On Biologically Inspired Computation
“a.k.a. The Field”

Essay

Technical Report 5-02

Jason Brownlee
jbrownlee@ict.swin.edu.au

PhD Candidate
Master of Information Technology, Swinburne University of Technology, 2004
Bachelor of Applied Science, Computing, Swinburne University of Technology, 2002

Centre for Intelligent Systems and Complex Processes
Faculty of Information and Communication Technologies
Swinburne University of Technology
Melbourne, Victoria, Australia

July 2005

Introduction

What is biologically inspired computation and why is it interesting? More importantly, why is it worth time/money/resources to investigate in this area of research?

These are hard questions; firstly, due to the poor use of standardised vocabulary in computer science, it is difficult to concisely define most sub-fields of study, let alone specific tools in those subfields. Secondly, questioning the validity of any field of study is a dubious exercise; surely, the answer “for the good of human knowledge” suffices. These are questions that I have recently been pondering both for my own satisfaction and for a second reason that was beginning to trouble me, namely the apparent lack of pride in the field by a few of my colleagues and peers. I reasoned that perhaps the cause was a little of the “forest for the trees” syndrome, and so I have taken it upon myself to attempt to elucidate the defendable strengths of the field as well as what I have discovered to be the poorly defined and yet commonly accepted problems or faults in the field.

This document has three aims:

1. To provide a canonical definition of biologically inspired computation as a field of study, and provide a basic framework for integrating work in the field suitable for an intelligent layperson and researcher alike.

2. To provide a clear understanding of the scope of this field for researchers who are new to this area, and also broad arguments (or perhaps a first line of defence) to justify work in the field.

3. To rekindle some passion of some current researchers in the field to support the validity of the field of study that may have been forgotten or discarded due to the pressures of overabundant scepticism.

It is hoped that this document ‘at the very least’ will generate discussion, debate, or even thought into some of the issues mentioned. I think it is a useful exercise to stop occasionally and take stock of “where we fit” in the scheme of things and somewhat objectively evaluate that position. Ideally, this work could become a living document or transformed into an online Wiki [22] and contributed to by peers and perhaps used as a “first port-of-call” by new researches, old researches lacking enthusiasm; and sceptics alike.

What is Biological Inspired Computation

Everybody has an opinion on what it is they are working on or towards, and it is difficult and problematic to use broad-sweeping labels. Nevertheless, labels and even stereotypes are needed at some point if for no other reason than a concise descriptive tool.

You do not really understand something unless you can explain it to your grandmother.

- Proverb

(*Though it helps if your grandmother has a PhD*)

Definitions depend on your audience, though I still enjoy the above proverb because it highlights again the benefit and perhaps the need, to be able to take stock, and evaluate a broader perspective. I personally think that I work on computers with software algorithm implementations that use ideas from biological systems and I refer to this endeavour as "biologically inspired computation".

Perhaps computer systems exist for the purpose of consistently repeatable automation, and work on digital computers is called computation. At the "grandmother" level, the field of study could be described as computation (*the digital computer part*) imbued with characteristics of biological systems (*the inspiration or metaphorical part*) with the intent of addressing complex problems (*artificial intelligence part*).

Top-Down

Many problems can be represented in a digital computer using the level of abstraction software provides on top of hardware. Many problems can or could be addressed with raw human power and intellect however; automation provides the capability of addressing these problems with the advantage of "the algorithmic approach", additional resource and economic advantages (among many other things).

Simple problems can be addressed with somewhat simplistic engineered operations or top-down algorithms where the computer is instructed explicitly "what to do", "how to do it", and "what to expect". This is a pseudo-standard approach to problem solving with computers and many problems are conventionally addressed in this way.

Bottom-Up

An alternative approach to addressing problems is a bottom-up methodology where rather than instructing the computer exactly how to address a problem, a solution can be devised which uses a description of the problem to determine what is needed to solve the problem. The computer is still instructed with an algorithm, though the algorithm to "go learn" the solution rather than "calculate the solution". These computer systems can be engineered, though typically are inspired by a metaphor due to the complexity associated with eliciting such an effect from the computer. One source of such metaphors is biological systems.

Perhaps a more canonical definition from within the field [13-15] would be:

> "*Biologically inspired computation is the investigation of mathematical and or engineering tools that have been imbued with selected higher level (systemic) characteristics that emerge from lower level component interactions and processes, inspired by a biological system or systems*"

The field[1] is primarily concerned with devising and demonstrating computer algorithms that attempt to (among other things) mimic the superficial characteristics of biological systems which are directed towards some computational end, such as exploiting the desired characteristics to address specific complex problems described in digital computers.

[1] defined as "Biologically Inspired Computation"

Methodology

The bottom-up methodology is typically called *computational intelligence* in the broader field of artificial intelligence. It refers to the fact that the algorithm is not instructed as to how to address a specific problem, rather through the iterative application of component-level interaction and local rules a system produces in higher-level or emergent behaviours, which are exploited for the problem.

It is useful to highlight three compositional elements common to these inspired bottom-up systems:

1. **Emergent Effects:** Systemic behaviours or characteristics that are desirable in a computational system, which emerge from exposure of the system to a particular problem instance. These are typically easy to identify, observe, and yet have complex relationships underpinning them.

2. **Local Interactions:** Iterative processes, discrete units, local information, and their interactions that are identified as engendering the desired systemic behaviours. These elements are typically simplistic meaning they are easy to describe and implement.

3. **Intermediate Dynamics:** Dynamics of a system that describe how and why discrete units and local rules result in the desired emergent behaviours. These interactions and dynamics are typically complex, non-linear, and difficult to model and describe.

The focus of the field[2] is the formation, implementation, and demonstration of tools. These tools may be analytical, mathematical, or engineering. This implies that such tools are related to the general field of computation theory, not necessarily computer algorithm implementations. The development or investigation into tools could be motivated by the need to address a specific problem (represented in a digital computer), or the motivation to imbue computational systems with specific "generally desirable" characteristics. Again, this highlights the point that the motivations for research in this field are complex and multifaceted.

Framework and Scope

The notion of "vocabulary" has connotations depending on context, and computer science specifically artificial intelligence does an excellent job of demonstrating the hazards of borrowed and re-defined vocabularies. The selection of the specific term to describe the field; "biologically inspired computation", was deliberate as it provided a somewhat less ambiguous label for the general field of study. The following provides a list of some common synonyms:

1. Bio-inspired computation
2. Biocomputation
3. Biologically motivated computation
4. Biologically inspired research
5. Problem solving from nature
6. Natural computing
7. Nature inspired computation
8. Computing with biological metaphors
9. Biomorphic Software
10. Bio-mimicry

[2] defined as "Biologically Inspired Computation"

Ambiguity in the field is caused by not only the multitude of names by which the field can be called, but also the multitude of fields that can be labelled thus (a many-to-many relationship). A useful framework or structured division for work that could be labelled in this manner is suggested by de Castro and Timmis [15] which is further refined by de Castro and Von Zuben [14]. This is somewhat supported by Forbes [16,17] and Paton [21].

Natural Computation can be broken down into three main categories:

1. **Biologically Inspired [Engineering/Computation]**
   The intent is to devise mathematical or engineering tools to address problem domains. *Biologically inspired computation* fits into this category, as do other non-computational areas of problem solving not discussed. At its simplest, its using solutions (a procedure for finding solutions is considered a solution) found in the biological environment.

2. **Computationally Motivated [Biology]**
   (*Biology with digital computers*)
   The intent of this area is to use information sciences and simulation to model biological systems in digital computers with the aim to replicate and better understand behaviours in biological systems. The field facilitates the ability to better understand *life-as-it-i*s and investigate *life-as-it-could-be*. Typically, work in this subfield is not concerned with the construction of mathematical and engineering tools, rather it is focused on simulating natural phenomena. Common examples include artificial life (A-Life), fractal geometry (L-systems, iterative function systems, particle systems, Brownian motion), and cellular automata (CA).

3. **Computing with [Biological] Mechanisms**
   (*Computation with biology/nature*)
   The investigation of alternative substrates other than silicon in which to implement computation. Common examples include molecular or DNA/RNA computing and quantum computing.

The intent of this breakdown is not to provide a framework in which all work must conform; rather it provides a high-level indication of some of the primary intents and their distinctness within the field of *natural computation*. From this breakdown, it is clear that the intention of biologically inspired computation is centred on tool *formulation*, *construction*, *demonstration*, *formalisation,* and *application* (among other things)[3].

Issues

Inspirational biological systems and processes generally have several similar fundamental properties.

[3] here a tool has a function within the context of a problem

They are typically:

- *messy* rather than *precise*
- real-time solutions are fast and approximate
- long-term solutions are highly specialised and optimised
- involve randomness for both fault tolerance and unplanned change
- probabilistic decision making
- composed of many discrete units (*redundancy*)
- act autonomously (*decentralised control*)
- exist in a spatial environment (*distributed*)
- operate at the same time (*parallelism*)

It is useful to break down the desirable characteristics of the inspired biological systems into those that are architectural and those that are behavioural (as described by Lodding [13] and somewhat by Marrow [19]). It is common in artificial intelligence, and other areas of computer science, to encode specific behaviours and architectures in a top-down manner. It is interesting to note that unlike the conventional top-down approach, biological systems are capable of demonstrating desired systemic behaviours on desirable architectures without planning or design. The following list provides an indication of common architectural and behavioural characteristics, identified in biological systems, which are desirable in various forms of computation.

Architectural Characteristics

- Decentralised control
- Distributed knowledge
- Emergence (collective interaction)
- Dynamic
- Fault & Noise Tolerance
- Flexibility
- Parallel Processing
- Persistent
- Large number of Redundant / Disposable discrete components
- Robustness
- Scalability

Behavioural Characteristics

- Adaptation
- Autonomous
- Fast and approximate solutions in the short-term
- Intelligence
- Learning
- Memory
- Probabilistic
- Self-assembly
- Specific and optimised solutions in the long-term
- Self-regulation
- Self-organisation

List 1 - Common architectural and behavioural characteristics identified in biological systems and imbued in biologically inspired systems

Inspired tools typically exploit the functionality of biological systems to achieve useful emergent behaviours with loosely coupled, parallel architectures. The result is algorithms that are able to execute complex processes we cannot completely describe to achieve tasks or address problems in a way that we completely describe – which could be considered the exhibition of human-like capabilities [20].

The Good – Top Five

This section provides the top five reasons in the literature why biologically inspired computation may be a useful area of investigation. This section attempts to preliminarily address some of the big questions such as "why use biological metaphor", "why do computation", and "why do both together".

1. "Biology is a useful motivation"

Darwin's theory of natural selection [2] suggests that the natural world has had many millions of years to devise and refine processes and structures, implying that the solutions employed may have some merit of applicability or *closeness of fit* to the natural problems in which they are addressing [15,18]. Other engineering disciplines have used biology as a metaphor devising technologies such as aircraft wings, Velcro and bullet proof vests [15,21]. Although nature can take a long time (geological time) to optimise designs, the complexity and ingenuity observed in nature can far exceed that of human designed systems [7,8,20]. Further, there are solutions that have been designed by humans that, on a retrospective analysis, have been found to already exist in the natural world (eg. radar antennae-bats ears, infrared imaging systems, sonar echolocation) [21].

2. "Software is a useful substrate for biologically inspired research"

Perhaps one of the most obvious benefits of the software substrate is the capability of easily executing controlled experiments. Software permits experimentation to be controlled and measured in all regards right down to the (pseudo) randomness that is used to govern the fundamental probabilistic nature of the tools. Software, as already mentioned, also permits the sharing of tools, the consistent repeatability of execution and the integration or embedding of tools into systems. Many specific problems and problem domains can be represented in computer software thus making them accessible to biologically inspired computational research. It is important to note that although computation can be embodied in software, software is but one part of the field of research.

3. "The field facilitates the development of alternative problem solving tools"

There are hardware limitations in terms of time, space, heat, and capacity of solid state switches (the basis for digital computers), and so biological inspiration may provide one avenue to addressing these limitations [4]. More hardware, higher clock frequencies and more memory may not be the answer; rather the answer may lie with intelligent software or better biologically inspired hardware architectures. There are problem domains which are impervious to increases in hardware or which will not be addressed even when the limitations of silicon are reached [16,17]. Further, the capabilities and complexities of software systems are not keeping pace with the increase in capacity of hardware, perhaps implying that conventional problem solving methodologies may not be suitable and that alternatives may be required [19].

4. "The innovations of biological inspiration can be demonstrated"

Perhaps the most well known demonstrations would be Sims virtual creatures [11,12] and Reynolds flocking "boids" [3]. These are great examples of the power of biological observation coupled with computation, though in applying the framework defined above it would perhaps be more accurate to classify these works as artificial life. A so-called "poster-boy" for biologically inspired computation is John Koza, and his genetic programming methods that demonstrate what is termed "human-competitive intelligence” capable of many innovations that infringe on existing patents and is capable of deriving new work that is patentable [10]. This is but one extreme example which is heavily evangelised. A more useful demonstration is to simply say that AI, including biologically inspired techniques, is everywhere [9]. AI is pervasive in our technological culture. Once AI techniques become commonly used “mainstream”, they are no longer called “AI” but just “tools” or “intelligence” from dishwashers, to mobile phones, cameras, cruse control, etc. This is what the American Association for Artificial Intelligence (AAAI) refers to as the “AI Effect” [1]. It should suffice to say there are many good case study implementations in literature for specific techniques, and many that will never be written about due to concerns of commercial confidence.

5. “The tools typically have simplistic underpinning”

There is simplicity in the field, which is one of the reasons why it attracts such a diverse array of people. Firstly, there is simplicity in the higher-level emergent behaviours in that they are typically easy to identify and qualify (though not easily quantified) in biological and computation systems alike. Secondly the discrete units, local interactions, and processes are easy to describe in terms of their biological inspiration, easy to relate to the uninitiated (grandmother), and easy to implement in computer software. This provides a “low barrier of entry” for implementing computational tools that have behaviours and architectures, which are complex.

The Bad – Top Five

This section provides the top five identified problems in the field of biologically inspired computation collected mostly through first hand conversations and correspondence with peers and colleagues. The problems listed are not perpetrated by the majority in the field, and it is not even valid to call these generalisations. These are identified problems that occur in the minority, though have far reaching effects both of the perception within and from outside of the field of study.

1. “Lack of rigor and discipline”

The systems are simplistic at the lowest level, and the emergent properties are easily identified and qualified at the highest level, though there are problems with the area in between, what I have referred to as the intermediate dynamics. As mentioned, it is difficult to describe how and why the systems work because of the complexities involved. The mathematics required is typically underdeveloped, esoteric or requires the technique to be reduced to a trivial form. There is a lag time for the theory of these tools over their application, which results in the implementation of systems that are not fully understood. The field, like the broader field of computer science, does suffer from some lack of scientific discipline. There can be somewhat poor or ineffective use of statistics, and a poor or lack of experiential description, which impedes reproducibility. Given that a part of the field (perhaps an un-proportionally large part) is focused on implementation of tools, there is an observed distinct lack of software engineering principles, which results in a lack of reuse (duplication of effort) and lack of testing resulting in lack of trust in software. These effects likely stem from the fact that the field attracts people from so many disparate disciplines that are simply not educated in statistics, software engineering or scientific methodology. Rigor is an open problem, whereas discipline is perhaps somewhere we need to raise the bar.

2. "Departure from biology"
One interpretation for the field is the "construction of tools inspired by characteristics of biological systems". A problem with this premise is the determination of how much of the biology and how much tool to use / investigate. Consider that perhaps biology is only useful in the inspiration phase of development, and should be departed from in most if not all regards (see 3. nomenclature) in the tool phase. This requires all tool characteristics, properties and dynamics to be described and defended in other contexts such as mathematics, or the problem/s the tool is suited for. Given the simplicity of the metaphor, it can be tempting to "hold on" to the biology for too long, which may hinder the goal of the exercise. An example of this would be the defence of tool feature modifications and additions in the context of the biological inspiration, and the duplication of all biological features rather than those that are identified as engendering the desired properties. The advantage of not modelling biological systems is that we can drop biology and mangle / augment tools in such a way as to bare little or no resemblance to the biological systems from which they were inspired while maximising the inspired benefit.

3. "Lack of standardised nomenclature"
The use of biological terminology is ubiquitous in the field, even though we are <u>not</u> modelling biological systems. The use of vocabulary from the biological inspiration is related to (2) the problem of knowing when and in what way to depart from biology. Perhaps it is not suitable to use any biological terminology at all when defining and describing the inspired techniques, after all we are addressing computation, not simulating biology. This can be a less than comfortable thought for some, though there are consequences for lacking a consistent nomenclature. The terms from biology were devised to describe biological systems and when used out of the scope of biology can have undesired connotations depending on context and audience, requiring the specific meaning to be repeatedly defined to ensure useful communication. The classification, definition and ultimately the relations between tools / techniques becomes muddied and problematic. This results in difficulty in locating relevant or related work impinging on productive research and perhaps the greater growth and maturity of the field.

4. "(Un) Suitability of application"
The resulting techniques can be described in a manner that is simple enough to be implemented by a programmer or applied by someone without a full understanding of the implications or suitability. This is a problem firstly (as mentioned) in that it is not known with sufficient rigor <u>how</u> or <u>why</u> many of the tools work, and secondly the problem instances or domains to which they are best suited is not known. As mentioned numerous times, the tools/systems can be complex. Empirical investigation is an alternative to analytical rigor, and is typically applied to investigate the suitability of a technique through the use of exploratory evaluation/demonstration on "data sets". It is important to differentiate between <u>investigation</u> (demonstration) and <u>application</u>, and this line is easily blurred in the greater field computer science given the ease in which problems can be described in software. There are common data sets for exploratory and demonstration purposes, though there are few standardised catalogues on categories of problems. Further, it is problematic to generalise performance on demonstration "toy" problems and datasets to real-world applications. Perhaps the failure of unsuitable application is the most commonly cited by critics. It is an area researches and developers need to pay close attention.

5. "Lessons from "*no free lunch*" are acknowledge but not practiced"

The crux of the "no free lunch" theorem [5] is that there are no algorithms that are good at <u>all problems</u> – there is no such thing as a "silver bullet" or "god" algorithm. Many biologically inspired techniques are considered "general problem solvers", in that they can be applied to many different problem domains and find solutions, however they will not be the best method of finding solutions to all problems. A tool / technique devised for a specific problem domain, that incorporates domain specific information, will outperform a general problem solver – this is the duality nature of <u>specialisation</u> and <u>generalisation</u> [6]. This lesson seems to be easily forgotten when a biologically inspired tool works on each problem instance it is tested on, when the very nature of general problem solvers is that they are "fire-and-forget" on most problem domains. This problem is, again related to that of <u>suitability</u> of application. It is an important open question in artificial intelligence, and specifically in biologically inspired techniques, to identify suitable relationships between techniques and problem domain applications.

Concluding Remarks

The digital computer is a relatively new invention, and its use as an automation tool is prevalent is most fields of research. Computer science can be observed as an area of research, which addresses how to make better use of the digital computer. Artificial intelligence and biologically inspired computation have existed somewhat since the inception of the computer (Alan Turing and others) though the fields and have grown as areas of research in response to the development and rapid adoption of the desktop personal computer. Perhaps this rapid expansion the field suffers the stigma of immaturity as a result of some of the problems identified above, and will continue to do so until such time as the pace slows or a great standardisation or unification within the field occurs.

To cut both ways
- Idiom

Conjecture and hand-waving aside, if nothing else, the elucidation of the attractive features and areas of concern within the field of biologically inspired computation have highlighted some well-known and observed dichotomies. In addition to the commonly sighted efficiency versus efficacy (commonly referred to as exploration versus exploitation in search) from complexity theory, and specialisation versus generalisation from "no free lunch" theorem, perhaps there is a biology-tool or an "easy to use"-"hard to use well" duality in this field of study. As discussed, the very properties that make the field attractive to such a diverse array of people, perhaps hinders the growth or potential of the field of study.

Although the lasting effects of this type of work are unknown, I am proud to be a part of this field and I hope that at the very least this work has prompted some thought or debate on the nature of what this thing is that we do.

Acknowledgements

I'd like to thank my colleagues at the Centre for Intelligent Systems and Complex Processes (CISCP) that were kind enough to take the time to discuss the "hard questions" with me; Tim Hendtlass, Nigel Fisher, Irene Moser, Andrew Murray, Hans Pierrot, Ben Prime, Cameron Taylor, and Clinton Woodward. I would like to thank Daniel Angus for his thorough review and discussion of an early draft. Thanks to Steven Dower for additional grammatical and style suggestions. Finally, I would also like to thank various other people (whom I have forgotten) who helped prompt and address this line of inquiry.

References

[1] American Association for Artificial Intelligence (AAAI). The AI Effect. [Online] http://www.aaai.org/AITopics/html/aieffect.html . American Association for Artificial Intelligence (AAAI). 2005.

[2] Charles Darwin. *On the origin of species by means of natural selection, or, The preservation of favored races in the struggle for life*, Champaign : Project Gutenberg, 1859.

[3] Craig W. Reynolds, "Flocks, herds and schools: A distributed behavioral model," *Proceedings of the 14th annual conference on Computer graphics and interactive techniques,* pp. 25-34, 1987.

[4] Dan Hammerstrom, Biologically Inspired Computing *Cavin's Corner, Column written for Semiconductor Research Corporation*, vol. 2001.

[5] David H. Wolpert and William G. Macready, "No Free Lunch Theorems for Search," Santa Fe Institute, Sante Fe, NM, USA, Technical Report SFI-TR-95-02-010, 1995.

[6] Hans-Paul Schwefel. Advantages (and disadvantages) of evolutionary computation over other approaches. In: *Evolutionary Computation 1 - Basic Algorithms and Operations* , eds. Thomas Back, David B Fogel, and Zbigniew Michalwicz. UK: Institute of Physics Publishing, 2000.pp. 20-22.

[7] Janine Benyus. Biomimicry. Pop!Tech 2004 (Maine, USA. [Online] http://www.itconversations.com/shows/detail241.html . IT Conversations. 2004.

[8] Janine M. Benyus. *Biomimicry: Innovation Inspired by Nature* , Perennial , 1998.

[9] Jennifer Kahn, It's Alive! *Wired*, vol. no. 10.03 , Mar, 2002.

[10] John R. Koza, Martin A. Keane, Matthew J. Streeter, William Mydlowec, Jessen Yu, and Guido Lanza . *Genetic Programming IV: Routine Human-Competitive Machine Intelligence*, Cambridge, Mass, USA:

Kluwer Academic Publishers, 2003.

[11] K.Sims, "Evolving 3D Morphology and Behavior by Competition," *Artificial Life,* pp. 353-372, 1994.

[12] K.Sims, "Evolving Virtual Creatures," *Proceedings of the 21st annual conference on Computer graphics and interactive techniques,* pp. 15-22 , 1994.

[13] Kenneth N Lodding and NASA, The Hitchhiker's Guide to Biomorphic Software *Queue*, vol. 2, pp. 66-75, Jun, 2004.

[14] Leandro N de Castro and Fernando J. Von Zuben. Chapter 1: From Biologically Inspired Computing to Natural Computing. In: *Recent developments in biologically inspired computing*, eds. Leandro N de Castro and Fernando J. Von Zuben. Hershey, PA: Idea Group, 2005.

[15] Leandro N. de Castro and Jon Timmis. *Artificial Immune Systems: A new computational intelligence approach*, Springer-Verlag, 2002.

[16] Nancy Forbes , Biologically Inspired Computing *Computing in Science and Engineering*, vol. 2, pp. 83-87, 2000.

[17] Nancy Forbes. *Imitation of life: how biology is inspiring computing*, Cambridge, Massachusetts: Mit Press, 2004.

[18] Nigel Shadbolt, From the Editor in Chief: Nature-Inspired Computing *IEEE Intelligent Systems*, vol. 19, 2004.

[19] P Marrow, Nature-Inspired Computing Technology and Applications *BT Technology Journal*, vol. 18, pp. 13-23, Oct, 2000.

[20] Peter Bentley. An Introduction to Evolutionary Design by Computers. In: *Evolutionary Design By Computers*, ed. Peter J. Bentley. CA, USA: Morgan Kaufmann , 1999.

[21] Ray Paton. Introduction to computing with biological metaphors. In: *Computing With Biological Metaphors*, ed. Ray Paton. London, UK: Chapman & Hall, 1994.pp. 1-8.

[22] Wikipedia. Wiki. [Online] http://en.wikipedia.org/wiki/Wiki . 2005. Wikimedia Foundation Inc. 2005.