Q: What Do You Get If You Cross a Computer Scientist With a Biologist? **A:** A Digital Biologist.

Why Biologists and Computer Scientists Should Work Together

Peter J. Bentley

Department of Computer Science, University College London,
Gower Street, London WC1E 6BT, UK.
P.Bentley@cs.ucl.ac.uk
http:/www.peterjbentley.com/
http://www.cs.ucl.ac.uk/staff/P.Bentley/

Abstract. This is a time of increasing interdisciplinary research. Computer science is learning more from biology every day, enabling a plethora of new software techniques to flourish. And biology is now beginning to see the returns, with new models, analyses and explanations being provided by computers. The merging of computer science and biology is a hard thing to achieve. It takes a lot of effort. You have to overcome much resistance on both sides. But it's worth it.

In this paper, which accompanies the keynote presentation for Evolution Artificielle 2001, Peter J. Bentley discusses a new breed of scientist called the Digital Biologist, and why they are so important. Examples of research that benefit both fields will be provided, including swarming systems, computational development, artificial immune systems and models of ecologies. Only by working together will biology learn how nature works, and computer science develop techniques that have some of the awesome power of nature.

1 Introduction

"What do you get when you cross a computer scientist with a biologist?"

No, it's not the first line of a joke, although many computer scientists and biologists might laugh at the idea of working together. The biologists might find the idea that computers could have any relevance to biology very amusing. The computer scientists might find the idea that the natural world was related to their work quite funny too. But this is not a joke. It's a way of performing research.

So what do you get when they cross? Or to be more precise, what do you get when they collaborate? In truth, you get misunderstandings: headaches of new terminology or different meanings for existing terms, and sometimes even a complete inability to understand the words of your collaborator. You also get

confusing ideas, strange motivations, different ways of performing experiments, alternative ways of interpreting the results and unlikely-sounding theories. Should you pluck up the courage to attend (or even present a paper) at the conference in your collaborator's field, you get overwhelmed with all of the above multiplied by several hundred.

As difficult as all this sounds – and it is difficult – it's worth it. After a few weeks of learning each other's vocabulary you are able to communicate. The chances are you'll also find some fascinating new concepts along the way. The new ideas you hear will spark off exciting ideas of your own, the different motivations might suggest new applications to you. The alternative ways of performing experiments and analysing the results could suggest new ways for you to test your own work. The unlikely-sounding theories might explain something in your own field. And although you may feel a little lost in the alien territory of your collaborator's conference, you can guarantee there'll be at least one or two papers that will have your heart beating faster with excitement at the possibilities for your own work.

Many of the problems will never go away: you will probably always have different ways of thinking, different vocabularies and different motivations. But these are good things. Once you understand how your collaborator works, the differences produce far more significant and original research than you could have produced alone.

And sometimes, after computer scientists and biologists have worked together long enough, they change a little. They realise the value of using computers to model biological processes. They see the new understandings of nature and the new computational techniques that such interdisciplinary research can bring. They become *digital biologists*.

In this paper I argue that collaborations between biologists and computer scientists are providing the next crucial steps on the road to understanding biology and exploiting biological processes in computation. I discuss the problems of beginning collaborations and how to make them succeed. Examples of such collaborations at University College London (UCL) are provided.

2 Starting collaborations

Scientists can be very territorial creatures who loathe venturing far from their familiar surroundings. Computer scientists are perhaps more adventurous than biologists in this respect: because computers are a means to an end, these scientists have to find something for the computers to *do*. This normally means finding applications or problems to solve. While computer scientists can be talented at making up theoretical problems, these are often unsatisfying and even insufficient to test their ideas. Instead they need a real application, and this is provided by industry or academics in different fields. So, many computer scientists are quite used to working with people from outside of their field. Biologists, on the other hand, tend to be more insular. They train, research and present their results only within their

communities (and sometimes to the outside world via press releases). Now and again, some may get together from different fields and grudgingly compare notes, but this is less common. Perhaps more than any other field of science, biology is subdivided and segregated into a huge number of separate disciplines.

The nature of the fields means that should a computer scientist wish to learn about techniques inspired from biology or even about modelling biology, most will still only look within their own field for work performed by other computer scientists. And should a theoretical biologist decide that some computer modelling, visualisation or analysis is necessary, he is more likely to try and learn how to do the programming himself or use another biologist's software, than to talk to a computer scientist. These are fundamental barriers that are very effective in preventing collaboration. They are caused by lack of knowledge, misunderstandings and prejudices.

So how do we make the two sciences communicate? The answer is plain: educate the scientists. Spread the word about the research going on in the different fields. Let computer scientists know the value and relevance of biologists' research and let the biologists see the value of computer science.

Interestingly, one of the most successful ways of achieving such education is through popular science books. Although not necessarily written with this aim in mind, the genre of "pop science" allows a curious scientist to learn important achievements and discover current ideas in fields far from their own. Because such books are written for the general public, the terminology is drastically reduced (or at least explained a little more thoroughly than usual), overcoming the normal language barriers between fields. A number of collaborations at UCL between biologists and computer scientists (and mathematicians) were begun primarily because the biologist happened to read a pop science book (such as Kauffman's *Origins of Order* [1] or Bentley's *Digital Biology* [2]), or because the computer scientist read such a book (e.g., Dawkins's *The Blind Watchmaker* [3]). Other types of books also aid collaboration, for example edited collections of chapters that bring together specialists from different fields (such as Bentley's *Evolutionary Design by Computers* [4] or *Creative Evolutionary Systems* [5]).

Books are not the only trigger for collaboration. Another successful route at UCL has been the formation of special interest groups (for example, nUCLEAR: the nexus for UCL Evolutionary Algorithm Research). These meet regularly and discuss current publications or invite speakers on interesting topics. Most importantly, they focus on interdisciplinary subjects and bring together researchers from different universities as well as different fields in an informal atmosphere. A number of new collaborations and opportunities for funding at UCL have been created by these groups.

If these approaches don't appeal, there is of course the simplest of all. If you'd like to collaborate with someone in a different field, look through their web pages and publications. If they show an interest in something related to your research, just go and see them. You may get blank faces and no interest, but sometimes you may find extreme excitement and the source of an exciting new interdisciplinary research project. Again, some successful research has begun at UCL using this approach.

3 Making collaborations work

Once you've found a collaborator or two, you need to work out how to perform research together. As with any project with a number of researchers, the objectives need to be clearly understood and the work subdivided appropriately. With collaborations between biologists and computer scientists, this is more interesting.

From experience, these interdisciplinary projects tend to fall into four categories: biology-driven research, computing-driven research, parallel biology and computing-driven research and, rarest of all, combined biology and computing-driven research. Usually the initiator of the collaboration will determine the type of research. In more detail:

TYPE 1: *Biology-driven research*. Initiated by a biologist, this form of research will focus on modelling or processing the data of real biological systems. The skills of a computer scientist or mathematician are normally crucial to ensure accurate results, but the findings will be mostly of importance to biology and not computer science.

TYPE 2: *Computing-driven research*. Initiated by a computer scientist, the aims are to use the expertise and knowledge of biologists to improve existing algorithms or create new ones. The knowledge of biological processes provided by the biologist will be invaluable for the development of new computational techniques, but the results will be more significant to computer science than biology.

TYPE 3: *Parallel biology and computing-driven research*. Initiated by either specialist, this type of project is two in one. It may have began as a "Type 1" project, with the computer scientist suddenly becoming inspired into developing a new algorithm. Alternatively it may have begun as a "Type 2" project, with the biologist realising that the computer could also be used to help understand some aspect of biology. Either way, two separate strands of research form, related but distinct. The results of the research will benefit both biology and computer science equally.

TYPE 4: *Combined biology and computer-driven research*. Still the rarest form of research, these projects are the sole domain of the digital biologist. This type of research is usually initiated by biologists with some expertise in computing, or by computer scientists with knowledge of biology, and is a single project designed to benefit biology and computer science equally. Merging biology and computer science to this extent is difficult, but this type of research project can produce some very interesting results that would not be possible without such close collaboration.

At University College London we have much experience in all of these types of research. Indeed, whole research centres have been set up to tackle research in the ways described above. For example, the Centre for Mathematics and Physics in the Life Sciences and Experimental Biology (CoMPLEX) is a virtual group bringing

together mathematicians with biologists to perform "Type 1" research. The Gatsby Computational Neuroscience Unit focuses on "Type 3" research in computational models of neurons and neural networks (but has projects of all types). The Computer Science Department, UCL, also performs all types of research mentioned above.

Choosing which type of research to perform is purely subjective. If you're a biologist uninterested in anything except using the best techniques available to solve your problems, you'll favour "Type 1" projects. If you're a computer scientist only interested in developing the best techniques using some inspiration from nature, you'll prefer "Type 2" projects. But if you're a computer scientist or biologist willing to invest a little more effort in an interdisciplinary project in the hope of far greater rewards, you may consider "Type 3" or even the challenging "Type 4" projects.

4 Getting results

To illustrate the kinds of collaborations possible between biologists and computer scientists and show their benefits, the next sections briefly review a selection of projects begun by Peter Bentley at UCL's Department of Computer Science. These are, in order: artificial immune systems, swarms for learning, musical swarms, computational development, computational ecology, and evolving vision systems.

4.1 Artificial Immune Systems

Over the last four or five years, research performed by computer scientists Jungwon Kim and Peter Bentley has focussed on the combination of a set of biologically-inspired algorithms for the application of intrusion detection [6] (e.g., network intrusion detection, or the detection of hackers or unauthorised users in a system). These algorithms are all based on processes from the human immune system. They are known as *negative selection*, *clonal selection* and *gene library evolution*.[1]

Each algorithm has one aspect of our immune systems as its inspiration, and each has specific strengths. For example, the negative selection algorithm is based on the way our immune systems remove harmful antibodies from our bodies. Antibodies are generated by a variety of white blood cell known as a B-cell and help attack unwanted viruses and bacteria within us. Each B-cell produces a single, unique antibody, and a clever randomising gene expression method is employed to ensure that a huge diversity of different antibodies can be made by all the B-cells combined. Unfortunately, some B-cells produce antibodies that mistakenly attack our own 'self cells'. Luckily our immune system has a clever process (one of many) known as negative selection, which ensures that any B-cells that produce such harmful antibodies die. All that remain are B-cells that produce antibodies that do

[1] Initially developed by Stephanie Forrest in "Type 2" interdisciplinary research, which has now developed into "Type 3".

not attack self cells. In other words, negative selection tries to ensure there are antibodies for everything other than self cells.

The negative selection algorithm uses the same trick: antibodies (or *detectors* for some problem) are randomly generated. If the detectors are triggered by normal behaviour of the system they are supposed to protect, they are simply deleted. This leaves only the detectors that are not triggered by normal behaviour, or to put it another way, detectors for abnormal behaviour.

In contrast, the clonal selection algorithm is based on the way B-cells are duplicated within our bodies. As B cells produce a wide diversity of antibodies, only a very small number will be effective against a particular pathogen. But our immune system is able to increase its response until there are sufficient concentrations of the correct antibody to help destroy the pathogen. It does this by cloning B-cells that make the right kind of antibody: the more of the right kind of B cells there are, the more the corresponding kind of antibody is produced. But our immune systems also have a couple of other tricks: as well as cloning the B-cells, *hypermutation* is used, ensuring that many slight variations of the current B-cells are produced. Should any mutated B-cell produce an even more effective kind of antibody, then it will undergo clonal selection and its solution will soon propagate through our blood streams. This is an evolutionary process being used by our immune systems: new B-cells (and the DNA within them) are evolved within us to ensure the most effective immune response to pathogens. And it doesn't end there, for our immune system also generates *memory cells* that are stored within us, in case the same pathogen is encountered in the future. These cells give us immunity to the disease.

The clonal selection algorithm uses these ideas: it evolves detectors for patterns of abnormal behaviour (or *antigens*), but evaluates them in a special way. Random groups of individuals in the population are selected and 'shown' to a single antigen. The best at detecting the antigen in the group has its fitness increased; the fitnesses of the others remain unchanged. Then another random group is picked and compared to another antigen, and so on. Finally, the fittest detectors are cloned with some mutation. The result is an evolutionary algorithm that develops niches of detectors that work together to detect a large number of different antigens.

The final algorithm under investigation was gene library evolution. This is based on the way the DNA within B-cells is generated and used. As described above, each B-cell produces a unique type of antibody, which is used to help remove unwanted viruses and bacteria. The antibodies are unique because each B-cell uses a unique (and partially random) combination of DNA fragments to specify the antibody it produces. This is a tremendously clever and complex process in itself, but there's more: the DNA fragments used to build antibodies are not completely random. Many of the fragments have been carefully evolved over millions of years to ensure that *effective* antibodies are usually produced. How? Through the action of the Baldwin effect. Good DNA fragments that get used in B-cells that produce good antibodies that are effective in keeping a creature alive, are more likely to be passed onto future generations than bad ones. In effect, evolution improves the *capability* of our immune systems to adapt to as yet unseen assailants. And research in using a

genetic algorithm to evolve effective *gene libraries* which are used to produce a diverse and effective range of detectors has shown good results.

Through investigation of these ideas, research at UCL has shown some of the drawbacks of using small, highly abstracted processes from the immune system in isolation. For example, the basic negative selection algorithm was shown to be unable to cope with real-world network traffic data – randomly generating detectors was too inefficient. Recent work has focussed on combining the separate algorithms, and attempting to make a computer immune system that uses more of the carefully integrated processes in our own immune systems. To this end, we have developed a system which evolves useful gene libraries, which then specify useful detectors, which are kept valid by negative selection and improved through clonal selection. This integration of immune algorithms ensures that each process functions in a manner more similar to the way it was evolved in biological immune systems. The system is now being extended by Jungwon Kim at Kings College, London, to check continuously changing data (from the UK's Post Office) for fraud.

The research at UCL has not taken place without assistance. Prof. Robin Callard (an immunobiologist at the Institute of Child Health, London) has provided some invaluable support. The processes of our immune systems are highly complex and difficult to follow – it was the help of Robin that enabled us to understand the details of the processes we were interested in. We also gained his inside knowledge of which theories are most relevant and accurate about the immune system. His complete disregard of Jerne's Network Theory (which is used as the basis for other computer immune systems) was a surprise to us. Robin is no stranger to interdisciplinary research – after being inspired by a popular science book, he has worked with CoMPLEX to model the immune system. His assistance on this work (begun after we simply went to see him for a chat) has helped make this "Type 2" research project successful.

4.2 Musical Swarms

Another source of new biology-inspired algorithms has been the findings of Entomologists such as Nigel Franks, Jean-Louis Deneubourg, and Tom Seeley [rev. 2]. By studying the movement of insects such as bees, it has been discovered that the majority of the observed swarming behaviour can be produced by applying a small number of rules to every insect, or in a computer, *agent*. The rules cause multiple interactions of the agents with positive and negative feedback and the amplification of small random fluctuations. Together, these cause the astonishing coordination and illusion of central control so typical of swarming or flocking systems.

Exactly which rules to use in an algorithm depends on which "discoverer" you wish to follow [rev. 2]. For example, Reynolds suggests that each agent in a flock should:

1. Try to avoid colliding with any of its companions.
2. Try to move towards the centre of the flock.

3. Try and match the speed of its companions.

Alternatively, Eberhart suggests that every 'particle' in a 'particle swarm system' should also:

4. Be attracted to a 'roost' or target.

At UCL, a recent research project by Tim Blackwell and Peter Bentley investigated these ideas [7]. This research used a combination of the above rules for each agent in a swarm:

1. Try to avoid colliding with any of its companions.
2. Try to move towards the centre of the flock.
3. Be attracted to a 'roost' or target.

The work applied the swarming agents to the problem of music improvisation. Given a real-time audio input such as a saxophone or singer, the audio waveform is analysed, individual notes are identified and these are positioned into a "music space" with axies comprising pitch, loudness and start time. Because the input is constantly changing, the target continuously moves in this space. Musical agents are then allowed to swarm in the same space, each following the three rules above. As they move, their positions are used to define musical notes (for every point in the space gives a specific pitch, volume and start time). The result is a swarming behaviour that follows the target, giving the musical sensation of *listening* and *responding*, whilst the swarm's own uncertain dynamics provides novel musical ideas [7].

Although the application may be a little unusual, it did enable some interesting findings. By analysing the ability of the swarm to respond to changing targets, it was discovered that the first rule used in this work (avoid colliding with companions) played an important role in damping the oscillations of the swarm around the target. This rule is not used as standard in particle swarm optimisation research – our work suggests that making the swarm slightly more realistic will assist the ability of the swarm to search a problem space.

Again, this "Type 2" research was not performed in isolation. Assistance on insect behaviour had been provided previously by entomologist Andrew Bourke, at the Institute of Zoology in London. And this is not the only research investigating swarms at UCL.

4.3 Swarms for Learning

Another project, this time being performed by Supiya Ujjin and Peter Bentley, investigates the use of swarms for *recommender systems* [8]. These are software tools designed to learn the preferences of a shopper, and recommend products and services that are specifically tailored for each person. Such systems are already in use for many on-line stores. Often the user is asked to provide some feedback on products they have bought, and this information is used to work out suggestions. For example, if your feedback on one or more products is similar to the feedback

provided by someone else, then it is possible that you will like other products that person likes. Indeed, given sufficient data, it is even possible to predict what your feedback might be for that product.

Currently, most recommender systems do not pay much attention to the vagaries of human beings – they do not attempt to model customer's likes and dislikes with any great sophistication. But in reality, people pay attention to specific, but different features of products. For example, my main reason for choosing a movie might be because it is science fiction, while you might choose it because it stars your favourite actor. Neither of us would be served well by a system that only suggests movies based on a general voting system.

Following work using genetic algorithms for this task, research in the early stages at UCL is examining how a swarm could search a problem space of feature weightings. These would be specific to each customer and would enable the calculation of best recommendations based on a "swarmed" feature-weight profile. The ability of swarms to cope with sparse data and continuously changing data may make this swarm intelligence-based system more effective than existing approaches.

4.3 Computational Development

For some years, I have been advocating a greater use of the mapping stage from genotypes to phenotypes in evolutionary systems [rev 9]. These views followed the discovery that for many types of problem that require complex solutions, a simple one-to-one mapping would prevent evolution from finding a result. It seemed that as the complexity of the required solution increased (e.g. requiring features such as modularity, self-similarity, symmetry, duplication, and hierarchies) so the need for a new approach to evolutionary computation increased.

Looking to nature provides the solution to this dilemma: embryology, growth, morphogenesis, or more correctly, *development*. Natural systems do not have a one to one mapping from gene to phenotypic effect. A highly complex process of development uses the DNA as instructions on how to build the phenotype. There is no concept of one gene specifying one feature in nature: genes only specify proteins. The proteins from one cell trigger or suppress the activation of other genes in other cells, which trigger or suppress the activation of yet more genes in other cells, and so on. At the same time, the proteins *change* the cells: new cells are made, existing cells are destroyed, cells are told to reshape themselves, extrude substances or even to move. Some are told to have more specialist children, which then have more specialist children again, and so on, in a process of differentiation that enables the creation of over 200 different types of cell.

Through these clever processes the most complex entities on the planet are formed: you and me. There is modularity as genes that perform similar tasks become grouped together in our chromosomes, and also as cells that perform similar tasks become grouped together as organs. There is self-similarity and duplications as genes that perform useful tasks are repeated or used repeatedly, resulting in duplicated structures such as vertebrae, ribs, or segmentation in insects. There is

symmetry as the same, or similar genetic instructions are triggered on both sides of the body.

So the logical solution was to somehow incorporate development into an evolutionary algorithm. To do this required major changes to our representations: we needed genotypes that act as instructions, the use of some kind of component-based (or *cell*-based) representation to develop the phenotypes with, and possibly even a final phenotype representation [10]. Work at UCL began with some initial visits to see developmental biologist Paul O'Higgens. It soon developed further as Sanjeev Kumar joined UCL to work on this full time. We now have the support of eminent embryologists Lewis Wolpert and Michel Kerszberg and the research has developed from a "Type 2" project into "Type 3" work, with Sanjeev using a genetic algorithm to test Michel's theories in parallel to his own work on computational development. The main research, however, has been an investigation of how a biologically-plausible developmental system, that incorporates realistic gene-activation and suppression, protein diffusion and cellular behaviours, can aid the evolution of complex solutions. Previous work has shown indications that scalability and possibly evolvability may be greatly improved [11]. Other investigations (with Tim Gorden at UCL) include examining the potential of development-inspired methods for evolvable hardware. Work is ongoing to explore the capabilities of such systems further.

4.4 Computational Ecology

Ecologists also benefit from collaborations. For 18 months I have been working with Jacqui Dyer, an ecologist interested in the evolution of life in disturbed environments. She believes that traditional numerical models used by ecologists do not capture the behaviour of evolution with respect to environments prone to disasters such as earthquakes or fires. Such models predict that population dynamics in disturbed environments will fluctuate more strongly than those in stable environments, resulting in higher extinction rates, lower biodiversity and more simple community structures in disturbed, compared to stable environments. But these models ignore empirical data that show that many ecosystems evolve to overcome or even make use of such disasters for their survival. Frustrated by the assumptions and inaccuracies of numerical models, she approached me with the idea of developing a more realistic computational model. Early on, we decided to simplify things: we would model the evolution and responses of plants only.

With the help of undergraduate student Panash Shah, a model was created [12] (which I later rewrote and optimised for speed). The PLANTWORLD model was initially developed in order to examine the effects that the evolution of a functional response - in this case, dormancy - might have on the population dynamics of PLANTS. Each PLANT requires a single resource, *moisture*, which varies in availability both spatially and temporally. In addition, this implementation allows us to study the effects of two further strategies that can influence dynamics: (i) the effects of PLANT *Storage*

Capacity ii) the effects of an alternative source of moisture, in the form of a *Water Table.*

Two objectives motivated the development of this system. The far-reaching objective was to attempt to develop systems that could integrate evolutionary and ecological dynamics in spatially extensive and temporally variable environments. Such an objective is prohibited in numerical models by the sheer complexity required and is only recently becoming a realistic objective in computational models. PLANTWORLD represents only the initial stages in the development of such a system, only modelling one type of agent, PLANTS, and a single resource, moisture. However, it is capable of supporting populations of 400,000 or more at the equivalent of 24000 months every hour (on a 500 Mhz Pentium III laptop computer) and uses real rainfall data to provide realistic environmental conditions. It is envisaged that other agents (herbivores, pests, etc) and variables (nutrients, light, fire, etc) will be added at later stages. In the meantime, the development of PLANTWORLD has a more immediate objective. One of the advantages of agent-based models over numerical models of population dynamics is that our agents can exhibit *behaviours*. Combined with evolutionary computation, such behaviours can evolve. Thus, we can examine how the evolution of traits in different environments affects the population dynamics in these environments. The immediate objective for building PLANTWORLD is therefore to examine the evolution and effects of plant dormancy on population dynamics in different spatially and temporally variable environments. The simulation is not intended to capture realistic behaviour of any specific flora but rather to test the veracity of predictions about population dynamics that arise from numerical models.

This is an example of that rare type of project, “Type 4” research - the model is the result of a close collaboration and provides fascinating results for both computer science and ecology. For example, there are no fitness functions describing what is, and what is not, fit. A PLANT merely begins as a seed, which germinates given sufficient resources. It then grows until it reaches a mature size defined by a gene, and will be fertilised by a nearby mature PLANT, producing its own seeds (with sufficient resources). At all times it follows the strategies defined in its genes, going dormant or growing during certain months. If its genes help it to survive and propagate in the environment, then those genes will be passed onto its offspring. From an evolutionary computing perspective, the model provides fascinating evidence of the evolution of different solutions to a dynamically changing and unpredictable problem. Stable niches of different types of plants evolve and coexist, from tiny, short lived “grasses” to large, long-lived “trees” that can make use of the water table below. From an ecology perspective, the model shows realistic population dynamics: interdependent cycles of population sizes emerging, or the evolution of more dynamic strategies of survival for disturbed environments.

4.5 Evolving Visual Systems

The final collaborative project I will mention here is the most recent. A couple of months ago, two people contacted me within a few days of each other: Beau Lotto, a neurobiologist at UCL's Institute of Ophthalmology, and Marcel van Gerven, a student wanting to do a Ph.D. at UCL. By some stroke of luck, both wanted to do a similar kind of research: evolve neural networks for vision recognition. I put them in touch with each other, and now we have all begun work together on what should be a fruitful "Type 4" research project.

The aim is to a test Beau's general theory of how vision evolved [13], paying particular attention to colour vision. This theory suggests the visual system perceives colour based not on the light that actually reaches the eye, but on the reflectances and different illuminances that generated the stimulus in the past. So, for example, when we see a shiny black object, we perceive it as being shiny and black, even though our eyes might be seeing something that has greys, reflections and even patches of white on its surface. We know it is black because we know that in the past, such combinations of shades mean "shininess" with specific reflections and lighting. But because our visual systems make use of past experiences of the sources of different stimuli when they process current stimuli, they can be fooled. Optical illusions demonstrate this, particularly those demonstrating that we perceive colours differently depending on which other colours are nearby.

The intention is to evolve and train neural networks such that they are capable of recognising various coloured stimuli, even when under different lighting conditions. The resulting networks will then be analysed and tested, firstly to see if they are also fooled by the same optical illusions as us, and secondly to see what kinds of neural network perform such tasks.

We are hoping that the results of this research will both help explain the evolution and functioning of our own visual systems, as well as point to new ways of developing computational visual systems in the future. Whatever we learn, the chances are it will be interesting.[2]

5 Summary:

Why Biologists and Computer Scientists Should Work Togther

In this paper I have advocated greater collaboration between biologists and computer scientists. In a field known as "evolutionary computation", one would think such views are commonplace, but in reality there are surprisingly few researchers who attempt any form of communication, let alone collaboration with their biologist counterparts. Of course biologists do not have all the answers any more than computer scientists do. However, they do often have many years of experience, knowledge and understanding that is simply ignored by most of

[2] For more details on these and other projects, see *Digital Biology* [2].

computer science. Likewise, computer scientists have many years of expertise that is usually ignored by biologists.

Two years before his death in 1954, Alan Turing published a paper that laid the foundations of understanding for generations to come. The paper was entitled “The Chemical Basis of Morphogenesis”. This advance was not in computer science like much of his previous and very famous work, but in developmental biology.

Let’s not forget our roots. There have always been links between biology and computer science. By forging new ones, we can make progress in both fields at a pace greater than ever before. Digital biologists are the future.

Acknowledgements

My thanks to the following people for their assistance and for allowing me to mention our work here: Tim Blackwell, Andrew Bourke, Robin Callard, Jacqui Dyer, Marcel van Gerven, Tim Gordon, Michel Kerszberg, Jungwon Kim, Sanjeev Kumar, Beau Lotto, Paul O’Higgens, Panash Shah, Supiya Ujjin, and Lewis Wolpert.

References

1. Kauffman, S. A. (1993). *The Origins of Order: Self-Organization and Selection in Evolution*. Oxford University Press.
2. Bentley, P. J. (2001). *Digital Biology*. Hodder Headline Press, London.
3. Dawkins, R. (1991). *The Blind Watchmaker*. Penguin Books.
4. Bentley, P. J. (1999). *Evolutionary Design by Computers*. Morgan Kaufmann Publishers Inc., San Francisco, CA.
5. Bentley, P. J. and Corne, D. W. (2001). *Creative Evolutionary Systems*. Morgan Kaufmann Publishers Inc., San Francisco, CA.
6. Kim, J. and Bentley, P. J. (2001). Investigating the Roles of Negative Selection and Clonal Selection in an Artificial Immune System for Network Intrusion Detection. To appear in the Special Issue on Artificial Immune Systems in *IEEE Transactions of Evolutionary Computation*.
7. Blackwell, T. (2001). Making Music With Swarms. M.Sc. Project Report, Department of Computer Science, University College London.
8. Ujjin, S. and Bentley, P. J. (2001). Building a LifeStyle Recommender System. In *Proc.of the Tenth International World-Wide-Web Conference*. RN/01/5
9. Bentley, P. J. (2000). Representations Are More Important Than Algorithms: Why Evolution Needs Embryology. Keynote speech, ICES2000, Edinburgh, 17-19 April 2000.
10. Bentley, P. J. (2000). Exploring Component-Based Representations - The Secret of Creativity by Evolution? In Proc. of the Fourth International Conference on Adaptive Computing in Design and Manufacture (ACDM 2000), April 26th - 28th, 2000, University of Plymouth, UK.
11. Kumar, S. and Bentley, P. J.(2000). Implicit Evolvability: An Investigation into the Evolvability of an Embryogeny. A late-breaking paper in the second *Genetic and*

Evolutionary Computation Conference (GECCO 2000), July 8-12, 2000, Las Vegas, Nevada, USA.

12. Jacqueline R. Dyer, Peter J. Bentley, Panash Shah (2001) PLANTWORLD: The Evolution of Plant Dormancy in Contrasting Environments. A late-breaking paper in the third *Genetic and Evolutionary Computation Conference (GECCO) 2001*.
13. Polger, T. W., Purves, D. Lotto, B. (2000). Color Vision and the Four-Color-Map Problem. In *Journal of Cognitive Neuroscience*, 12(2):233-237.