Figure 3: Spikes in a ferrofluid resulting from the application of a magnetic field.**

The ferrofluid consists of a colloidal suspension of nanometre-scale ferromagnetic particles dispersed in a liquid. In the absence of a magnetic field, the ferrofluid behaves like a regular, slightly viscous liquid. When it is placed into a magnetic field, however, the ferrofluid organises itself into regular spikes, the size and spacing of which depend on the strength of the magnetic field. This period structure arises from the interaction of the ferrofluid with the magnetic field, as neither the fluid nor the magnetic field individually has any intrinsic structure of this kind. The external field magnetises the fluid, which then forms into spikes as this leads to a lowering of the magnetic energy. This phenomenon of energy minimisation is familiar from the behaviour of two bar magnets, which, if allowed to move freely, will align with their poles pointing in opposite directions and will then come together. The ferrofluid likewise adopts a configuration that minimizes the overall energy of the system, with the spikes growing until the reduction in magnetic energy is balanced by the increase in gravitational and surface tension energies. As long as the magnetic field is sufficiently strong, the formation of spikes is energetically favourable.

Clearly a different process must be responsible for pattern formation in biological systems. The specific mechanism proposed by Turing is known as a *reaction-diffusion* system. Here ‘reaction’ refers to chemical reactions, while the concept of diffusion can be explained using the illustration in Figure 4.



**Figure 4: The mechanism of diffusion. See text for details.**

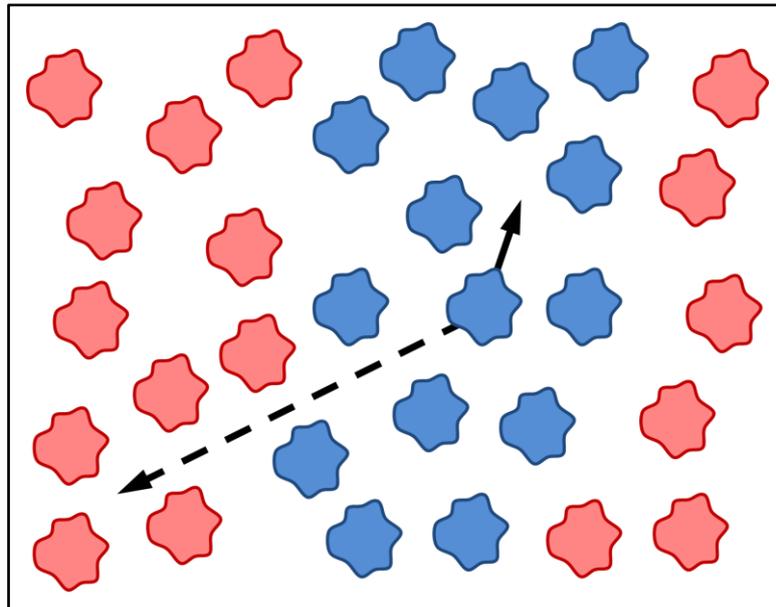
This diagram depicts a molecular-level view of a situation in which there is a spatial gradient in the concentration of some substance. In this case the concentration gradient is from left to right. The molecules themselves are in constant, random motion as a result of being at a non-zero temperature (**heat is just the random motion of molecules**) and this is illustrated by the arrows attached to some of the molecules. Each molecule is equally likely to move to the left as to the right. However, there are more molecules on the left of the picture than on the right, so on average there will be a greater flow of molecules from left to right compared to the flow from right to left. **Overall, therefore, there is a net flow of molecules from left to right, and over time this will act to reduce the concentration gradient.** This effect is easily demonstrated experimentally by carefully adding a few drops of **food colouring** to a beaker of water, without stirring. Over the subsequent hours and days the **colouring** will gradually spread through the water until the **colour** of the solution is uniform. As the gradient approaches zero so too will the net flow of molecules. This statistical property whereby on average there is a flow of a substance in the opposite direction to a concentration gradient, **as a result of the random thermal motion of the molecules, is called diffusion.**

Now consider a situation in which there are two or more species of molecule, each having gradients in concentration, in which the species can undergo chemical reactions with each other. Because chemical reactions transform molecules of one type into molecules of other types they effectively change the concentration and hence the concentration gradients. Conversely, the rate of chemical reaction depends on the concentrations of the reacting species. **The result is a complex interaction between reaction and diffusion.**

Turing analysed a particular class of reaction-diffusion systems mathematically and discovered that, under appropriate conditions, such systems can exhibit a wide variety of behaviours, but that they always converge to one of six stable states [Kondo *et al.*, 2010]. The first of these involves concentrations that are uniform in space and constant in time, and are therefore relatively uninteresting. The second consists of solutions which are uniform in space but which **oscillate** through time, as seen in circadian rhythms and the contraction of heart muscle cells. The third and fourth classes of solutions consists of 'salt and pepper' patterns of high spatial **frequency** which are either constant (third class) or which **oscillate** through time (fourth class). Of these, the former are seen in neuro-progenitor cells in the epithelium of *Drosophila* embryos, while no examples the latter have yet been identified in living organisms. The fifth class exhibits travelling waves having spatial structure which evolve through time. Beautiful demonstrations of such solutions can be performed in the laboratory using simple chemistry, while biological examples include the spiral patterns formed by clusters of the amoeba *Dictyostelium discoideum*. The sixth class of solutions consist of spatial structures that are constant in time, and have become known as *Turing patterns*. These are remarkable solutions since the patterns are stable, and can even regenerate following an external disturbance. They represent a **dynamic equilibrium** in which the effects of **reaction and diffusion are balanced**, leading to a stationary structure. Such structures are **self-organising** and do not require any pre-existing spatial information. There is now good evidence to support the role of this mechanism in creating a variety of biological patterns including complex patterns of seashells, the patterning of bird feathers, and the impressive diversity of vertebrate skin patterns.

We don't have to have a literal reaction-diffusion system for Turing's mechanism to operate. More recently it has become clear that the conditions for Turing pattern formation are more general, and require only the presence of two competing effects having rather general properties [Kondo *et al.*, 2010]. One effect acts over a short range and encourages regions to be the same (in terms of concentration, **colour**, or some other characteristic) while the other effect acts over a longer range and encourages regions to be different. This

can most easily be illustrated using the simple example shown in Figure 5, which considers the specific example of **cells which can communicate via chemical signalling**.



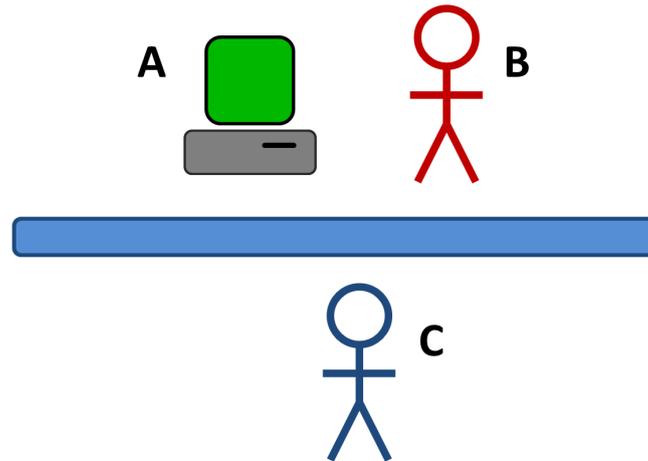
**Figure 5. General mechanism for the formation of Turing patterns.**

We shall suppose that each cell can take one of two **colours**, red or blue. Each cell can also send two chemical **signals** which can be detected by other cells. One of these **signals** (indicated for a particular cell by the solid arrow in Figure 5) acts over a short range and encourages other cells to take on the same **colour** as the transmitting cell. The second **signal** (indicated by the dashed arrow in Figure 5) acts over a longer range and has the converse effect, encouraging other cells to have the opposite **colour** to the transmitting cell. Initially, the cells have randomly assigned **colours**, but the effects of the interactions due to the chemical **signalling** is to cause the ensemble of cells to reach a stable configuration. If the parameters of the **signalling** are suitable, then this stable configuration will exhibit a Turing pattern, such as the strips indicated in the figure.

We shall return to a discussion of Turing patterns later, but for the moment we consider another of the great intellectual challenges tackled by Turing.

## Artificial intelligence

Visions of intelligent machines go back to antiquity, but it is with the development of programmable digital computers that the realistic possibility arose of building such machines. Alan Turing's seminal work on the theory of computation laid the foundations for modern digital computers, and Turing himself was also fascinated by the challenge of building intelligence machines. One obvious problem is that there is no simple definition of what it means to be intelligent, and therefore no simple criterion for success. Turing addressed this by introducing an operational procedure for determining whether a machine is intelligent [Turing, 1950]. This has become known as the *Turing test*, and is illustrated in Figure 6.



**Figure 6. Schematic illustration of the ‘Turing test’.**

The essential idea is to compare the artificial system with a human on a series of tests, and if it proves impossible to distinguish which is which, then the machine is deemed to be intelligent. This is achieved by separating the interrogator C from the computer A and the comparison human B (Figure 6). The interrogator can communicate with A and B only through a text-based system such as a keyboard and screen, and initially does not know which of A and B is the computer and which is the human. This use of text as a medium for communication was chosen to bypass issues with the computer having to interpret and synthesise audio speech. The interrogator can ask questions of both A and B and, based on the answers, must decide which is human and which is the machine. If the interrogator is unable to distinguish them then, according to the Turing test, the machine is deemed to be intelligent.

The Turing test has generated intense discussion and controversy over the years. While it has the benefit of being an operational definition, it also has some clear limitations. In particular, there are many other forms of “intelligence” exhibited by living systems which are not captured by the test. For example, a chimpanzee could never pass the Turing test because it cannot read and write text, but it has numerous capabilities (recognising objects, locomotion, fine motor control, planning, use of tools, and many others) that it could be very beneficial to emulate in machines. Moreover, from a technological perspective, imitation of humans is, of itself, only of limited interest. If the interrogator asked a computer to solve a complex arithmetical problem and the answer came back correctly a split second later it would be clear that this was a machine not a human, and so the computer would have failed the Turing test, even though it was outperforming the human! It has been said that “aeroplanes are measured by how well they fly, not by how accurately they mimic birds”, and so it is with machines.

Artificial intelligence became a popular field of research from the mid-1950s, with much of the effort focussed on techniques based on hand-crafted solutions, often involving logical processes which manipulate symbolic representations of the world. A specific approach, which became popular in the 1970s, was based on rules derived from a human expert through a process called *knowledge elicitation*. The resulting artificial instantiation of those rules was known as an *expert system*. To give a concrete example, suppose we wish to build a system that can diagnose a human disease, given a list of the patient’s symptoms. The computer might be fed with hand-crafted rules of the form “if the patient has a high temperature and low blood pressure then ...” in which the rules are obtained from human experts. For a specific application domain, a set of rules would be compiled, known as a *knowledge base*, and then a piece of software called an *inference engine* would invoke appropriate rules in answering a specific query.

Although this approach of hand-crafting intelligent behaviour has had some worthwhile successes in niche applications, it has broadly failed to deliver a general framework for creating machine intelligence. The fundamental problem seems to be that, except in very specific domains, the real world cannot effectively be captured by a compact set of human-expressible rules. Attempts to capture exceptions to rules using additional rules, leads to a combinatoric explosion of exceptions-to-exceptions.

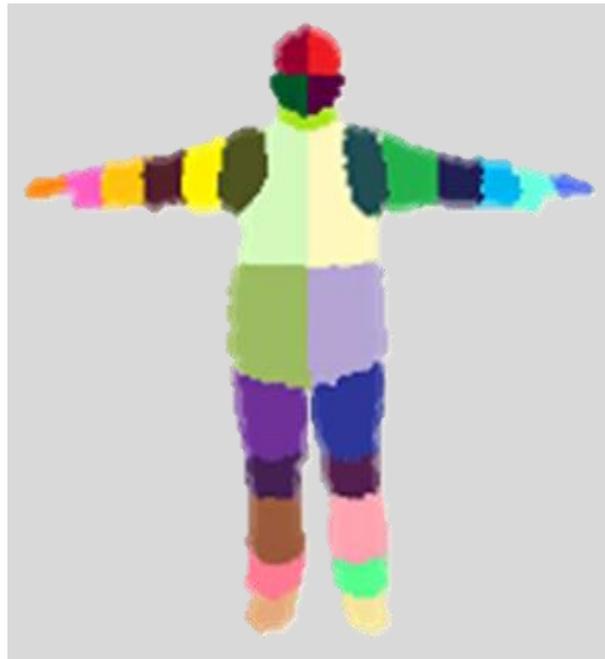
Since the mid-1980s the most prevalent, and most successful, approach to creating intelligence in machines is based on a very different paradigm known as *machine learning* [Bishop, 2006]. Instead of programming the computer directly to exhibit a particular form of intelligent behaviour, the computer is programmed to be *adaptive* so that its performance can improve as a result of experience.

In the case of a medical diagnosis system, the 'experience' might consist of a data base of examples, each of which comprises a list of observed symptoms along with a diagnosis of the disease provided by a human expert or from some other form of ground truth such as a blood test. Such a data base is known as a *training set*. A simple form of machine learning technique in this case could consist of a non-linear mathematical function that maps symptoms as inputs to diseases as output. This function is governed by some adjustable parameters, and the computer has an algorithm for optimising these parameters so as to make the accurate prediction of the corresponding disease for each set of symptoms. One well-known class of such non-linear functions, that was very prevalent in the 1990s, is called a *neural network*. Today there are hundreds, if not thousands, of different machine learning techniques.

It is worth noting that such systems can easily be made to achieve good predictions on the training data examples, but that the real goal of the system is to be able to make good predictions for new examples that are not contained in the training set. This capability is called *generalization*. Essentially it requires that there be some underlying regularity in the data, so that the system can learn the regularity rather than the specifics of each instance, and hence can generalise effectively.

Techniques of this kind have achieved widespread success in many different application domains. A recent example is the Kinect® full-body tracking system used on the Xbox® games console, which launched in November 2010 and which set a new Guinness world record as the fastest-selling consumer electronics device of all time. This is the first piece of consumer electronics to provide real-time tracking of the entire human body, and has been deployed initially as a hands-free games controller. It is based on a special type of infra-red camera that measures depth rather than intensity, so that each pixel in each frame of the captured video represents the distance from the sensor to the corresponding point in the scene. Unlike a conventional camera, this sensor thereby has a 3-dimensional view of the world, which eases the task of tracking the human body in 3D space.

The fundamental problem that needs to be solved in order to use the sensor to track the human body is the ability to recognise the various parts of the body in real time using the depth data. This was solved using a machine learning approach as follows. First, 31 regions of the body are defined, as illustrated in Figure 7.



**Figure 7. The 31 body parts to be recognised by Kinect®.**

Then in each frame of the video, the games console must take the depth image and, for each pixel in the image, classify that pixel according to which of the 31 body regions it corresponds to (or whether it belongs to the background). This is solved using a machine learning technique called *random forests of decision trees* [Shotton *et al.*, 2011]. In order to train the machine learning system, a training set was created consisting of one million body poses, each with a 3D depth image and each having every pixel labelled with the correct body region. In order to ensure that the training set contains a representative range of body positions, the poses were collected using a Hollywood style motion capture suite. An actor wears a special suit which has high-visibility markers attached at key points, and these are viewed by multiple conventional cameras placed in different positions looking at the actor. By correlating the data from the cameras the 3D body pose can be inferred. These body poses are then used to generate a synthetic training set which accounts for variations in body size and shape, and clothing, as well as artefacts introduced by the depth camera itself.

The one million examples in the training set are then used to tune the parameters of the random forest of decision trees. This is done using a large array of PCs, and is a computationally very intensive process. Once the system has been successfully trained, however, it can be deployed on the games console where it can classify new pixels very quickly. In fact it can classify all of the relevant pixels in each frame of the video (at some 30 frames per second) in real time while using less than 10% of the processing power of the Xbox console. Although Kinect was originally conceived for use as a games controller, it is being tested in a variety of user-interface applications, for example in operating theatres to allow surgeons to manipulate digital images of the patient without contact with physical mice or keyboards that would compromise sterility.

Another impressive example of the power of machine learning took the form of the Grand Challenge organised by the U.S. Defence Advanced Research Projects Agency (DARPA). The goal was for a fully autonomous vehicle to drive a distance of 240 km along a pre-determined path through the Mojave Desert, with a prize of \$1M. The competition first took place in March 2004, and none of the vehicles managed to complete the course, with the most successful vehicle managing to cover less than 12 km, and the prize

was not awarded. The following year, the competition was repeated, this time with a prize of \$2M and an even more challenging course. Five vehicles managed to complete the course, with the winning team from Stanford receiving the prize. Their algorithms used machine learning to solve the key problem of obstacle avoidance, and was based on learning from a log of human responses and decisions under similar driving conditions.

Following the success of the 2005 race, DARPA held the *Urban Challenge* in 2007, involving a 96 km urban route, to be completed in under 6 hours, with a total prize of \$3.5M. Vehicles had to obey all traffic signals and road markings while also dealing with the other robotic vehicles on the course. The winning entry was from Carnegie Mellon University, completing the course in just over 4 hours. Autonomous vehicle technology for cars to replace human drivers offers the promise of reduced numbers of accidents and fatalities, increased efficiency, and the opportunity for drivers to make better use of their time (e.g. to work, sleep or watch a film) while the car drives itself.

DARPA recently announced a new *Robotics Challenge* which will involve the solution of complex disaster-relief tasks by humanoid robots in situations considered too dangerous for humans to operate. Unlike previous challenges, the teams will be provided with identical humanoid robots, and so the focus will be on the artificial intelligence software for controlling them. The robots will be expected to make use of standard tools and equipment commonly available in human environments, as illustrated conceptually in the DARPA image shown in Figure 8.



**Figure 8. Conceptual image of the DARPA humanoid *Robotics Challenge*, depicting two robots tackling an accident in a chemical plant.**

As our final example of the rapid progress of artificial intelligence through the use of machine learning, we look at the widely publicised participation in 2011 of the IBM computer system *Watson* in the US television quiz programme *Jeopardy!*.

The *Jeopardy!* show was created in 1964, and over 6,000 episodes have been broadcast. Its format is slightly unusual in that the game host poses trivia puzzles in the form of answers to which the contestants must provide the appropriate questions. A special series of three programmes was run to host this contest in which *Watson* played against the two leading human players from previous years. In preparation to play *Jeopardy!*, *Watson* had been fed with 200 million pages of information from encyclopaedias (including the whole of Wikipedia), dictionaries, thesauri, news articles, and other sources, consuming a total of 4

terabytes of disk storage. However, Watson was not connected to the internet during the game. Watson's software is based on a complex system of interacting modules that process natural language, retrieve relevant information retrieval, and represent and reason about knowledge, with machine learning playing a key role. The Watson hardware was capable of processing around 80 trillion computer instructions per second, and was represented on stage by an Avatar placed between the two human contestants, the actual hardware being far too large to fit on stage.

During the game, Watson's three highest-ranked potential responses were displayed on the television screen, but Watson only buzzed when the confidence of the highest ranked answer exceeded a suitable threshold. Watson then provided a response using a synthesised voice. In a combined-point match spread over three episodes, Watson was the comfortable winner and collected the \$1M first prize which was subsequently donated to charity. This head-to-head competition with humans in a natural language setting is somewhat reminiscent of the original Turing Test. It represents a remarkable development in artificial intelligence because natural language has long been seen as a highly challenging domain for computers. Today machine learning has become one of the most active and important frontiers of computer science.

## Synthetic biology

Our third and final viewpoint on artificial life reveals some deep connections between biology and computer science, and again relies on some key insights from Alan Turing. **Research in molecular biology has revealed extraordinarily complex networks of biochemical interactions in living cells.** As the details of such networks have been identified, it has become increasingly clear that much of the functionality of these networks is concerned with information processing. It is not just computer scientists who hold this view. In his recent talk at the Royal Institution, Sir Paul Nurse, Nobel Prize winner and President of the Royal Society, spoke of the *Great Ideas of Biology*, which he listed as the cell, the gene, evolution by natural selection, and biology viewed as complex chemistry. He then suggested that the next great idea of biology would be the recognition that life is **"a system which manages information"**.

The view of molecular biology as information processing is much more than simply a convenient analogy, and to understand the depth of this viewpoint we need to look at a few key concepts in molecular biology and also to turn again to the work of Turing. At the heart of the biomolecular machinery of the living cell is DNA, or deoxyribonucleic acid. This is a molecule with a double helix structure in which each strand of the helix comprises a sequence of bases, each of which is either C (cytosine), A (adenine), G (guanine), or T (thymine). If we imagine unwinding the helices we would obtain a structure of the form shown conceptually in Figure 9.

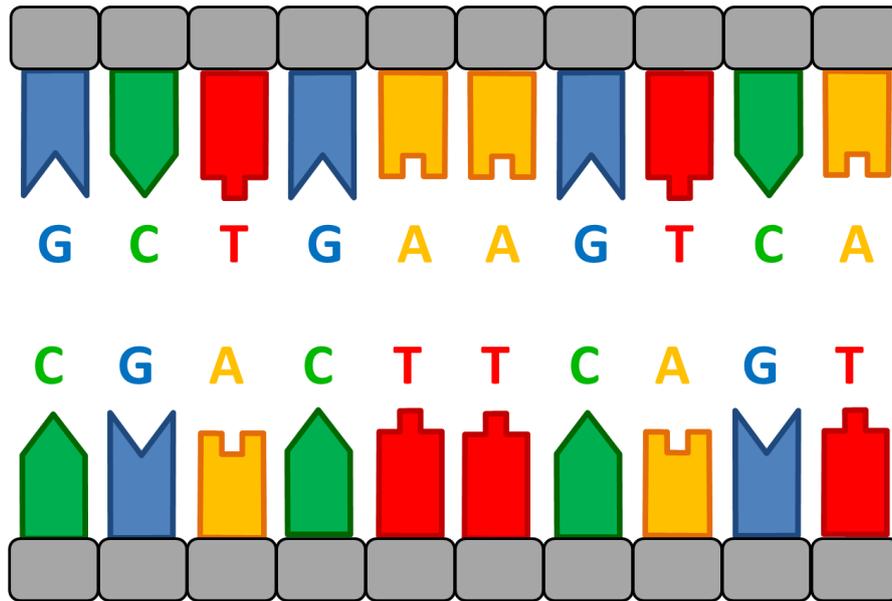


Figure 9. The two strands of DNA showing the complementary sequences of bases.

We can see immediately that DNA is a digital information storage system in which the information is represented by the sequence of bases. Note that the two strands of DNA are complementary, in that a G always occurs opposite a C, and a T always occurs opposite an A. Thus the two strands carry the same information, but in complementary representations. The same information could equally well be represented in the binary representation used by computers, for example by making the (arbitrarily chosen) associations shown in Figure 10.

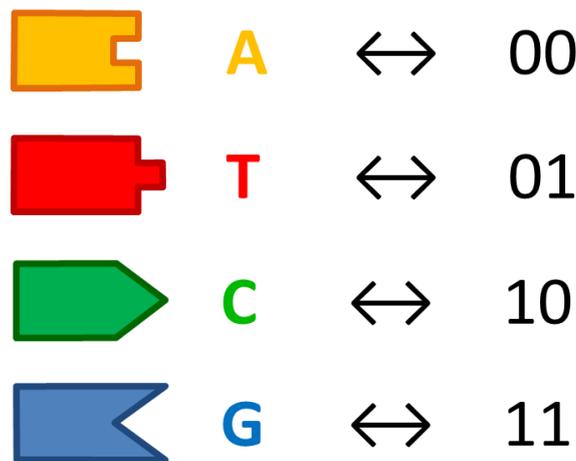


Figure 10. An association between bases and pairs of binary digits.

Thus the sequence of bases GCTGAC would be represented as 111001110010. Only ten base pairs were shown in Figure 9. **The DNA for a human has around 3 billion pairs of bases, and this DNA is contained in almost all of the 100 trillion cells in the human body.** The DNA in each cell therefore stores around 6 billion bits of information, or roughly 750 megabytes, which is roughly equivalent to data stored on an audio CD.

The DNA contains genetic information that is inherited by an organism’s offspring and which control many aspects of the development and function of the organism. Specific section sections of the DNA, called genes, control the production of proteins through a two-stage mechanism in which the sequence

information is first copied to a related type of molecule called *messenger RNA* in a process called *transcription*. The messenger RNA is then used to synthesise proteins by structures called ribosomes in a process called *translation*. Although essentially every cell in an organism contains the same DNA, in a particular cell and at a particular time, only some of the genes are *expressed*, with others essentially dormant. The degree of expression of a particular gene is controlled by an associated section of the DNA called a *regulatory region*. Proteins binding with the regulatory region can alter the degree of gene expression. Here we see the beginnings of a very complex feedback system in which DNA controls the production of proteins which themselves can influence the production of other proteins. We therefore see that the DNA is much more than simply a static information store. It is a key component in a highly complex dynamical system with multiple feedback loops in a process that strongly resembles computation in digital computers.

By modifying the DNA the properties of the organism can be changed. This has been done for thousands of years through selective breeding in order to produce genetically altered organisms having desired characteristics (e.g. a stronger horse or a more nutritious variant of wheat). With the discovery of the molecular basis of inheritance, came the opportunity to do this more specifically and more rapidly, leading to the field of *genetic engineering*. This involves copying a gene from one organism and inserting it into the DNA of another organism, for example, a gene from a bacterium which codes for an insecticidal protein can be inserted into the peanut plant, thereby protecting its leaves from attack by the larvae of the cornstalk borer. When the larvae eat the leaves they are killed by the insecticide which is now being synthesised by the peanut plant.

Since then, significant advances have been made in our ability both to sequence and to synthesise DNA. We are now no longer restricted to cutting and pasting single genes from one organism to another, but can read and write whole genomes. The power of this approach was first demonstrated by Craig Venter and colleagues, who in 2010 took the DNA from a species of bacteria called *Mycoplasma mycoides* and ran it through a machine that measured the sequence of its 1.2M base pairs, reproducing the DNA sequence information in digital form on a computer. Next they inserted various watermarks including quotations from famous people, and an email address. The resulting sequence was then sent to another machine, somewhat analogous to an inkjet printer, but using the four DNA bases rather than four colours, which synthesised the corresponding DNA. They then took a different species of bacteria called *Mycoplasma capricolum* and removed its DNA and replaced it with the synthetic DNA. The resulting organism, nick-named *Synthia*, multiplied and exhibited all the characteristics of the original *Mycoplasma mycoides*. Overall this project took 20 scientists around 10 years at a cost of \$40M.

While this was a remarkable technical achievement, few would regard *Synthia* as truly an artificial life form, as it relied both on an existing host cell with its bimolecular machinery, and on the known DNA sequence of the original organism. It is somewhat analogous to deleting the operating system on a computer and replacing it with the operating system from a different brand of computer, which can be done with very little understanding of how the operating system actually works. A much more exciting, and substantially more challenging, goal is to produce modified organisms by redesigning parts of the DNA sequence rather than simply by copying genes from one organism to another. To see how we might achieve such a goal, it is useful to look more closely at the nature of information processing and computation.

We are familiar with a wide variety of computational devices, from cell phones to desktop PCs, and from pocket calculators to super-computers. Clearly these machines differ in speed, physical size, cost, and other attributes, but do they also differ in the kinds of computational problems which they can solve? The answer

to this question was given by Alan Turing and is rather surprising. Turing first showed that there are computational problems which no computer will ever be able to solve. As an example he described the “halting problem” in which a machine is presented with an example of a computer program and must decide if that program will halt after a finite number of steps or if it will run forever. For many programs this is easy to solve, but Turing showed that no machine can exist which can provide the correct answer for all possible programs. Thus, there is a limit to the computational capability of computers! Turing then went on to show that there can exist a computer which can reach this limit, that is one which can solve any problem which is computationally solvable. This is known as *Turing universality*, and it follows that the set of problems which can be solved by one universal computer is exactly the same as the set which can be solved by any other universal computer. There is a technical caveat that the computers each have access to unlimited amounts of data storage (i.e. memory) otherwise the limit on available memory could further limit the range of computational tasks which the computer could solve. In order to achieve Turing universality a computer must have at least a minimum degree of complexity. A simple pocket calculator with a fixed set of functions, which cannot be programmed to alter its behaviour, is not a universal Turing computer. However, cell phones, PCs, supercomputers, and indeed most other computational devices in everyday use are Turing universal. Even the humble chip inside a chip-and-pin credit card is a universal computer, albeit a relatively slow one.

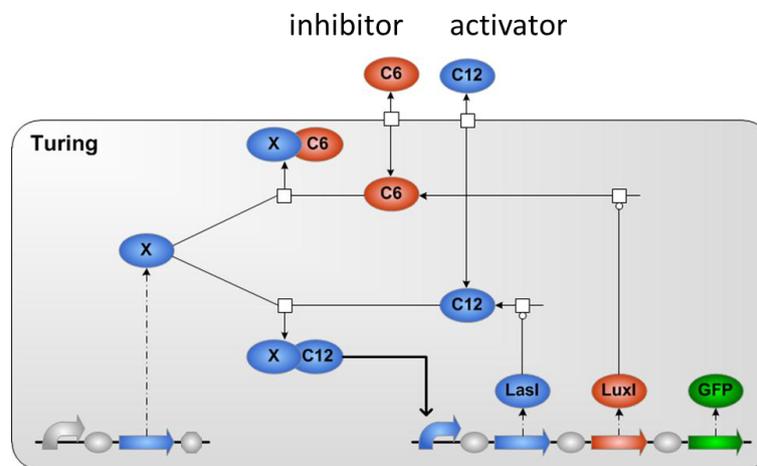
It has been proven formally that the information processing capabilities of the biomolecular machinery in living cells is Turing complete [Cardelli, 2010]. Indeed various sub-systems are themselves Turing complete computational mechanisms. Of course there are many differences between biological and silicon computation. Biological computers can store a gigabyte of information in less than a millionth of a cubic millimetre, they can perform huge numbers of computations in parallel at the same time, they are robust to failure, they are self-repairing, they are very efficient in their use of **energy**, and they can reproduce themselves. So our silicon computers have a long way to go. However, when it comes to the raw ability to do number crunching they are way ahead of biology, and your laptop is not going to be replaced by a blob of green slime any time soon.

The realisation that molecular biology is performing universal computation highlights some deep connections between biology and computer science, and is reflected in the new cross-disciplinary field of research called *synthetic biology* [RAEng, 2009]. The goal is to use the insights and techniques of engineering and computer science to understand the extraordinary complexity of living systems, and thereby allow living cells to be reprogrammed in order to modify their properties.

Synthetic biology is not just the preserve of professional scientists with multi-million dollar research laboratories. There is an increasing movement of amateur groups, and even high school students, experimenting with synthetic biology. Short sequences of DNA can be ordered over the web, simply by typing in the base-pair sequences into a web browser (along with credit card payment) and samples of the corresponding DNA arrive by post a few days later. Currently the cost is around 30p per base pair, and falling with time. Note that such sites only deliver to bona fide users, and sequences are checked first for known pathogens to prevent nefarious activities. What if you don't know which sequences to use? Well, in 2003 Tom Knight at MIT introduced *Biobricks*, a standard registry of biological components (each consisting of a DNA sequence). The registry currently lists around 5,000 items. There is an annual competition called iGEM (international Genetically Engineered Machine) for undergraduate teams around the world. An example iGEM entry is called *E-chromi*, and is a variant of the *E-coli* bacterium which has been modified to produce one of five different **colours** in the presence of an input **signal** such as an environmental toxin.

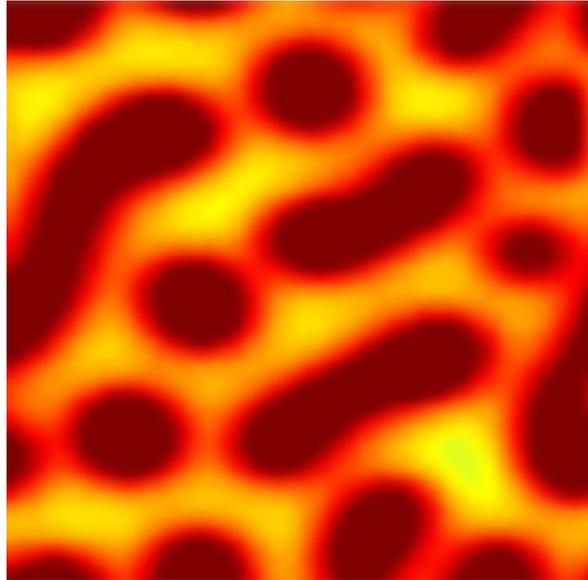
On a larger scale, Jay Keasling from UC Berkeley, with support from the Gates Foundation, has created a yeast strain, using 10 genes from 3 organisms, that is able to produce the anti-malarial drug artemisinin. This is an effective anti-malarial treatment that is traditionally extracted from plants and is very expensive. The new yeast could reduce the production cost by a factor of 10. Keasling also co-founded Amyris, which uses a similar approach to engineer yeast for making diesel fuel and high-grade chemicals.

Much of the work to date has focussed on the use of existing genes having known functionality. To realise the full potential of synthetic biology, it will be essential to work in the reverse direction: for a given desired property of an organism, what is the corresponding DNA sequence that will give rise to that property? It is here that computer science has much to offer, and as an example we return to the topic that we started with: Turing patterns. A collaboration between Microsoft Research and the Department of Plant Sciences at Cambridge University aims to produce a variant of the bacterium *E-coli* which will form colonies that exhibit Turing patterns. Figure 11 shows a simple biomolecular circuit that should be capable of achieving this [Service, 2011].



**Figure 11. Biomolecular circuit for the production of Turing patterns in *E-coli* bacteria.**

The cell can emit and also sense two chemical messengers, one acting as a short-range activator and the other as a long-range inhibitor, to implement the mechanism discussed earlier. The resulting balance of activation and inhibition determines the level of production of a protein called GFP (green fluorescent protein) which can be detected by shining ultraviolet light on the cell. A description of the desired mechanism is then translated into computer code and fed into a piece of software called GEC ('genetic engineering of cells') [Pedersen and Phillips, 2009] which designs a biomolecular system having the desired behaviour, drawing on a database of DNA components. A computer simulation of the behaviour of a colony of bacteria equipped with the corresponding DNA is shown in Figure 12.



**Figure 12. Simulation of cell development showing the emergence of a Turing pattern.**

Although still in its infancy, the field of synthetic biology has tremendous potential. Today we live in an unsustainable world in which the technologies we use to feed and clothe ourselves, and to provide materials for modern life, involve the one-way consumption of finite resources. We urgently need to move to a sustainable world based on renewable resources, and for the most part that will mean biologically-based resources. Our ability to reprogram biology, with the help of tools and insights from computer science and engineering, will be hugely valuable in that endeavour.

### **Acknowledgements**

I am very grateful to many colleagues for their assistance in preparing this lecture, including Luca Cardelli, Andrew Phillips, and Matthew Smith.

## References

Bishop C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Cardelli L., and G Zavattaro. (2010). Turing Universality of the Biochemical Ground Form. *Mathematical Structures in Computer Science*, **20** (1) 45–73 .

Hodges, A. (1992). *Alan Turing: the Enigma*. Vintage publishing.

Kondo S., and Miura T. (2010). Reaction-Diffusion Model as a Framework for Understanding Biological Pattern Formation. *Science*, **329**, 1616–1620.

Pedersen, M. & Phillips, A. (2009). Towards programming languages for genetic engineering of living cells. *J R Soc Interface*, **6** Suppl 4, 437–450.

RAEng (2009). *Synthetic Biology: Scope, Applications, and Implications*. Royal Academy of Engineering. [www.raeng.org.uk/synbio](http://www.raeng.org.uk/synbio)

Shotton J., Fitzgibbon A. W., Cook M., Sharp T., Finocchio M., Moore R., Kipman A., and Blake A. (2011). Efficient Human Pose Estimation from Single Depth Images, in *IEEE Conference on Computer Vision and Pattern Recognition*, 1297–1304.

Service, R. F. (2011). Coming Soon to a Lab Near You: Drag-and-drop Virtual Worlds. *Science*, **2011**, 331, 669-671.

Turing, A.M. (1950). Computing Machinery and Intelligence, *Mind* **LIX** (236): pp. 433–460.

Turing, A. M. (1952). The Chemical Basis of Morphogenesis. *Philosophical Transactions of the Royal Society of London. Series B, Biological Sciences*. **237** (641): pp. 37–72.