Genet Program Evolvable Mach (2017) 18:399–405
DOI 10.1007/s10710-017-9289-9

Just because it works: a response to comments on "On the Mapping of Genotype to Phenotype in Evolutionary Algorithms"

Peter A. Whigham[1] · Grant Dick[1] · James Maclaurin[2]

Published online: 24 February 2017

Abstract This response examines the context and implications of the comments to "On the Mapping of Genotype to Phenotype in Evolutionary Algorithms" that appears in this journal. The notion of metaphor is first considered and then the general themes of the commentaries addressed. The response subsequently focuses on representation and operators, noting that many of the comments support our basic premise.

The main conclusion is that Sterelny's conditions do form a suitable basis for representation and operator design and that the collection of responses form an excellent basis for further discussion and research in evolutionary computation.

Keywords Genetic programming · Biological analogy · Grammatical evolution · Representation

Our paper "On the Mapping of Genotype to Phenotype in Evolutionary Algorithms" has attracted several interesting responses [1–7]. We thank our colleagues for taking the time to read our work and provide comments. Hopefully, our paper and the subsequent responses will drive future discussion and research into the efficient design and application of evolutionary algorithms.

We begin by noting a point about the language employed in this debate. Many of those who commented on our article (Ekárt and Lewis, Kell, O'Neill and Nicolau,

Commentaries to this article can be found at doi: 10.1007/s10710-017-9287-y, 10.1007/s10710-017-9290-3, 10.1007/s10710-017-9291-2, 10.1007/s10710-017-9292-1, 10.1007/s10710-017-9293-0, 10.1007/s10710-017-9294-z, 10.1007/s10710-017-9295-y, 10.1007/s10710-017-9296-x.

✉ Peter A. Whigham
peter.whigham@otago.ac.nz

1 Department of Information Science, University of Otago, Dunedin, New Zealand

2 Department of Philosophy, University of Otago, Dunedin, New Zealand

as well as Squillero and Tonda) referred to the use of evolution as a metaphor inspiring work on evolutionary algorithms. Some were specifically quoting Sörensen's "Metaheuristics—the metaphor exposed" [8]. The general tenor of these comments was that "evolution" here was merely a façon de parler, a source of inspiration. The use of "metaphor" in this context is unfortunate and potentially confusing.

When we employ a metaphor, we appear to assert the identity of two things, as in "time is money" or "all the world's a stage". We do this for rhetorical effect which might, in turn, be pedagogical or perhaps just recreational. We are not of course really asserting the identity of these things. We do not mean that time really is money or that the world really is a stage. So when computer scientists refer to "evolution", are they talking about the very same process that drives adaptation in biological populations? The reason it is confusing to refer to "evolution" as a metaphor here is that the question about whether or not biological evolution and evolutionary computation are instances of the very same process, does not have a simple answer. This is because most processes can be described at various levels of abstraction. By "evolution" we might just mean any cumulative change that is an instance of the Darwinian Principles; variation, inheritance, and differential fitness [9, p. 1]. If so, then the evolution of computer code, and languages, and social customs, and cats and dogs will all be instances of the *very same* process. Conversely, if by "evolution" we mean to refer to "all the functions performed by all the elements of the cellular machinery that secure biological inheritance and variation" then talk about evolutionary computation really is mere metaphor.

In short, whether or not some particular use of "evolution" by a computer scientist counts as a metaphor depends on the intentions of the speaker. So, we cannot objectively evaluate the claim that such use is merely metaphorical. Worse still, such debate threatens to mask the real question to be answered here, namely "in what ways should biological and computational systems be similar in order for fitness to drive adaptation in the way that we intend?"

A common theme that is present in several responses is that evolution in biological systems is complex, and Sterelny's conditions are violated by replication as observed in nature [1, 3, 5]. This is true, and precisely why they serve as useful guidelines for evolutionary computation (EC). Sterelny's conditions identify a set of characteristics under which an idealised "inheritance system" should operate, rather than the way replication actually operates in biological systems [10]. In evolutionary computation, we have the luxury of being able to construct efficient inheritance systems free from the constraints of the natural world. Therefore, Sterelny's conditions form an excellent base from which synthetic evolution, such as evolutionary computation, could be conducted.

Another theme in the responses raise questions about the ease with which our proposed guidelines for representation design can be applied [1–3]. Some argue that the guidelines are imprecise and prevent rigorous analysis, while others claim that the guidelines are of limited use as we cannot know what the resulting evolutionary dynamics will be when designing a representation. In response, we would suggest that our proposal is not intended for rigorous mathematical analysis. Our recommendations do not define a strict set of rules that must be followed. Instead, they should be considered as a set of guidelines that become more informative when coupled with knowledge of the problem domain. In terms of the practical utility of our guidelines, we would argue that adopting an evolutionary search implies some middle ground between complete knowledge of search behaviour, which permits a strong, directed search for solutions, and a complete lack of domain knowledge, in which the only recourse is random search. It is not acceptable to assume that evolutionary computation will be used when one does not understand the structure of the search space; if the knowledge of the problem structure is completely absent, then the only justifiable search strategy is random, not evolutionary. Therefore, adopting an evolutionary approach implies knowledge that inheritance and replication will yield benefits over pure random search. Even with only partial knowledge, our recommendations become useful guidelines to support the design of more effective representations. The guidelines that we propose are idealistic, and will inevitably be violated to some degree when designing representations and operators. However, if they are used to steer the general direction of representation and operator design, then we believe that the resulting evolutionary algorithms will be better suited to complex problems.

In addition to the overall themes of commentary, each response raised specific issues that we will now address.

Altenberg in general agrees with our argument that caution must be applied when using biological analogies to inspire algorithm design [2]. However, he also highlights the importance of not overlooking the dynamics of the evolutionary system itself. Evolutionary processes can be unpredictable and produce what appear to be counterintuitive results, such as bloat in genetic programming and the 1-in-5 rule of evolution strategies [11, 12]. We agree that the evolutionary dynamics of systems should not be overlooked, but do not consider our recommendations to be orthogonal to this concern. For example, identification of the 1–5 rule led to the subsequent design of an adaptive mutation mechanism to encourage more rapid accumulation of advantageous mutations. We argue that this outcome is aligned with the conditions put forth by Sterelny's for an ideal inheritance system.

Ekárt and Lewis suggest that justification is needed around our apparent implication that researchers assume that evolutionary processes are optimal [3]. It is indeed fair to state that few, if any, evolutionary computation researchers would make such a claim, but then neither do we. Instead, we propose that there is a tacit assumption that the genotype-phenotype mapping found in molecular biology is the best model for evolutionary search (regardless of whether evolutionary search itself is optimal). We see evidence of this in previous work, such as:

> We believe GE to have closer biological analogies to nature than GP, particularly with its use of linear genomes and the manner in which it uses proteins to affect traits [13, p. 95]

and:

> GE further extends inspiration taken from the biological analogy by employing neo-Darwinian principles of genetics that have been uncovered by molecular biologists. The most significant of these is the adoption of a

distinction between the genotype and phenotype, similar to that which exists in nature [14, p. 357]

Examples such as these, while not strictly inferring optimality, do suggest a view that increasingly faithful replication of biology is desirable.

Ekárt and Lewis also raise the issue of designing generic representations and operators, and the implication that this has for evolutionary systems in practice. We do not wish to raise this as an issue within this response, except to propose that discussions on this topic are present in the literature (take, for example, the historical discussion of binary coded versus real coded genetic algorithms [15–18]). Finally, Ekárt and Lewis propose a framework upon which representation design should be conducted. Our intention was not to produce such a framework, but rather introduce a set of loosely defined guidelines from which such frameworks could be developed. To this end, we thank them for their contribution, and hope that this encourages more work in this area.

Kell states that representation and the mapping of genotypes to phenotypes is not the only mechanism that produces suboptimal evolution in natural systems, and reminds us of a number of properties of natural systems that are inefficient [4]. He suggests that our interpretation of 'molecular biology' is limited to natural systems, and that other approaches to molecular biology are possible. Kell subsequently introduces the concept of synthetic biology, in which complete control over the direction that evolution takes is made possible through exploitation of domain knowledge of gene sequences. Synthetic biology may be an interesting area of inspiration for future evolutionary algorithm design, and we agree that our interpretation of molecular biology is limited to natural systems. It is worth noting that the directed evolution of synthetic biology appears to align very well with the guidelines that we propose for effective evolutionary search.

O'Neill and Nicolau argue that the concerns raised in our paper are addressing the greater field of EC rather than specific to the issue of genotype to phenotype mappings [5]. Having adopted this larger view, they then argue that the field of EC has moved on from discussions of optimality. We have no dispute with this view, and do not claim that the entire EC community believes in the optimality of evolutionary search. While the abstract in our paper may have led to a view that we claim optimality, we believe that the remainder of our paper expresses our view more clearly. However, we remain committed to the view that aspects of EC research have an underlying tacit assumption that increasing fidelity to biological evolution is conducive to effective search. We believe that this underlying assumption needs to be justified.

An additional concern raised by O'Neill and Nicolau relates to the manner in which grammatical evolution (GE) is considered by our work, as well as an earlier paper we wrote comparing GE with context-free grammar genetic programming (CFG-GP). In particular, they suggest that the version of GE we examine is a historical artefact and does not reflect the significant research made in subsequent years to improve its performance. Indeed, in both of our papers, we examine GE in its purest form, where the search operators are completely decoupled from the phenotype. As we and O'Neill and Nicolau have noted, subsequent improvements to

GE make extensive use of knowledge of the grammar space (either through redesigning the grammar, or making use of derivation tree information) to improve search [5, 19, 20]. In doing so, GE search no longer operates in the genotype, but effectively operates in the space of derivation trees, resulting in a search behaviour that closely resembles CFG-GP. To be clear, we do not aim to question the search performance of more modern variants of GE. Instead, our question is this: had GE been developed under the guidance of Sterelny's conditions and our subsequent recommendations, would its initial design have been more effective, and therefore not require subsequent research to address its limitations?

Perhaps the most prudent advice from O'Neill and Nicolau is found in the final sentence of their response: "What are the sufficient set of features of natural, genetic, evolutionary and developmental systems, which can translate into the most effective computational approaches for program synthesis?" Clearly, we are in agreement with this, and hope that this sentiment is shared and embraced by the greater EC community in future work.

Ryan also takes particular exception with our examination of grammatical evolution. As a co-inventor of GE, it is understandable that he feels GE was singled out [6]. Although other examples could have been examined, the choice of GE best fits the scope of the paper and of *Genetic Programming and Evolvable Machines*. Ryan also raises the issue that grammatical evolution "works", and presents numerous examples to support his claim. We completely agree that GE works as a search technique. However, for effective application, it is not sufficient that something just works—selecting the best example from a random sample also works as a search mechanism. Instead, we argue that the research community needs to focus its attention on designing operators that work as well as possible on the problems being tackled. We need to move past a mentality of "it works" and make effective search a stronger priority in evolutionary algorithm design. This should also mean that good benchmarking methods are used, however this is beyond the scope of this commentary and is discussed elsewhere [21, 22].

Foster argues that biological systems are too complex to be analysed under the lens of Sterelny's conditions, and that biological evolution differs from evolutionary computation in the constraints to which it is subject [1]. They then proceed to examine our proposed guidelines from an engineering perspective, and suggest that, while admirable, the advice that they give may not be practical. The chief concern is that EC methods are typically used when little is known about the structure of the search space, and that our advice is dependent upon the presence of domain knowledge. However, it is important to note that our proposed guidelines can be adopted even with only partial knowledge of the structure of the search space. A complete picture of the search space is not necessary (and, as pointed out in Foster's response, if this was known then an evolutionary search would not be a suitable choice).

Squillero and Tonda extend our discussion and question the relevance of biological analogies in future work in the design of evolutionary algorithms [7]. They then present memetic algorithms as an example of evolutionary search that does not adhere to a strong biological analogy. They conclude with a remark that they do not believe that Sterelny's conditions serve as a useful guide for steering future research in established evolutionary algorithms. Squillero and Tonda's remarks are a welcome and useful contribution to this discussion to contextualise it into the larger body of evolutionary computation research. However, we disagree with their final remark on the utility of Sterelny's conditions for improving established EAs. While they might be best suited to guiding a clean-slate implementation of new evolutionary algorithms, we still feel that there is utility in these guidelines for re-evaluating existing EA concepts.

Once again, we thank all authors for their responses. Collectively, we feel that our work, in conjunction with these responses, should serve as a stimulus into stronger, more effective application of evolutionary search. We look forward to seeing future developments in this area in the EC community.

Acknowledgements Special thanks must go to Lee Spector, Editor-in-Chief of Genetic Programming and Evolvable Machines, for managing the submission and editorial process for this discussion. His efforts in streamlining this process have been greatly appreciated.

References

1. J.A. Foster, Taking "biology" just seriously enough: Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Peter A. Whigham, Grant Dick, and James Maclaurin. Genet. Program. Evolvable. Mach. (2017). doi:10.1007/s10710-017-9296-x
2. L. Altenberg, Probing the axioms of evolutionary algorithm design: Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Peter A. Whigham, Grant Dick, and James Maclaurin. Genet. Program. Evolvable. Mach. (2017). doi:10.1007/s10710-017-9290-3
3. A. Ekárt, P.R. Lewis, Genotype–phenotype mapping implications for genetic programming representation: Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Peter A. Whigham, Grant Dick, and James Maclaurin. Genet. Program. Evolvable. Mach. (2017). doi:10.1007/s10710-017-9291-2
4. D.B. Kell, Evolutionary algorithms and synthetic biology for directed evolution: Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Peter A. Whigham, Grant Dick, and James Maclaurin. Genet. Program. Evolvable. Mach. (2017). doi:10.1007/s10710-017-9292-1
5. M. O'Neill, M. Nicolau, Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Whigham, Dick and Maclaurin. Genet. Program. Evolvable. Mach. (2017). doi:10.1007/s10710-017-9293-0
6. C. Ryan, A rebuttal to Whigham, Dick, and Maclaurin by one of the inventors of grammatical evolution: Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Peter A. Whigham, Grant Dick, and James Maclaurin. Genet. Program. Evolvable. Mach. (2017). doi:10.1007/s10710-017-9294-z
7. G. Squillero, A. Tonda, (Over-)Realism in evolutionary computation: Commentary on "On the mapping of genotype to phenotype in evolutionary algorithms" by Peter A. Whigham, Grant Dick, and James Maclaurin. Genet. Program. Evolvable. Mach. (2017). doi:10.1007/s10710-017-9295-y
8. K. Sörensen, Metaheuristics—the metaphor exposed. Int. Trans. Oper. Res. **22**(1), 3–18 (2015)
9. R.C. Lewontin, The units of selection. Annu. Rev. Ecol. Syst. **1**, 1–18 (1970)
10. K. Sterelny, Niche construction, developmental systems and the extended replicator, in *Cycles of Contingency: Developmental Systems and Evolution*, ed. by S. Oyama, R.D. Gray, P.E. Griffiths (MIT Press, Cambridge, Mass, 2001)
11. R. Poli, A simple but theoretically-motivated method to control bloat in genetic programming, in *European Conference on Genetic Programming* (Springer, 2003)
12. I. Rechenberg, *Evolutionsstrategie: Optimierung Technischer Systeme Nach Prinzipien der Biologischen Evolution* (Frommann-Holzboog, Stuttgart, 1973)
13. C. Ryan, J. Collins, M.O. Neill, Grammatical evolution: Evolving programs for an arbitrary language, in *Genetic Programming: First European Workshop, EuroGP'98* Paris, France, April 14–15*, 1998 Proceedings*, ed. by W. Banzhaf, et al. (Springer, Berlin, Heidelberg, 1998) pp. 83–96

14. A. Brabazon, M. O'Neill, S. McGarraghy, *Natural Computing Algorithms* (Springer-Verlag, Berlin, Heidelberg, 2015)
15. L. Davis, Hybridization and numerical representation, in *Handbook of Genetic Algorithms*, ed. by L. Davis (Van Nostrand Reinhold, New York, 1991) pp. 61–71
16. D. Goldberg, Real-coded genetic algorithms, virtual alphabets, and blocking. Complex Syst **5**(2), 139–167 (1991)
17. C.Z. Janikow, Z. Michalewicz, An experimental comparison of binary and floating point representations in genetic algorithms, in *Proceedings of the Fourth International Conference on Genetic Algorithms*, ed. by R.K. Belew, L.B. Booker (Morgan Kaufmann, San Diego, CA, 1991) pp. 31–36
18. A.H. Wright, Genetic algorithms for real parameter optimization, in *Foundations of Genetic Algorithms*, ed. by G.J.E. Rawlins (Morgan Kaufmann, San Mateo, California, 1991) pp. 205–218
19. P.A. Whigham, G. Dick, J. Maclaurin, *On the Mapping of Genotype to Phenotype in Evolutionary Algorithms.* Genetic Programming and Evolvable Machines, 2016. TBA
20. P.A. Whigham et al., Examining the "Best of Both Worlds" of grammatical evolution, in *Proceedings of the 2015 Annual Conference on Genetic and Evolutionary Computation* (ACM, Madrid, Spain, 2015), pp. 1111–1118
21. J. McDermott et al., Genetic programming needs better benchmarks, in *Proceedings of the 14th Annual Conference on Genetic and Evolutionary Computation* (ACM, Philadelphia, Pennsylvania, USA, 2012) pp. 791–798
22. D.R. White et al., Better GP benchmarks: community survey results and proposals. Genet. Program. Evol. Mach. **14**(1), 3–29 (2013)