

# Exploiting Genomic Knowledge in Optimising Molecular Breeding Programmes: Algorithms from Evolutionary Computing

Steve O'Hagan<sup>1,2</sup>, Joshua Knowles<sup>2,3</sup>, Douglas B. Kell<sup>1,2\*</sup>

**1** School of Chemistry, The University of Manchester, Manchester, United Kingdom, **2**The Manchester Institute of Biotechnology, The University of Manchester, Manchester, United Kingdom, **3** School of Computer Science, The University of Manchester, Manchester, United Kingdom

## Abstract

Comparatively few studies have addressed directly the question of quantifying the benefits to be had from using molecular genetic markers in experimental breeding programmes (e.g. for improved crops and livestock), nor the question of which organisms should be mated with each other to best effect. We argue that this requires *in silico* modelling, an approach for which there is a large literature in the field of evolutionary computation (EC), but which has not really been applied in this way to experimental breeding programmes. EC seeks to optimise measurable outcomes (phenotypic fitnesses) by optimising *in silico* the mutation, recombination and selection regimes that are used. We review some of the approaches from EC, and compare experimentally, using a biologically relevant *in silico* landscape, some algorithms that have knowledge of where they are in the (genotypic) search space (G-algorithms) with some (albeit well-tuned ones) that do not (F-algorithms). For the present kinds of landscapes, F- and G-algorithms were broadly comparable in quality and effectiveness, although we recognise that the G-algorithms were not equipped with any 'prior knowledge' of epistatic pathway interactions. This use of algorithms based on machine learning has important implications for the optimisation of experimental breeding programmes in the post-genomic era when we shall potentially have access to the full genome sequence of every organism in a breeding population. The non-proprietary code that we have used is made freely available (via Supplementary information).

**Citation:** O'Hagan S, Knowles J, Kell DB (2012) Exploiting Genomic Knowledge in Optimising Molecular Breeding Programmes: Algorithms from Evolutionary Computing. PLoS ONE 7(11): e48862. doi:10.1371/journal.pone.0048862

**Editor:** Josh Bongard, University of Vermont, United States of America

**Received:** July 6, 2012; **Accepted:** October 4, 2012; **Published:** November 21, 2012

**Copyright:** © 2012 O'Hagan et al. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

**Funding:** No current external funding sources for this study.

**Competing Interests:** The authors have declared that no competing interests exist.

\* E-mail: dbk@manchester.ac.uk

## Introduction

A common circumstance in biotechnology [1] is that we have an entity (be it a protein, nucleic acid or organism) that we wish to improve for some specific purposes, and this is largely done by genetic means, involving either the mutation of a single parent entity or (if mating is possible) by means that additionally involve genetic recombination. Of course, observations of the success of experimental breeding (or 'directed evolution') contributed significantly to Darwin's thinking, even before we knew anything about the existence of genes, and many examples (e.g. [2–5]) show the huge variation in phenotype achievable in inter-breeding populations. The usual metaphor here, due to Sewall Wright [6], is that of a 'fitness landscape' that relates genotype (i.e. sequence) to fitness, and that involves moving around this landscape by mutation and recombination while seeking to improve the fitness(es) of our offspring en route to some variety that is highly improved relative to the starting position.

We note too that we are usually interested in optimising for multiple traits, which may be largely independent of or at best only partially linked to each other (e.g. disease resistance in a crop plant is essentially independent of the metabolic processes governing primary yield, but would potentially contribute to overall yield, and we might select for larger roots [7,8] separately from the

agronomic benefits such as drought tolerance that they might bring). This makes our searches for improved strains a multi-objective optimisation problem [9]. Of course plant and animal breeding holds considerable potential for improving food security, a topic of substantial current interest [10–17].

The issue, then as now, is that mostly we carry out breeding experiments 'in the dark' genetically, and though we may seek to combine complementary traits from individual parents (whether phenotypically or via Quantitative trait loci (QTLs)), the number of experiments we can typically carry out in a generation is tiny relative to the number of possible matings (given the size of the parental populations available) – if there are *m* and *f* possible parents of each gender the number is obviously *mf*. Note too the gigantic (genetic) sequence space: a microarray of just the set of 30mers, made using spots of just 5 μm diameter, would cover 29 km<sup>2</sup> [18]! This begs the question of how we optimise the breedings we choose to perform, i.e. which parents we select to mate with each other, and which offspring we select for further breeding. It might be assumed that it would simply be best to breed from the fittest parents obtained in the previous generation, but this is well known not to be the case, since (i) the landscapes are both complex and epistatic (the allele in one location affects what is optimal in another location), and (ii) most mutations tend to be deleterious and would therefore be selected out, even if it was in

**Table 1.** Some evolutionary algorithms: F-Algorithms.

ALGORITHM	COMMENTS	REFERENCES
Breeder genetic algorithm	Basic evolutionary algorithm using truncation selection	[74,148]
Phenotypic niching with fitness sharing	The reproductive opportunities of individuals are shared amongst members of a niche. A niche is defined by a neighbourhood in phenotype-space, i.e. as a vector of attributes or traits. The scheme seeks to preserve diversity.	[130,149]
Deterministic crowding	Crowding is a reproduction scheme in which individuals are forced to replace individuals in the population that are most like them. In deterministic crowding this is achieved without inspection of the genotype; offspring merely replace their parents (depending on the relative fitness of the parents and offspring). Preserves diversity.	[150,151]
Local selection; local breeding; local mating; spatially structured populations	The population is given some spatial structure (usually independently of fitness), and mating is allowed to occur only between neighbours in this structure. Similarly, offspring replace low-fitness individual(s) within their own neighbourhood. Preserves diversity.	[152]
Island model GAs (Alba and Tomassini)	Several populations evolve on separate islands using locally panmictic mating. There is limited but occasional migration from one island to the other. Preserves diversity.	[54,153–155]
Landscape state machine tuning of directed evolution (LSM-DE)	In this technique, choices for mutation rate, population size, selection pressure and other evolutionary parameters are based on some prior sampling – and subsequent modelling – of the fitness landscape (or one believed to have similar topological features). Tunes the search algorithm specifically to the problem.	[90,156,157]
Fitness uniform selection scheme (FUSS)	FUSS is a selection scheme that preserves phenotypic diversity.	[158,159]
Hybrid local search or memetic algorithms	Evolution scheme in which selected individuals (usually fitter ones) in each generation are improved by performing a fitness-directed walk on the landscape. Improves exploitation of fit individuals, driving them towards optima. May not be feasible for some types of Directed Evolution experiment. These may not perform well when large populations but small numbers of generations are available (since the adaptive walks necessarily take the equivalent of several breeding generations to complete.)	[160,161]
Statistical Racing	Several evolutions, each with different parameters controlling selection pressure, mutation rates, and so on, are run simultaneously. At intervals, any evolution that is performing statistically significantly worse is dropped and its resources are allocated equally to the others. Expensive but effective.	[162,163]
Self-tuning evolutionary algorithms	Mutation rates, selection pressure, rates of recombination are controlled during evolution. These may be changed deterministically according to a schedule; changed according to some rules based on the progress being made; or actually evolved by making the parameters themselves subject to selection and variation.	[60]

doi:10.1371/journal.pone.0048862.t001

fact necessary to retain them to get to a fitter part of the landscape. In one computational example [19], more than one third of the non-neutral mutations (steps) in an evolutionary programme that were productive overall were via entities with fitness lower, sometimes much lower, than that of their parents. The intrinsic irreversibility of evolution [20] also makes this a significant issue. The high level of epistasis necessarily contingent upon the organisation of gene products into pathways is probably a major contributor to the comparative ineffectiveness of genome-wide association studies that look solely at individual genes [21–24], as well as the simple fact that most genes individually contribute very little to the fitness of complex traits overall [23,25–27], albeit that the right combinations of just a few can indeed do so [28].

While proteins link to pathways, and the enzymes encoding them may be on different chromosomes, in our discussions we essentially assume that we are dealing with one chromosome, as the main analysis of genotype-phenotype mapping cares little [29,30]. This may lead to a minor issue with respect to the effects or effectiveness of recombination, but the frequency of recombination per meiotic event is not in fact very great in most natural populations [31–35] and we shall ignore this.

There is also an assumption that, given our ability to generate knowledge of genetic sequences (and markers) at massively increasing rates (e.g. [36,37]), knowledge of parental genotypes should somehow better inform the way in which we carry out molecular breeding programmes, and there is some highly encouraging evidence for this based on ‘genomic selection’ [38–46]. Certainly there is increasing interest in and use of marker-assisted selection (MAS) in experimental breeding programmes generally (e.g. (Whittaker et al. 1995; Jones et al. 1997; Moreau et al. 2000; Kim and Park 2001; Dekkers and Hospital 2002; Koebner and Summers 2003; Dekkers 2004; Frary et al. 2005; Williams 2005; Xu et al. 2005; Eathington et al. 2007; Ribaut and Ragot 2007; Collard and Mackill 2008; Hospital 2009; Utomo and Linscombe 2009; Edwards and Batley 2010; Graham et al. 2010; Bancroft et al. 2011)), with particular interest lying in the use of dense genetic markers where available. Again, however, we know of very few published studies [4,47–49] in which the purpose was to compare the effectiveness of breeding programmes that have been performed with and without the use of genetic markers. This matters, because the continuing increase in sequencing speeds means that it is reasonable that before long we shall have the opportunity to acquire complete genome sequences of every

**Table 2.** Some evolutionary algorithms: G-algorithms.

ALGORITHM	COMMENTS	REFERENCES
Niching by genotypic fitness sharing	Fitness (reproductive opportunity) of individuals is shared amongst members of the same genotypic niche. Maintains diversity.	[130,149]
Learnable Evolution Model (LEM)	Uses classifiers to learn the genotypic basis of fitness during an evolutionary run; the inferred basis is used to alter selection to favour those with high predicted fitness, and disfavour those inferred to be deleterious.	[131,164]
Metamodel-assisted EAs	A regression model relating fitness to genotype is learned during evolution. The model is used to filter offspring individuals before they are evaluated (if their predicted fitness is low).	[78–81]
Efficient Global Optimisation (EGO, ParEGO)	A regression model of Gaussian process type is used to relate fitness to genotype (based on a sparse initial sampling of individuals). The model is globally searched to find the individual with the best “expected improvement” in fitness. This individual is then evaluated and used to update the model, and the process iterated.	[83,87,165]

doi:10.1371/journal.pone.0048862.t002

organism in a breeding population [50], i.e. knowledge of precisely where we are in the genetic landscape for each organism. It is reasonable that the much lowered costs of sequencing will be outweighed significantly by the knowledge that they bring. However, knowing how much we can benefit from this knowledge, and in particular how best to exploit it (in terms of the breeding algorithms), will be extremely important to the future of plant and animal breeding.

### *In silico* evolution – evolutionary computing

The large field of *in silico* evolution or evolutionary computing has remained surprisingly divorced from ‘real’ biological genetics (albeit that it has been of considerable value in helping our thinking about these landscapes, e.g. [51,52]; Stadler, 2002; Jones, 1995; Weinberger 1990). Meanwhile, it has developed a plethora of algorithms to search these fitness landscapes with great facility (e.g. [53–63]). Since the search spaces are typically astronomic, and the problems NP-hard [64,65], they seek (as heuristic methods (Pearl, 1984; [55,66,67], to find ‘good’ but not provably (globally) optimal solutions.

Evolutionary computing algorithms are based loosely on the principles of Darwinian evolution, involving the generation and analysis of variation, and selection on the basis of one or more measured fitnesses. Typically they are used to solve combinatorial optimisation problems (e.g. [55]) as well as nonconvex continuous problems such as parameter estimation [68] in which there are many possible configurations. The ‘search space’ is the number of possible solutions given the dimensionality of the problem as defined. If there are  $n$  variables (dimensions), each of which can take  $m$  distinguishable values, the number of possible combinations is evidently  $m^n$ , i.e. such problems scale exponentially with

the number of variables. While evolutionary algorithms come in many flavours, some of which we discuss below, in all cases they involve a population of (*in silico*) individuals each encoding a candidate solution to a problem of interest. Each of the members in the population has a fitness that is evaluated. Selection biased by fitness is used to determine which members of the population breed in a given ‘generation’, and offspring of these ‘parent’ individuals are created either by sexual recombination (after matching up of parents), or asexually by cloning a single parent. In either case, mutation is typically applied so that offspring differ from their parents. New individuals are evaluated for fitness, and then these replace (all or some) members of the parent population to form the next generation population. Evolution continues with rounds of mutation/recombination and selection until a desired endpoint is achieved (or a certain number of evaluations performed).

In Genetic Algorithms, both cloning and recombination are biased toward more fit individuals via the tournament selection used; however cloning tends to preserve the population genetic profile, whilst recombination creates more diversity. The (low) mutation rate on both cloned and recombined offspring provides more diversity and helps to prevent stagnation.

In the context of breeding, ‘cloning’, could represent the individual being retained for the study rather than culled; for plant studies ‘cloning’ may be reproduction via cuttings etc. It may be argued that the stochastic nature of the algorithms will mimic the real-world environmental factors that influence breeding experiments.

Much of the literature of evolutionary algorithms is concerned with optimising the nature of the search, typically couched in terms of a tension between ‘exploration’ (wide search to find

**Table 3.** Parameter settings used in the *F* and *G* algorithms.

Algorithm	Type	Key parameters
Breeder	<i>F</i>	$\lambda = 1000$ ; $p_m = 1/N$
Standard GA	<i>F</i>	$\lambda = 1000$ ; $p_m = 1/N$ ; $p_c = 0.7$ ; $t_{size} = 10$
Local mating	<i>F</i>	$\lambda = 1000$ ; $p_m = 1/N$ ; $p_c = 0.7$ ; $t_{size} = 10$
Niching	<i>G</i>	$\lambda = 1000$ ; $p_m = 1/N$ ; $p_c = 0.7$ ; $t_{size} = 10$ ; <i>niche radius</i> = dynamic with target niche count, $T_q = 5$
EARL1	<i>G</i>	$\lambda = 1000$ ; $p_m = 1/N$ ; $p_c = 0.7$ ; $t_{size} = 10$
EARL2	<i>G</i>	$\lambda = 1000$ ; $p_m = 1/N$ ; $p_c = 0.7$ ; $t_{size} = 10$

doi:10.1371/journal.pone.0048862.t003

**Table 4.** Tuneable Algorithm Parameters.

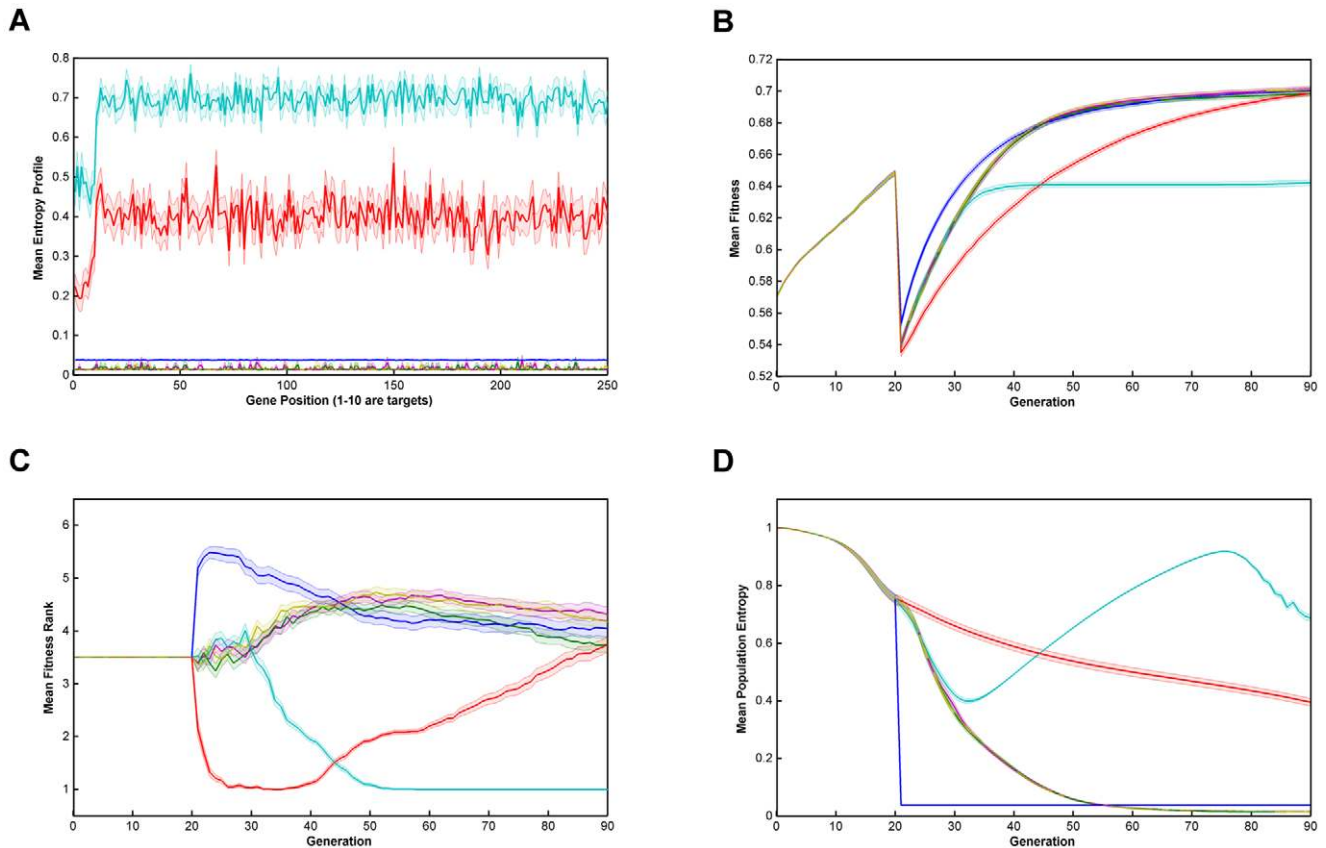
Computation Parameter	Corresponding Breeding Scenario
Population Size, P	Size of experimental population
Mutation Rate	Radiation dose
Cross-over rate	Self-pollination vs Cross-pollination?
Tournament size	Could be done directly via statistical sampling pool used to select breeders.
Niche Radius/Target No. of Niches	Could be done directly by statistical methods of measuring similarity.

doi:10.1371/journal.pone.0048862.t004

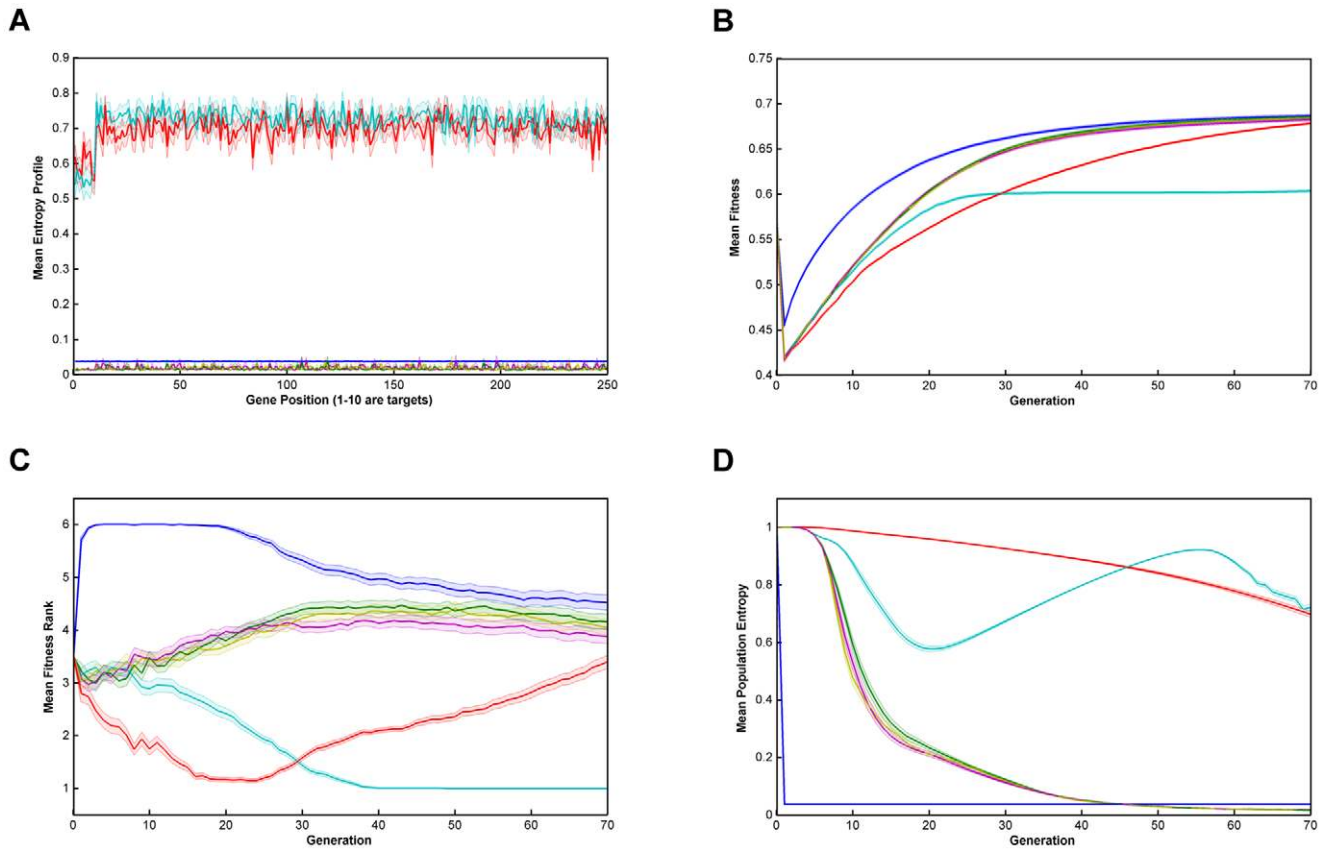
promising areas of the search space) and ‘exploitation’ (a narrower search to optimise within such areas), while seeking to avoid ‘premature convergence’ to a local optimum that may be far less good than the global (or other local) optimum.

For present purposes, the methods (algorithms) fall into two broad camps: those that at each generation know only the fitnesses of individuals, and those that also know where they are in the sequence/search space (and that thus at each generation increase our knowledge of the landscape). We shall refer to these as Fitness-only (F-) and Genomic (G-) methods. The first question then arises as to whether the kinds of algorithm available with G-methods typically outperform those available with F-methods, in other words whether the genomic knowledge buys you anything, and if so how best to exploit it. A second motivation for the present work

hinges on the fact that while the experimental breeding programmes are comparatively slow and costly, computational power is increasingly cheap, and should be exploited (as in projects such as the Robot Scientist [69–71], Robot Chromatographer [72] and experimental directed evolution [18]) to optimise the experimental design via ‘active learning’ (see also [73]). Indeed, one algorithm – known as the breeder genetic algorithm [74] – is based on the principles of classical experimental breeding and simply selects the fittest  $n$  individuals of a population and breeds with them. Clearly this is likely to lead to premature convergence to a local optimum that is much less fit than others possible. A final recognition is that while no algorithm is best for all landscapes [75] (there is ‘no free lunch’ [76,77]), some algorithms do indeed do considerably better in specific domains, especially if given some



**Figure 1. Relative performance of the six algorithms on the modified NK-landscapes where  $\lambda = 10000$ ,  $N = 250$ ,  $K = 5$ ,  $r = 10$  & burn-in = 20.** Note that there a discontinuity in fitness where the algorithm switches from ‘global fitness’,  $F$ , to trait fitness,  $F'$ . **Key:** **A.** Mean Entropy Profile at Final Generation; **B.** Mean Fitness; **C.** Mean Fitness Rank; **D.** Mean Entropy; — Breeder; — Standard GA; — Local Mating; — Niching; — EARL1; — EARL2; Error bands are  $\pm 1$  standard error =  $\pm \text{stdev}/\sqrt{n}$ . doi:10.1371/journal.pone.0048862.g001



**Figure 2. Relative performance of the six algorithms on the modified  $NK$ -landscapes where  $\lambda = 10000$ ,  $N = 250$ ,  $K = 5$ ,  $r = 10$  & no burn-in. Key: A. Mean Entropy Profile at Final Generation; B. Mean Fitness; C. Mean Fitness Rank; D. Mean Entropy; — Breeder; — Standard GA; — Local Mating; — Niching; — EARL1; — EARL2; Error bands are  $\pm 1$  standard error =  $\pm \text{stdev}/\sqrt{n}$ . doi:10.1371/journal.pone.0048862.g002**

knowledge of the landscape. Indeed, some G-algorithms (e.g. [78–81]) explicitly include iterative modelling (‘metamodelling’ [82–85]) of the landscapes themselves.

While much of the evolutionary computing literature has assumed access to large populations and generation numbers compared to the smaller populations and 70 generations we used, some methods have purposely focussed on small ones [86,87], and from the other side we have measured (experimentally) genotype-fitness data on more than 1 million samples (all 10-mers of nucleic acid aptamers [88]). Our discussion does therefore recognise the requirement to keep population sizes reasonably compatible with those likely to be accessible to experimenters.

The chief purposes of the present analysis, then, are (i) to bring to the attention of experimentalists interested in experimental breeding a knowledge of the literature of evolutionary computing, and (ii) to perform a study *in silico* to assess explicitly the kinds of benefits that can be had using knowledge of the genotype relative to those where only the (fitness of) the phenotype is known.

Table 3 illustrates possible correspondences between GA tuneable parameters and a real-world breeding scenario.

We have chosen uniform cross-over [89] in our genetic algorithms rather than tried to simulate real biology more closely. Whilst uniform crossover is reasonably efficient, the disadvantage is that it tends to preserve shorter schemata – however this is offset in that that it may be likely to explore the search-space more thoroughly. We believe this to be a more important consideration given the vast search-space in real genetics. We see no good reason to use other crossover methods since any simulated crossover

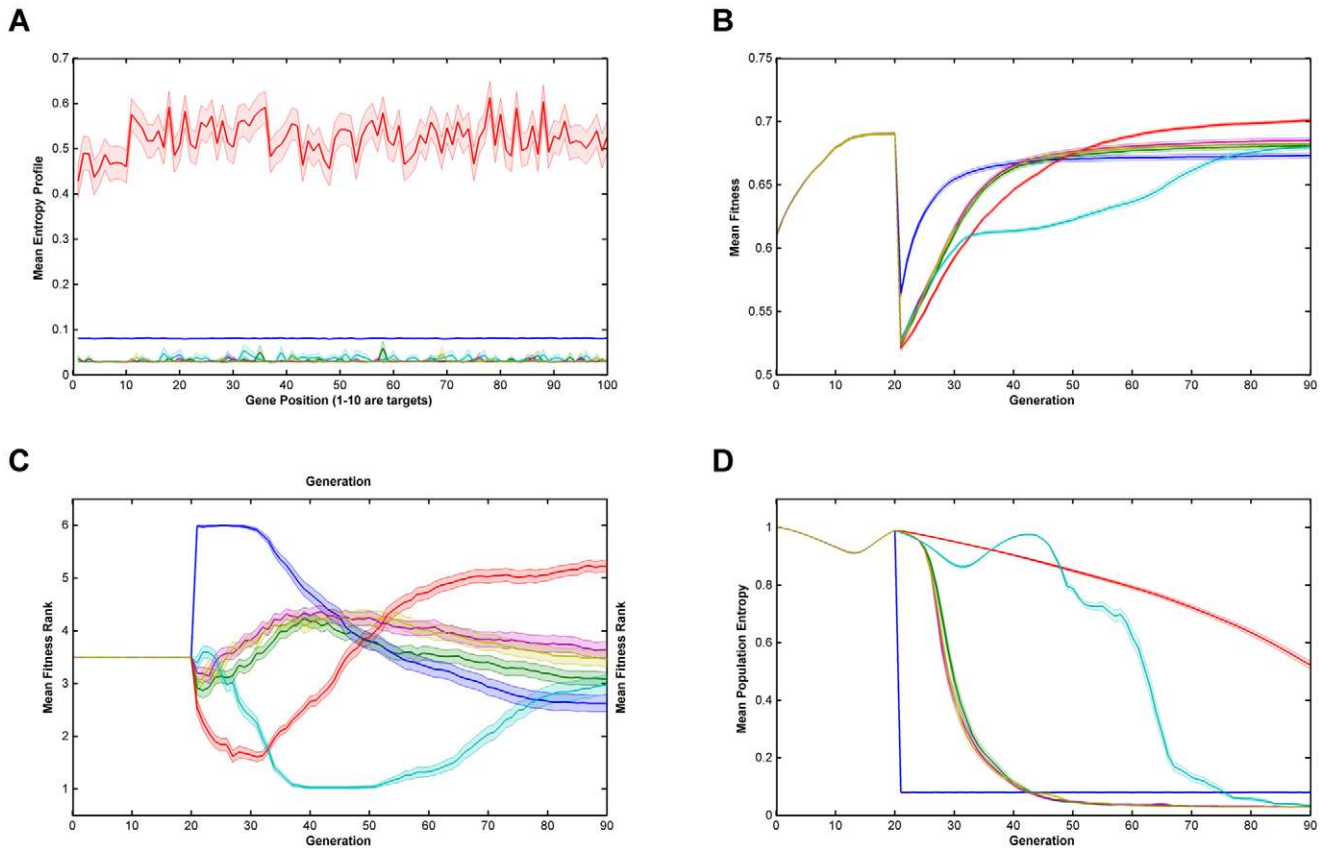
process that ‘breaks’ a conserved region of coupled genes will not influence the population as the corresponding individual will be automatically culled. In some sense, this means that the algorithm will ‘learn’ the real genetics.

## Experimental

### F- and G-algorithms

As a prelude to comparing F- and G-algorithms on the same landscapes, we here survey a selection of known algorithms of the two types. Tables 1 and 2 therefore summarises a number of the algorithms that have been proposed in the *in silico* literature, distinguishing those (F-algorithms) that know only the fitness (based on the ‘phenotype’) at each generation from those (G-algorithms) that also exploit knowledge of the position in the search space (the ‘genotype’).

There are essentially three ways in which an F-algorithm may be designed to improve over the basic Breeder Genetic Algorithm: (i) by better preserving diversity for longer, with the purpose of preventing convergence to a population that is located on a low fitness peak, and which has little ability to evolve further; (ii) by having mechanisms for tuning its own parameters adaptively, particularly the mutation and crossover rates; and, (iii) by using adaptive walks very near to the fitter ‘parent’ individuals to exploit these solutions more intensively (see Table 1; hybrid or memetic algorithms). Diversity preservation is by far the most important of these mechanisms in our case since (ii) and (iii) will usually demand large numbers of generations to be effective, whereas we are



**Figure 3. Relative performance of the six algorithms on the modified  $NK$ -landscapes where  $\lambda = 10000$ ,  $N = 100$ ,  $K = 5$ ,  $r = 10$  & burn-in = 20.** Note that there is a discontinuity in fitness where the algorithm switches from 'global fitness',  $F$ , to trait fitness,  $F'$ . **Key:** **A.** Mean Entropy Profile at Final Generation; **B.** Mean Fitness; **C.** Mean Fitness Rank; **D.** Mean Entropy; — Breeder; — Standard GA; — Local Mating; — Niching; — EARL1; — EARL2; Error bands are  $\pm 1$  standard error =  $\pm \text{stdev}/\sqrt{n}$ . doi:10.1371/journal.pone.0048862.g003

concerned with small generation numbers (see below for why). Although it is probable that most existing experimental breeding programmes do not in fact optimise (i), (ii) and (iii), we felt (a) that we should draw the attention of experimental breeders to this knowledge, and (b) we should not artificially 'load the dice' in favour of G-algorithms by using a poor F-algorithm as comparator. Thus we are likely to minimise the perceived benefits of G-algorithms over present practice.

G-algorithms may be able to achieve even more effective diversity preservation, as they have access to far more information, but the question is open. Moreover, G-algorithms have access to a further route to improvement not available to the F-algorithms: they may 'learn' or induce explicit statistical models of the sequence-fitness landscape, using this to select for breeding more accurately (see Learnable Evolution Model, metamodel-assisted EAs and EGO in Table 2, and the approach and data in [18]). It is also possible that G-algorithms could be designed to exploit the knowledge of the genotype-fitness map to effect *directed variation* (i.e. mutation and crossover) of individuals, not just to aid in more accurate selection for breeding. However, we exclude this avenue here as it assumes the ability to manipulate specific genes, which is an additional requirement separate from the ability to sequence (know about) the genotypes of individuals (though synthetic biology aims to make this a real possibility in future).

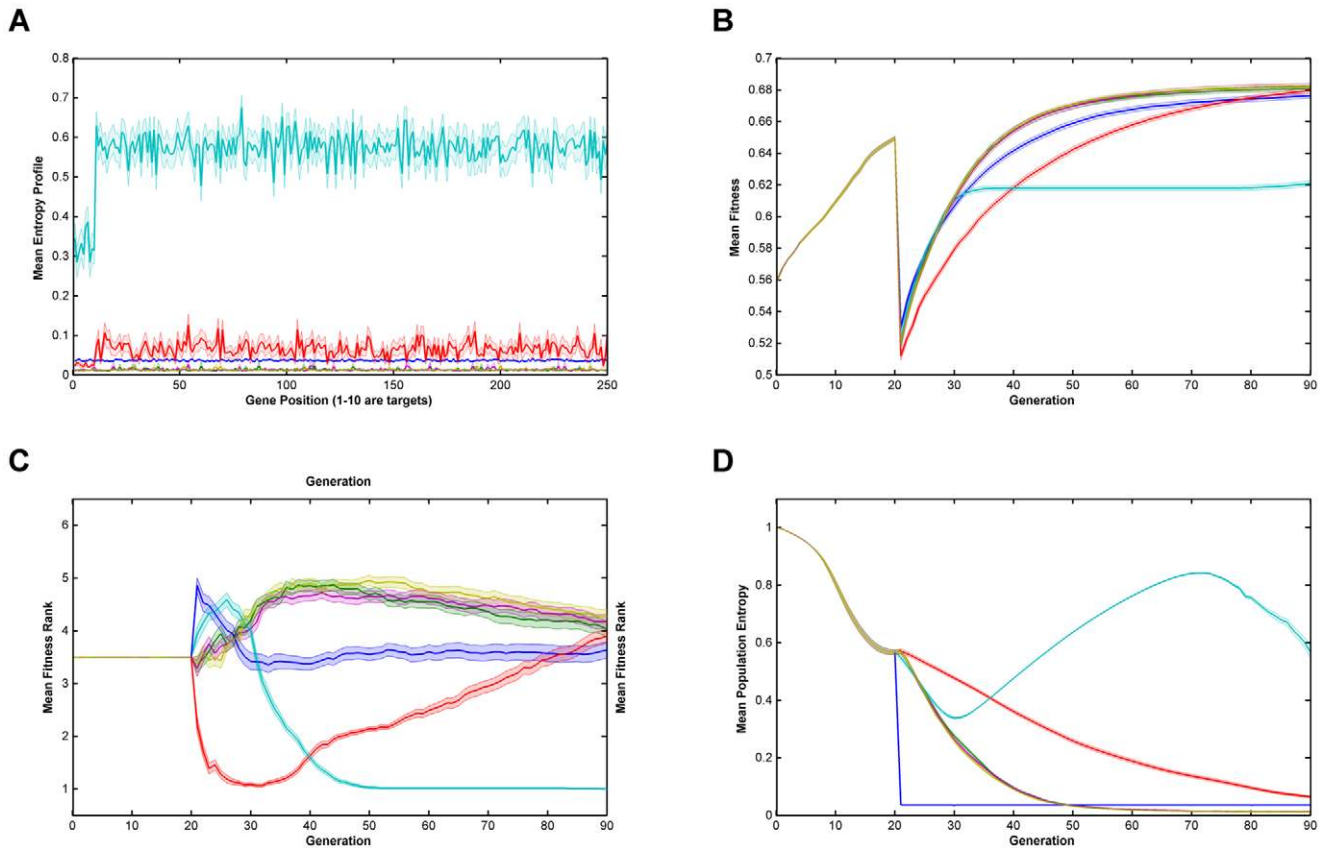
In our experimental study, we cannot hope to show a definitive advantage of one set of algorithms over the other on all problems (in fact that is a doomed proposition due to the No Free Lunch

Theorem [75,76]). Our aims are more illustrative. One aim is to compare the effectiveness (in terms of fitness improvement) of attempts to preserve diversity in an F-algorithm and a G-algorithm. The G-algorithm can explicitly measure and control genetic diversity, whereas an F-algorithm promotes diversity only by restricting mating. A second aim is to assess the relative effectiveness of G-algorithms that learn to model and exploit the fitness-sequence landscape, an approach borrowing from LEM (Table 2). These aims are reflected in the six algorithms that we use in our experiments, described in detail below.

The landscapes on which we evaluate these algorithms are described next.

### Landscapes

The choice of landscapes is more difficult, in that it is known that algorithms can be 'tuned' for specific landscapes. However, in this case we are purposely comparing different classes of algorithm on the same landscapes, and we choose landscapes of different character. The character of these landscapes is hard to define exactly, but here the concept of 'ruggedness' is important [52,90]. Ruggedness describes the likelihood that the fitness is well ('smooth') or poorly ('rugged') correlated with the genotypic distance from a particular starting point. In a very smooth landscape the fitness will decrease smoothly with distance, and the correlation will be good, whereas in a rugged landscape with many peaks and valleys the fitness-distance plot will be much more stochastic and the correlation lower.



**Figure 4. Relative performance of the six algorithms on the modified  $NK$ -landscapes where  $\lambda = 1000$ ,  $N = 250$ ,  $K = 5$ ,  $r = 10$  & burn-in = 20.** Note that there is a discontinuity in fitness where the algorithm switches from 'global fitness',  $F$ , to trait fitness,  $F'$ . **Key:** **A.** Mean Entropy Profile at Final Generation; **B.** Mean Fitness; **C.** Mean Fitness Rank; **D.** Mean Entropy; — Breeder; — Standard GA; — Local Mating; — Niching; — EARL1; — EARL2; Error bands are  $\pm 1$  standard error =  $\pm \text{stdev}/\sqrt{n}$ . doi:10.1371/journal.pone.0048862.g004

Because of the nonlinearity of enzyme kinetics, and the existence of feedback loops, real biochemical networks are highly nonlinear and epistatic (even if they occasionally appear linear/additive for small changes). Thus it is commonly the case that a change in one enzyme A or another enzyme B alone has little effect on a pathway (this follows from the systems properties of networks [91–93] and the evolution of biological robustness), but that changes in both have a major effect. This is then epistasis or synergy, and is very well established in both genetics and pharmacology (e.g. [94–108]). Epistasis is even observed within individual proteins (e.g. [109–115]). What landscapes should we then choose to model?

Despite the increasing availability of 'genome-scale' biochemical networks (e.g. [116,117]), these are mainly topological rather than kinetic and so it is not yet possible to model the detailed effects of multiple modulations on 'real' (*in silico*) biochemical networks, albeit that there are excellent examples where such models have proved useful in biotechnological optimisation [28,118–121]. For this reason, we have chosen to develop our analyses using 'artificial' landscapes with more or less known properties as a guide to the effectiveness (or otherwise) of adding knowledge of the genotype to the knowledge of the phenotype (fitness) during experimental optimisation/breeding programmes.

Whilst not directly modelled after "real" genetics, we believe that the chromosome representation used together with the  $NK$ -landscape-based fitness function mimic all the salient features of real genetics, this model is used for both F- and G-algorithms.

Whether the algorithm is F- or G- depends on the information obtained (phenotype/genetic information) and selection method used; these are entirely under the control of the experimenter in real breeding programs, therefore we expect no mechanism for unintentional bias towards F-algorithms, since selection is not coupled to the biological model.

### Modified $NK$ -landscape

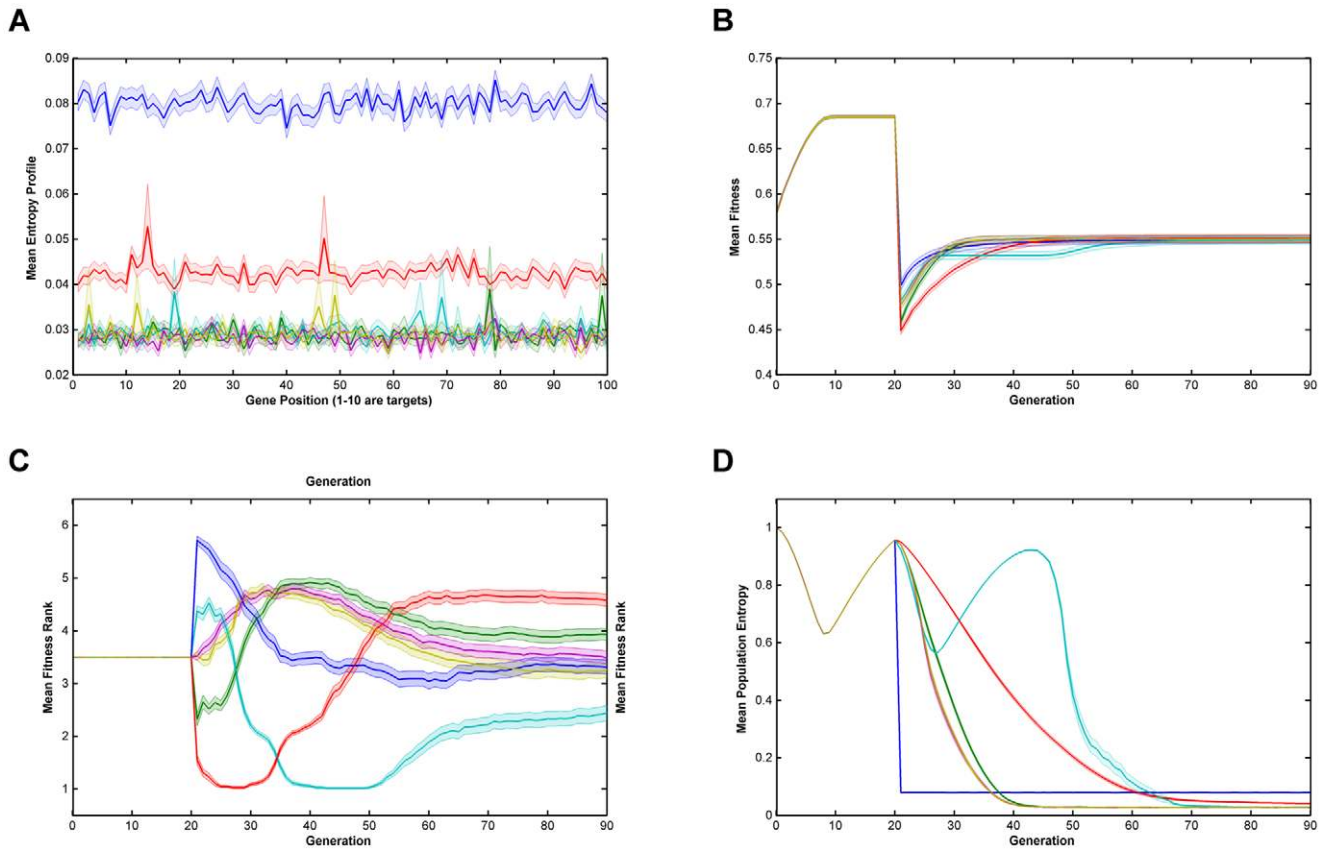
So-called  $NK$ -landscapes were developed by Kauffman [51,122].  $NK$ -landscapes are a class of synthetic landscapes that describe the epistatic relationships between genes in an organism and the consequent fitness of that organism. In the model genes are represented as bits in a binary string of length  $N$ . The total fitness  $F$  of a string (chromosome) is derived from the average fitness contribution of all the positions (genes):

$$F = \frac{1}{N} \sum_{i=1}^N f_i(\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,k+1}) \quad \alpha \in \{0,1\}^N$$

$\alpha_{i,j} \rightarrow$  elements of  $\alpha$  that contribute to  $f_i$

via mapping table of size,  $N(k+1)$

A chromosome,  $\alpha$ , is a binary vector of length  $N$ ,  $\alpha \in \{0,1\}^N$ . The fitness,  $f_i$ , of 'each position',  $i$ , is a function of a sub-vector of length  $k+1$ , which is constructed from a subset of elements drawn



**Figure 5. Relative performance of the six algorithms on the modified NK-landscapes where  $\lambda = 1000$ ,  $N = 100$ ,  $K = 1$ ,  $r = 10$  & burn-in = 20.** Note that there a discontinuity in fitness where the algorithm switches from 'global fitness',  $F$ , to trait fitness,  $F'$ . **Key:** **A.** Mean Entropy Profile at Final Generation; **B.** Mean Fitness; **C.** Mean Fitness Rank; **D.** Mean Entropy; — Breeder; — Standard GA; — Local Mating; — Niching; — EARL1; — EARL2; Error bands are  $\pm 1$  standard error =  $\pm \text{stdev}/\sqrt{n}$ . doi:10.1371/journal.pone.0048862.g005

from  $\alpha$  such that  $\alpha_{i,1} = \alpha_i$  and  $\alpha_{i,j}$  ( $j \neq i$ ) are selected uniformly at random without replacement from the elements of  $\alpha$ . Thus  $f_i$  has  $2^{k+1}$  states, and these are allocated at random in the range  $0 \rightarrow 1$ ,  $f_i \in [0,1]$ .

In the notation  $\alpha_{i,j}$ , ' $i,j$ ' contains the pseudo-subscript ' $j$ ' which represents position along the sub-vector of length  $(k+1)$ . The mapping of elements of the sub-vector to elements of  $\alpha$  is achieved via a look-up table of size  $N(k+1)$ .

In NK models, the  $N$  parameter controls the length of the chromosome and the corresponding search space, and the  $K$  parameter determines the degree of epistasis. It is not practical computationally to have  $N$  approach the number of genes found in organisms (yeast:  $\sim 6000$ , man:  $\sim 24000$ ), so one must choose a suitable compromise that nevertheless gives a sufficiently large search space to be comparable with real breeding populations, we have chosen:  $N = 1000 \rightarrow 2^{1000} \approx 10^{300}$ ,  $N = 10000 \rightarrow 2^{10000} \approx 2 \times 10^{3010}$ . The fitness of a gene depends on  $K$  other genes; increasing  $K$  increases the ruggedness of the landscape, for  $K=0$ , the fitness function generally has a trivial global optimum that is easy to locate. We have looked at  $K$  values in line with typical epistasis seen in real organisms.

NK-landscapes have been used extensively in the evolutionary computation literature as proxies for a variety of complex systems. It is rare that the exact properties of real biological fitness landscapes are known; however we can modify the properties of our models to the system we are studying more accurately, and the recent empirical studies on DNA binding support the view that NK

landscapes make a reasonable approximation to real biology [85,123].

Here, our aim is to use NK-landscapes to study the particular set of conditions faced by a human breeder trying to improve certain (quantifiable) traits. We do this by making a small modification to the basic NK model. In plants, as in animals and bacteria, certain genes will contribute more strongly to a particular trait than do others. When breeding plants, the aim is to optimize these traits whilst not comprising other qualities of the organism. In NK-landscapes the fitness contribution to each bit from all combination of  $k+1$  bits is assigned randomly. At low values of  $K$  the maximum contribution of individual bits may vary greatly across  $N$ , however as  $K$  increases this proportionality may be minimized. In modified NK-model we weight a portion of the chromosome ( $r$  bits) more highly by assessing these bits contribution to the overall fitness of the string.

$$F_r = \frac{1}{r} \sum_{i=1}^r f_i(\alpha_{i,1}, \alpha_{i,2}, \dots, \alpha_{i,k+1}) \alpha \in \{0,1\}^N$$

$\alpha_{i,j} \rightarrow$  elements of  $\alpha$  that contribute to  $f_i$

$$F' = \begin{cases} 0, & F < F_{th} \\ F \times F_r, & F \geq F_{th} \end{cases}$$



$F_i$  is multiplied by  $F$  to generate a new fitness value  $F$ . If  $F$  has a value below the fitness threshold,  $F_{th}$ ,  $F$  is reduced to 0; this is to simulate the effect of optimizing a trait at the expense of the overall fitness of the organism, and the catastrophic effect of certain deleterious mutations. The fitness threshold,  $F_{th}$ , was set at the minimum value (0.55) for which at least some fraction of the initial population had non-zero  $F$ .

## Assessment of Algorithms

Despite the wealth of EAs available our choices are constrained by the nature of the breeding problem. While advances in synthetic biology (e.g. [124–126]) may in time lead to the routine synthesis of bespoke genomes, this approach is currently not feasible. Many G algorithms (EDAs, LEM, proSAR) would require the complete synthesis of new chromosomes and so cannot be considered in this study.

We assess the performance of 6 algorithms on a group of modified  $NK$ -landscapes.

3 F algorithms:

- Breeder algorithm ( $1+\lambda$ ).
- Standard genetic algorithm.
- Local mating algorithm.

3 G algorithms:

- Niching genetic algorithm.
- Evolutionary algorithm with Rule-based Learning, type 1 (EARL1).
- Evolutionary algorithm with Rule-based Learning, type 2 (EARL2),

each of which we now describe. As noted, we have sought to tune each beforehand (via preliminary experiments that sought to optimise the mutation, crossover and selection pressure based on tournament size), so as not to give any one an obviously unfair advantage. In the Spirit of Open Science [127], the code is made freely available.

### Breeder algorithm ( $1+\lambda$ )

The breeder algorithm (Mühlenbein and Schlierkamp-Voosen 1993b) is an evolutionary algorithm that uses truncation selection, i.e. the best  $T\%$  of the offspring population is selected deterministically to become the parents of the next generation. In our version, we use a so-called ( $1+\lambda$ ) reproduction scheme: the offspring of a generation all have the same single parent, which is the fittest individual of the previous generation. The offspring are generated by cloning and mutation only. The algorithm has two free parameters,  $\lambda$ , the population size, and  $p_m$ , the per-bit mutation rate. The settings for both of these can be seen in Table 4. The algorithm represents a greedy (or strongly elitist) approach that is *a priori* unlikely to be competitive over larger numbers of generations, but may be expected to raise the population mean fitness very quickly. The initialization of the first-generation population used in this and all subsequent algorithms is the same, and is based on the niching GA described below. The procedure is detailed in a separate section below. Like all the algorithms described here, the algorithm terminates when a fixed, pre-ordained number of generations has been reached.

### Standard GA

The standard GA takes the form of a *generational* genetic algorithm, as described in [128]. The population size is  $\lambda$ . Each

generation,  $\lambda$  offspring individuals are produced, and these entirely replace the previous generation (even if less fit than the parents, i.e. the algorithm is non-elitist). Let  $P$  denote the current population and  $P'$  denote the population of the next generation. To produce one offspring individual for  $P'$ , a random variate is drawn to determine if the offspring is created by recombination of two parents, followed by mutation, or by cloning followed by mutation. The former occurs with probability  $p_c$ , the latter with probability  $1-p_c$ . For offspring produced by recombination, two parents are selected from  $P$ , with replacement, using *tournament selection* [53,59] with tournament size = 10 (based on preliminary experiments that suggested that this was optimal). Uniform crossover [89] is used. The resulting individual then undergoes mutation with a per-bit (i.e. per base) mutation rate of  $p_m$ . For an individual produced by cloning, just one parent is selected by tournament selection (from  $P$  with replacement), it is cloned, and the clone undergoes mutation with the same per-bit mutation rate of  $p_m$ .

### Local mating

The local mating algorithm follows principles set forth in [129]. The idea is for mating to occur only between individuals that are ‘geographically’ proximal to each other. Specifically, this structured population resists rapid take-over, preserving diversity for longer after a beneficial mutation or recombination event occurs in the population. Our standard generational GA is adapted so that each individual assumes a fixed location on a two-dimensional grid. To construct the next-generation population  $P'$ , each individual  $i$  in the current population  $P$  is taken in turn. If recombination of individual  $i$  occurs,  $i$  is recombined with the individual selected by an adapted tournament selection (described below) and the resulting offspring will replace  $i$  in the next generation population. If only mutation occurs, an individual found by the adapted tournament selection is cloned and mutated, and this will replace  $i$ . The tournament selection method is adapted so that given an individual  $i$ , the tournament returns the fittest individual from among the  $L\_size$  ( $= 10$ ) neighbours sampled at random (with replacement) from the 24 grid cells that surround  $i$  in a  $5 \times 5$  square.

### Niching Genetic Algorithm

The niching GA reduces the fitness of individuals that share a similar genotype with other individuals in the population, a concept known as genotypic fitness sharing [130]. The GA follows the standard one, except that the fitness associated with an individual in the population, which by definition determines its chances for reproduction via the tournament selection, is its shared fitness. The shared fitness is computed over the population by first counting, for each individual, the number of other individuals whose genotypes differ in ‘ $n_r$ ’ (the niche radius) or fewer genes; this number is referred to as the niche count  $n_c$ . The shared fitness is then  $f_{share} = 10f/(9+n_c)$ , where  $f$  is the normal fitness of the individual before sharing is applied (in our version of the algorithm, we have scaled  $f_{share}$  so that if the niche count is one,  $f_{share} = f$ ).

Simple versions of niching GAs use a fixed and arbitrary value of the niche radius,  $n_r$ ; we have adopted a dynamic niche radius scheme whereby the effective niche radius is adjusted at each iteration (generation) of the loop so as to approach a target number of niches,  $T_q$ . We estimate the number of niches,  $q$ , by use of a sampling strategy rather than evaluate the whole population, as the later would be computationally expensive  $O(n^2)$ . We chose  $T_q = 5$  based on preliminary runs of the niching GA.

## EARL1

Evolutionary Algorithm with Rule-based Learning (EARL1): Devised in the spirit of Michalski's LEM algorithm [131] and Llorà and Goldberg's "wise breeding" algorithm [132], EARL1 uses the statistical model AQ21 [133] which performs pattern discovery, generating inductive hypotheses. This is achieved by employing Attributional Calculus (<http://www.mli.gmu.edu/papers/2003-2004/mli04-2.pdf>), and produces attributional rules that capture strong regularities in the data, but may not be fully consistent or complete with regard to the training data (it is a 'fuzzy' algorithm). Each generation the AQ21 model is trained by selecting the top 20% (T-set) and bottom 20% (L-set) of individuals (in terms of fitness) from the current population. The AQ21 algorithm generates a vector of rules  $R$  (in the form attributes (loci) and attribute values), which are satisfied in the  $H$  set but not the  $L$  set.

When generating a new individual, a rule  $r_x$  is selected from  $R$ , as in LEM the rule is selected weighted by its abundance within the top  $T$  individuals within the population. Individuals from the current generation are evaluated in terms of whether they satisfy  $r_x$ , this subset,  $s$ , of individuals form a pool from which parents are selected.

Like a standard genetic algorithm, tournament selection (tournament size of 10) is used to select a parent; however, this parent can only be selected from  $s$ . When recombination is applied, tournament selection is used to select the second parent again from the subset of individuals that obey  $r_x$ ;  $s$ . The strategy aims to maintain rules within the next population and prevent the potentially destructive effects of recombination.

## EARL2

EARL2 presents an alternative hypothesis, that the loci that make up individual rules represent interesting points within the search space from which further improvements can be made. EARL2 has the same structure as EARL1, i.e. each new individual within a population is selected from a subset,  $s$ , which obeys a selected rule  $r_x$ . However when crossover is applied, the second parent is selected by tournament selection from the remainder of the individuals within the population which do not obey  $r_x$ .

This approach attempts to disrupt individual rules and may be particularly suited to highly non-linear landscapes, where moving to areas of higher fitness requires breaking of cooperative interaction between loci.

In both EARL1 and EARL2, where no rules are found by the AQ21 algorithm, selection defaults to tournament selection.

## Initialization Procedure (all algorithms)

In crop breeding (and in evolution generally), mutation rates are low [134] and beneficial mutations are rare, arguably reflecting the fact that plants have been evolving for many generations. The standard practice when using NK-landscapes is to generate the binary strings randomly in the first generation of an evolution. Mutating these strings will generate a substantial portion of beneficial mutations, which is unrepresentative of real biology systems. To capture the evolution of plants in a crop breeding programme more accurately, we implement some prior evolution before evaluating the performance of the various algorithms.

In order to simulate a 'burn-in' period or warm start to the evolution, i.e. to start with a genetically diverse population of moderately fit individuals that mimics the population one would see in a typical crop-breeding trial, we first run the Niching GA (see above) on the original (unmodified)  $NK$  landscape for a period of 0, 20 and 50 generations. The resulting populations are then the starting population for each of the six methods described above,

i.e. the starting population of each algorithm is paired (or matched) with the others. This procedure is repeated for each independently drawn  $NK$  landscape a total of 100 times, giving us 100 matched samples. We ran the algorithms for a further 70 generations after 'burn-in'.

## Evaluation of algorithms

In addition to plotting the mean fitness and mean population entropy of the repeated runs, we plotted the performance of the algorithms in terms of ranks. In each case (for each setting of  $N, K, r$ ) we measure the rank of an algorithm, on each of the 100 landscapes, as 1 if it has the lowest  $NKr$  fitness score of the  $C$  samples and,  $C$  if it has the highest fitness, where  $C$  is the number of algorithms (here, usually six), other ranks being assigned analogously, with ties being dealt with by assigning the mean rank to each of the tied results in the normal way [135]. For statistical robustness [136] we average these ranks over the 100 landscapes and plot these averages of ranks for each algorithm. This procedure averages the effect of the 100 independent landscapes, and allows us to see easily which algorithms have the best performance, especially where the absolute fitnesses are close together. (One might also argue that such regions indicate convergent performance for all practical purposes).

## Assessment of G-type algorithm performance

In the  $NKr$  landscapes,  $r$  loci have been artificial weighted in terms of their contribution to overall fitness. Accepting our rationale is correct, EARL1 should identify these  $r$  loci and quickly promote convergence on a consensus sequence, while EARL2 will maintain diversity across these positions. The Shannon Entropy  $H$  [137] is a measure of uncertainty or more precisely the minimum amount of information required to describe a discrete random variable  $X$  for a set of  $K$  observations  $x_1, \dots, x_k$ .

$$H = - \sum_i p(x_i) \log_2(p(x_i))$$

where  $p(x_i)$  is the probability mass function of  $x_i$

The algorithms were run up to generation 70 on the  $NKr$  landscapes beyond the 'burn-in' period. Shannon Entropy was measured for each position in the sequence as the mean of one hundred replicates. Convergence on a set of sub-strings across the chromosome will be evidenced by a consequent loss of entropy at these positions and thus demonstrate the mode of action of the two algorithms.

## Results and Discussion

From figures 1, 2, 3, 4 and 5 (and see supplementary information for the raw data), illustrating fitness and entropy, it can be seen the breeder algorithm displays inferior performance relative to the other evaluated algorithms, indicating that strong selection pressure without recombination is not conducive to optimisation on the modified  $NK$ -landscapes; whilst the fitness (apparently) rises rapidly, the entropy drop almost immediately to near zero, indicating that population diversity is low, conditions typically seen with premature convergence to a local maximum. On aggregate the model-based algorithms (EARL1 and EARL2) display very similar performance to the standard GA. Similarly the Niching algorithm (the other G-type algorithm assessed in this study) is much worse in terms of fitness for most of the early generations, it does however start to converge at generation  $\sim 70$

for  $N=100$ ; for  $N=250$  it reaches a plateau. The  $N$  parameter determines the gene length, and as the trait fitness is dependent on only 10 loci, any algorithm that preserves overall entropy will have a more detrimental effect on fitness the longer the gene length. Niching does conserve diversity for longer as indicated by the entropy. There appears to be a Diversity/fitness trade-off. The local mating algorithm is the F-type algorithm that also attempts to maintain population diversity; this too exhibits the same kind of trade-off.

The time and expense associated with breeding experiments necessitates the development of an individual with the desired phenotype within the fewest generations possible. The fitness penalty associated with maintaining diversity may therefore be too heavy a price to pay, and any of the EARL1, EARL2 or Standard GA may be preferable – however, EARL1 and EARL2 being more computationally expensive are slower to run an equivalent number of generations, and there appears to be little reason not to favour the simple GA algorithm.

We also looked at results for various values of the ‘ruggedness parameter’,  $k$  and the trait bit length,  $r$  ( $k=1,2,3,5$ ;  $r=1,5,10$ ). Whilst there was some inevitable shifting of the positions of the fitness and entropy curves, we did not see any evidence that there was any appreciable change in the relative performance of the algorithms overall. Similarly, runs with low population size (1000) compared to high population size (10000) showed similar behaviour, even though one might expect the higher population to provide a greater reservoir of genetic diversity.

On comparing the trait fitness of runs with no ‘burn-in’ to those with ‘burn-in’ set at 20 and 50 generations, we may gauge the effect of starting from an unfit breeding population compared to ‘well-bred’ population. Burn-in used the niching algorithm so as preserve population diversity. The results indicate that all the algorithms proceed with very similar fitness/entropy profiles (except for the translation of the starting point by virtue of the burn-in period used). Undoubtedly this is due to the trait fitness being very low whether or not any burn-in period is used; in general, the features that contribute to the global fitness will not contribute to the trait fitness chosen by the breeder. Additionally, we found that ‘burn-in’ had already essentially reached a fitness plateau at 20 generations, and that therefore runs for burn-in of 50 did not provide any additional information on algorithm performance.

The successful performance of genetic algorithms in optimisation is widely considered to be derived from their ability to identify a subset of strings with high sequence similarity in certain loci termed schema [128,138]. In the modified NK-landscapes  $r$  loci have been weighted to increase their contribution to the overall fitness of an individual. The ability of the algorithms to identify and optimise these loci will therefore dictate their performance. In figures are displayed the entropy across the loci after the first 70 generations of evolution. From the plots it can be seen the entropy of the first  $r$  loci is significantly different depending on the algorithms used and the  $N$  parameter chosen. After 70 generations the two entropy-preserving algorithms (niching and local mating) appear to have reduced entropy of the first  $r$  loci relative to the remainder of the genome. The remaining algorithms have flat or nearly flat and low entropies. This appears to be an inter-play between the natural trend of genetic algorithms to reduce the entropy in those loci that contain fitness information and the entropy-preserving qualities of the two algorithms. We see little evidence that any of the rule-based algorithms have ‘focused-in’ on relevant loci.

It might seem strange that rule-based approaches would not out-perform simple algorithms such as the Standard GA. Certainly, G-type approaches will still obey the principles of No

Free Lunch. For instance it has been previously observed that ProSAR [139] proves effective when addressing landscapes with low levels of epistasis and its bespoke role of optimization of protein sequences in directed evolution experiments, yet performances diminish relative to a standard F-type GA when levels of epistasis are increased [52,123]. This is arguably unsurprising, given that ProSAR is effectively a piecewise linear algorithm as it is based [140] on partial least squares [141]. Whilst EARL1 and EARL2 should not suffer from the disadvantage of being limited to linear problems (being based on the AQ21 attributional calculus learning module), it is by no means certain that the rule engine employed is suitable to the domain of application; the NKL landscapes we have used may not be amenable to description by such rule sets. In fact, even those runs with  $k=1$  (low ruggedness/epistasis) showed little or no improvement in the performance of EARL1 and EARL2 algorithms over the standard GA.

These data highlight the gains that can be achieved simply, both through the optimal choice of algorithm and through thoroughly tuning those algorithms to the landscape investigated, consistent with the No Free Lunch theorems alluded to above. We note that the comparative lack of benefit of G-algorithms here contrast somewhat e.g. with that observed in previously problems such as that studied using ParEGO [87]. However, that problem involved many fewer dimension (20) rather than the 1000 considered here, used a greater number of generations (250) than would be feasible in a ‘real’ breeding programme, and involved landscapes that were somewhat less rugged. Given the benefits of combining mutations in multiple known genes for pathway engineering [28,118–121], it seems likely that the greatest benefits of G-algorithms will be manifest when they are able to incorporate prior knowledge.

The benefit of tuning control parameters such as selection pressure and mutation rate is not a new concept to those practised in designing breeding programs. Inbreeding depression is well known as a consequence of over-selection in hermaphroditic crops. Similarly, the gains of increasing mutation rates to increase diversity in evolutionary search have become well known (e.g. [52,142,143]). Marker-assisted selection combined with backcrossing is a more direct method of improving phenotypic properties and bears similarities with algorithms that use machine learning to aid the evolutionary processes such as EARL1 and EARL2. The difference here is that we use knowledge of an entire *in silico* genome rather than just a few QTL markers for the breeding programme.

In marker-assisted selection (see above) the marker (usually a QTL, but increasingly a SNP) is used as an indicator of fitness for a desired phenotypic trait. In this approach, the phenotypic trait is assumed to be governed purely by the marker and not through cooperative interactions from elements on distal chromosome locations. The success of MAS is in some part reliant on how well the model reflects reality, and when the phenotypic trait is highly epistatic the model and consequently the evolution will tend to fail. Similarly, the success of evolutionary algorithms augmented with machine learning models will be highly reliant on how well the model reflects the true landscape.

The G-algorithms presented here were designed for applications in experimental breeding programmes, especially when we have sequences for all members of the breeding population of interest. To this end, they would appear to be a highly promising means of understanding the mapping between genotype and phenotype explicitly.

## Concluding Remarks

The increasing availability of genomic (sequence) knowledge, and our need to exploit it in experimental breeding programmes, points up the requirement for exploiting the best algorithms available. Those in use nowadays tend to exploit the methods of statistical genetics, but many more (and different) ones are known in the literature of evolutionary algorithms. We have here surveyed some of these approaches (and note that we do not include any consideration of epigenetics (e.g. [144])), finding that a well-tuned ‘simple’ GA can perform as effectively as some of the more sophisticated rule-based methods in these landscapes, that were not provided with any ‘prior knowledge’ of which genes might enjoy epistatic interactions. This has implications for experimental breeding programmes, especially when we can determine directly what we might wish to produce, for instance using the emerging methods of synthetic biology [126,145–147].

## Supporting Information

**File S1 Raw data for Fig 1.**  
(XLSX)

**File S2 Raw data for Fig 2.**  
(XLSX)

**File S3 Raw data for Fig 3.**  
(XLSX)

**File S4 Raw data for Fig 4.**  
(XLSX)

**File S5 Raw data for Fig 5.**  
(XLSX)

**File S6 zipped file containing non-proprietary code used.**  
(ZIP)

## Acknowledgments

DBK thanks Professors Darren Griffin (University of Kent) and Graham Moore (John Innes Centre), and in particular Professor Ian King (University of Nottingham) for useful discussions. We thank Dr Will Rowe for assistance with developing the rule-based algorithms.

## Author Contributions

Conceived and designed the experiments: SOH, JDK, DBK. Performed the experiments: SOH, JDK. Analyzed the data: SOH, JDK, DBK. Contributed reagents/materials/analysis tools: SOH, JDK. Wrote the paper: SOH, JDK, DBK.

## References

- Kell DB (2012) Scientific discovery as a combinatorial optimisation problem: how best to navigate the landscape of possible experiments? *Bioessays* 34: 236–244.
- Lippman Z, Tanksley SD (2001) Dissecting the genetic pathway to extreme fruit size in tomato using a cross between the small-fruited wild species *Lycopersicon pimpinellifolium* and *L. esculentum* var. giant heirloom. *Genetics* 158: 413–422.
- Hill WG (2005) A century of corn selection. *Science* 307: 683–684.
- Edgerton MD (2009) Increasing crop productivity to meet global needs for feed, food, and fuel. *Plant Physiol* 149: 7–13.
- Johansson AM, Pettersson ME, Siegel PB, Carlborg Ö (2010) Genome-wide effects of long-term divergent selection. *PLoS Genet* 6: e1001188.
- Wright S. The roles of mutation, inbreeding, crossbreeding and selection in evolution. In: Jones DF, editor; 1932; Ithaca, NY. Genetics Society of America, Austin TX. 356–366.
- Kell DB (2011) Breeding crop plants with deep roots: their role in sustainable carbon, nutrient and water sequestration. *Ann Bot* 108: 407–418.
- Kell DB (2012) Large-scale sequestration of carbon via plant roots in natural and agricultural ecosystems: why and how. *Phil Trans R Soc* 367: 1589–1597.
- Knowles J, Come D, Deb K, editors (2008) *Multiobjective Problem Solving from Nature*. Berlin: Springer.
- Beddington J (2010) Food security: contributions from science to a new and greener revolution. *Philos Trans R Soc Lond B Biol Sci* 365: 61–71.
- Fedoroff NV, Battisti DS, Beachy RN, Cooper PJ, Fischhoff DA, et al. (2010) Radically rethinking agriculture for the 21st century. *Science* 327: 833–834.
- Godfray HC, Beddington JR, Crute IR, Haddad L, Lawrence D, et al. (2010) Food Security: The Challenge of Feeding 9 Billion People. *Science* 327: 812–818.
- Godfray HCJ, Crute IR, Haddad L, Lawrence D, Muir JF, et al. (2010) The future of the global food system. *Philos Trans R Soc Lond B Biol Sci* 365: 2769–2777.
- Lal R (2010) Managing soils for a warming earth in a food-insecure and energy-starved world. *Journal of Plant Nutrition and Soil Science* 173: 4–15.
- Pretty J, Sutherland WJ, Ashby J, Auburn J, Baulcombe D, et al. (2010) The top 100 questions of importance to the future of global agriculture. *Int J Agric Sust* 8: 219–236.
- Tester M, Langridge P (2010) Breeding technologies to increase crop production in a changing world. *Science* 327: 818–822.
- Foresight (2011) *The Future of Food and Farming: final project report*. London: Government Office for Science.
- Knight CG, Platt M, Rowe W, Wedge DC, Khan F, et al. (2009) Array-based evolution of DNA aptamers allows modelling of an explicit sequence-fitness landscape. *Nucleic Acids Res* 37: e6.
- Lenski RE, Ofria C, Pennock RT, Adami C (2003) The evolutionary origin of complex features. *Nature* 423: 139–144.
- Bridgham JT, Ortlund EA, Thornton JW (2009) An epistatic ratchet constrains the direction of glucocorticoid receptor evolution. *Nature* 461: 515–519.
- Moore JH, Boczeko EM, Summar ML (2005) Connecting the dots between genes, biochemistry, and disease susceptibility: systems biology modeling in human genetics. *Mol Genet Metab* 84: 104–111.
- Moore JH, Williams SM (2005) Traversing the conceptual divide between biological and statistical epistasis: systems biology and a more modern synthesis. *Bioessays* 27: 637–646.
- Maher B (2008) The case of the missing heritability. *Nature* 456: 18–21.
- Moore JH, Asselbergs FW, Williams SM (2010) Bioinformatics challenges for genome-wide association studies. *Bioinformatics* 26: 445–455.
- Manolio TA, Collins FS, Cox NJ, Goldstein DB, Hindorf LA, et al. (2009) Finding the missing heritability of complex diseases. *Nature* 461: 747–753.
- Rockman MV, Skrovanek SS, Kruglyak L (2010) Selection at linked sites shapes heritable phenotypic variation in *C. elegans*. *Science* 330: 372–376.
- Yang J, Benyamin B, McEvoy BP, Gordon S, Henders AK, et al. (2010) Common SNPs explain a large proportion of the heritability for human height. *Nat Genet* 42: 565–569.
- Park JH, Lee KH, Kim TY, Lee SY (2007) Metabolic engineering of *Escherichia coli* for the production of L-valine based on transcriptome analysis and *in silico* gene knockout simulation. *Proc Natl Acad Sci U S A* 104: 7797–7802.
- Kell DB (2002) Genotype: phenotype mapping: genes as computer programs. *Trends Genet* 18: 555–559.
- Fisher P, Noyes H, Kemp S, Stevens R, Brass A (2009) A systematic strategy for the discovery of candidate genes responsible for phenotypic variation. *Methods Mol Biol* 573: 329–345.
- Gordo I, Charlesworth B (2001) Genetic linkage and molecular evolution. *Curr Biol* 11: R684–686.
- van Veen JE, Hawley RS (2003) Meiosis: when even two is a crowd. *Curr Biol* 13: R831–833.
- Hillers KJ (2004) Crossover interference. *Curr Biol* 14: R1036–1037.
- Chen JM, Cooper DN, Chuzhanova N, Ferec C, Patrinos GP (2007) Gene conversion: mechanisms, evolution and human disease. *Nat Rev Genet* 8: 762–775.
- Mancera E, Bourgon R, Brozzi A, Huber W, Steinmetz LM (2008) High-resolution mapping of meiotic crossovers and non-crossovers in yeast. *Nature* 454: 479–485.
- Shendure J, Ji H (2008) Next-generation DNA sequencing. *Nat Biotechnol* 26: 1135–1145.
- Metzker ML (2010) Sequencing technologies – the next generation. *Nat Rev Genet* 11: 31–46.
- Meuwissen TH, Hayes BJ, Goddard ME (2001) Prediction of total genetic value using genome-wide dense marker maps. *Genetics* 157: 1819–1829.
- Gianola D, Perez-Enciso M, Toro MA (2003) On marker-assisted prediction of genetic value: beyond the ridge. *Genetics* 163: 347–365.
- Xu S (2003) Estimating polygenic effects using markers of the entire genome. *Genetics* 163: 789–801.
- Schaeffer LR (2006) Strategy for applying genome-wide selection in dairy cattle. *Journal of Animal Breeding and Genetics* 123: 218–223.
- Goddard ME, Hayes BJ (2007) Genomic selection. *J Anim Breed Genet* 124: 323–330.

43. Luan T, Woolliams JA, Lien S, Kent M, Svendsen M, et al. (2009) The accuracy of Genomic Selection in Norwegian red cattle assessed by cross-validation. *Genetics* 183: 1119–1126.
44. Odgård J, Yazdi MH, Sonesson AK, Meuwissen TH (2009) Incorporating desirable genetic characteristics from an inferior into a superior population using genomic selection. *Genetics* 181: 737–745.
45. Fahrenkrug SC, Blake A, Carlson DF, Doran T, Van Eenennaam A, et al. (2010) Precision genetics for complex objectives in animal agriculture. *J Anim Sci* 88: 2530–2539.
46. Maenhout S, De Baets B, Haesaert G (2010) Graph-based data selection for the construction of genomic prediction models. *Genetics* 185: 1463–1475.
47. Moreau L, Lemarie S, Charcosset A, Gallais A (2000) Economic efficiency of one cycle of marker-assisted selection. *Crop Sci*, 40: 329–337.
48. Eathington SR, Crosbie TM, Edwards MD, Reiter RS, Bull JK (2007) Molecular markers in a commercial breeding program *Crop Sci* 47: S154–S163.
49. Kean S (2010) Besting Johnny Appleseed. *Science* 328: 301–303.
50. Meuwissen T, Goddard M (2010) Accurate prediction of genetic values for complex traits by whole-genome resequencing. *Genetics* 185: 623–631.
51. Kauffman SA (1993) *The origins of order*. Oxford: Oxford University Press.
52. Wedge D, Rowe W, Kell DB, Knowles J (2009) *In silico* modelling of directed evolution: implications for experimental design and stepwise evolution. *J Theor Biol* 257: 131–141.
53. Bäck T, Fogel DB, Michalewicz Z, editors (1997) *Handbook of evolutionary computation*. Oxford: IOP Publishing/Oxford University Press.
54. Haupt RL, Haupt SE (1998) *Practical Genetic Algorithms*. New York: Wiley.
55. Corne D, Dorigo M, Glover F, editors (1999) *New ideas in optimization*. London: McGraw Hill.
56. Zitzler E (1999) *Evolutionary algorithms for multiobjective optimization: methods and applications*. Aachen: Shaker Verlag.
57. Deb K (2001) *Multi-objective optimization using evolutionary algorithms*. New York: Wiley.
58. Coello Coello CA, van Veldhuizen DA, Lamont GB (2002) *Evolutionary algorithms for solving multi-objective problems*. New York: Kluwer Academic Publishers.
59. Reeves CR, Rowe JE (2002) *Genetic algorithms – principles and perspectives: a guide to GA theory*. Dordrecht: Kluwer Academic Publishers.
60. Eiben AE, Smith JE (2003) *Introduction to evolutionary computing*. Berlin: Springer.
61. Bongard JC, Lipson H (2005) Nonlinear system identification using coevolution of models and tests. *IEEE Trans Evol Comput* 9: 361–384.
62. Bongard J, Lipson H (2007) Automated reverse engineering of nonlinear dynamical systems. *Proc Natl Acad Sci U S A* 104: 9943–9948.
63. Knowles J, Corne D, Deb K, editors (2008) *Multiobjective problem solving from nature: from concepts to applications*. Heidelberg: Springer.
64. Garey M, Johnson D (1979) *Computers and intractability: a guide to the theory of NP-completeness*. San Francisco: Freeman.
65. Pierce NA, Winfree E (2002) Protein design is NP-hard. *Protein Eng* 15: 779–782.
66. Dasgupta P, Chakrabarti PP, DeSarkar SC (1999) *Multiobjective heuristic search*. Braunschweig: Vieweg.
67. Michalewicz Z, Fogel DB (2000) *How to solve it: modern heuristics*. Heidelberg: Springer-Verlag.
68. Mendes P, Kell DB (1998) Non-linear optimization of biochemical pathways: applications to metabolic engineering and parameter estimation. *Bioinformatics* 14: 869–883.
69. King RD, Whelan KE, Jones FM, Reiser PGK, Bryant CH, et al. (2004) Functional genomic hypothesis generation and experimentation by a robot scientist. *Nature* 427: 247–252.
70. King RD, Rowland J, Oliver SG, Young M, Aubrey W, et al. (2009) The automation of science. *Science* 324: 85–89.
71. King RD, Rowland J, Aubrey W, Liakata M, Markham M, et al. (2009) *The Robot Scientist Adam*. *Computer* 42: 46–54.
72. O'Hagan S, Dunn WB, Brown M, Knowles JD, Kell DB (2005) Closed-loop, multiobjective optimisation of analytical instrumentation: gas-chromatography-time-of-flight mass spectrometry of the metabolomes of human serum and of yeast fermentations. *Anal Chem* 77: 290–303.
73. Knowles J (2009) Closed-Loop Evolutionary Multiobjective Optimization. *IEEE Computational Intelligence Magazine* 4: 77–91.
74. Mühlenbein H, Schlierkamp-Voosen D (1993) Predictive models for the breeder genetic algorithm. 1. Continuous parameter optimization. *Evolutionary Computation* 1: 25–49.
75. Radcliffe NJ, Surry PD (1995) Fundamental limitations on search algorithms: evolutionary computing in perspective. *Computer Science Today* 1995: 275–291.
76. Wolpert DH, Macready WG (1997) No Free Lunch theorems for optimization. *IEEE Trans Evol Comput* 1: 67–82.
77. Corne D, Knowles J (2003) No free lunch and free leftovers theorems for multiobjective optimisation problems. In: Fonseca C, et al., editor. *Evolutionary Multi-criterion Optimization (EMO 2003)*, LNCS 2632. Berlin: Springer. 327–341.
78. Emmerich M, Giotis A, Ozdemir M, Back T, Giannakoglou K (2002) Metamodel-assisted evolution strategies. *Parallel Problem Solving from Nature-PPSN VII: 7th International Conference*, Granada, Spain, September 7–11, 2002: Proceedings: 361.
79. Jin YC, Olhofer M, Sendhoff B (2002) A framework for evolutionary optimization with approximate fitness functions. *IEEE Trans Evol Comput* 6: 481–494.
80. Jin Y (2005) *A comprehensive survey of fitness approximation in evolutionary computation*. *Soft Computing – A Fusion of Foundations, Methodologies and Applications* 9: 3–12.
81. Knowles J, Nakayama H (2008) *Meta-Modeling in Multiobjective Optimization*. *Multiobjective Optimization: Interactive and Evolutionary Approaches* 5252: 245–284.
82. Crary SB (2002) Design of computer experiments for metamodel generation. *Analog Integr Circ Sig Proc* 32: 7–16.
83. Sasena MJ, Papalambros P, Goovaerts P (2002) Exploration of metamodeling sampling criteria for constrained global optimization. *Engineering Optimization* 34: 263–278.
84. Chen VCP, Tsui KL, Barton RR, Meckesheimer M (2006) A review on design, modeling and applications of computer experiments. *IEE Transactions* 38: 273–291.
85. Rowe W, Wedge DC, Platt M, Kell DB, Knowles J (2010) Predictive models for population performance on real biological fitness landscapes. *Bioinformatics* 26: 2125–2142.
86. Knowles JD, Hughes EJ (2005) Multiobjective optimization on a budget of 250 evaluations. *Evolutionary Multi-Criterion Optimization (EMO 2005)*, LNCS 3410, 176–190. <http://dbkchumistacuk/knowles/pubshhtml>.
87. Knowles J (2006) ParEGO: A hybrid algorithm with on-line landscape approximation for expensive multiobjective optimization problems. *IEEE Trans Evol Comput* 10: 50–66.
88. Rowe W, Platt M, Wedge D, Day PJ, Kell DB, et al. (2010) Analysis of a complete DNA-protein affinity landscape. *J R Soc Interface* 7: 397–408.
89. Syswerda G. Uniform crossover in genetic algorithms. In: Schaffer J, editor; 1989. *Morgan Kaufmann*. 2–9.
90. Wedge D, Kell DB. Rapid prediction of optimum population size in genetic programming using a novel genotype – fitness correlation. In: Ryan C, Keizer M, editors; 2008; Atlanta, GA. *ACM*. 1315–1322.
91. Kell DB, Westerhoff HV (1986) Metabolic control theory: its role in microbiology and biotechnology. *FEMS Microbiol Rev* 39: 305–320.
92. Fell DA (1996) *Understanding the control of metabolism*. London: Portland Press.
93. Heinrich R, Schuster S (1996) *The regulation of cellular systems*. New York: Chapman & Hall.
94. Borisy AA, Elliott PJ, Hurst NW, Lee MS, Lehar J, et al. (2003) Systematic discovery of multicomponent therapeutics. *Proc Natl Acad Sci U S A* 100: 7977–7982.
95. Fan QW, Specht KM, Zhang C, Goldenberg DD, Shokat KM, et al. (2003) Combinatorial efficacy achieved through two-point blockade within a signaling pathway—a chemical genetic approach. *Cancer Res* 63: 8930–8938.
96. Ihekwaba AEC, Broomhead DS, Grimley R, Benson N, White MRH, et al. (2005) Synergistic control of oscillations in the NF- $\kappa$ B signalling pathway. *IEE Systems Biology* 152: 153–160.
97. Keith CT, Borisy AA, Stockwell BR (2005) Multicomponent therapeutics for networked systems. *Nat Rev Drug Discov* 4: 71–78.
98. Hopkins AL, Mason JS, Overington JP (2006) Can we rationally design promiscuous drugs? *Curr Opin Struct Biol* 16: 127–136.
99. Csermely P, Agoston V, Pongor S (2005) The efficiency of multi-target drugs: the network approach might help drug design. *Trends Pharmacol Sci* 26: 178–182.
100. Kell DB (2006) Metabolomics, modelling and machine learning in systems biology: towards an understanding of the languages of cells. The 2005 Theodor Bücher lecture. *FEBS J* 273: 873–894.
101. Kell DB (2006) Systems biology, metabolic modelling and metabolomics in drug discovery and development. *Drug Disc Today* 11: 1085–1092.
102. Lehar J, Zimmermann GR, Krueger AS, Molnar RA, Ledell JT, et al. (2007) Chemical combination effects predict connectivity in biological systems. *Mol Syst Biol* 3: 80.
103. Zimmermann GR, Lehar J, Keith CT (2007) Multi-target therapeutics: when the whole is greater than the sum of the parts. *Drug Discov Today* 12: 34–42.
104. Hopkins AL (2008) Network pharmacology: the next paradigm in drug discovery. *Nat Chem Biol* 4: 682–690.
105. Lehar J, Krueger A, Zimmermann G, Borisy A (2008) High-order combination effects and biological robustness. *Molecular Systems Biology* 4: 215.
106. Lehar J, Stockwell BR, Giaever G, Nislow C (2008) Combination chemical genetics. *Nat Chem Biol* 4: 674–681.
107. Jia J, Zhu F, Ma X, Cao ZW, Li YX, et al. (2009) Mechanisms of drug combinations: interaction and network perspectives. *Nat Rev Drug Discov* 8: 111–128.
108. Costanzo M, Baryshnikova A, Bellay J, Kim Y, Spear ED, et al. (2010) The genetic landscape of a cell. *Science* 327: 425–431.
109. Pritchard L, Dufton MJ (2000) Do proteins learn to evolve? The Hopfield network as a basis for the understanding of protein evolution. *J Theoret Biol* 202: 77–86.
110. Aita T, Hamamatsu N, Nomiya Y, Uchiyama H, Shibana Y, et al. (2002) Surveying a local fitness landscape of a protein with epistatic sites for the study of directed evolution. *Biopolymers* 64: 95–105.

111. Bershtein S, Segal M, Bekerman R, Tokuriki N, Tawfik DS (2006) Robustness-epistasis link shapes the fitness landscape of a randomly drifting protein. *Nature* 444: 929–932.
112. Hayashi Y, Aita T, Toyota H, Husimi Y, Urabe I, et al. (2006) Experimental rugged fitness landscape in protein sequence space. *PLoS One* 1: e96.
113. Bloom JD, Arnold FH, Wilke CO (2007) Breaking proteins with mutations: threads and thresholds in evolution. *Mol Syst Biol* 3: 76.
114. Romero PA, Arnold FH (2009) Exploring protein fitness landscapes by directed evolution. *Nat Rev Mol Cell Biol* 10: 866–876.
115. Östman B, Hintze A, Adami C (2011) Impact of epistasis and pleiotropy on evolutionary adaptation. *Proc Roy Soc B* 279: 247–256.
116. Herrgård MJ, Swainston N, Dobson P, Dunn WB, Arga KY, et al. (2008) A consensus yeast metabolic network obtained from a community approach to systems biology. *Nature Biotechnol* 26: 1155–1160.
117. Thiele I, Palsson BO (2010) A protocol for generating a high-quality genome-scale metabolic reconstruction. *Nat Protoc* 5: 93–121.
118. Patil KR, Rocha I, Förster J, Nielsen J (2005) Evolutionary programming as a platform for *in silico* metabolic engineering. *BMC Bioinformatics* 6: 308.
119. Park JH, Lee SY, Kim TY, Kim HU (2008) Application of systems biology for bioprocess development. *Trends Biotechnol* 26: 404–412.
120. Becker J, Zelder O, Häfner S, Schröder H, Wittmann C (2011) From zero to hero—design-based systems metabolic engineering of *Corynebacterium glutamicum* for L-lysine production. *Metab Eng* 13: 159–168.
121. Yim H, Haselbeck R, Niu W, Pujol-Baxley C, Burgard A, et al. (2011) Metabolic engineering of *Escherichia coli* for direct production of 1,4-butanediol. *Nat Chem Biol* 7: 445–452.
122. Kauffman SA, Weinberger ED (1989) The NK model of rugged fitness landscapes and its application to maturation of the immune response. *J Theor Biol* 141: 211–245.
123. Fox R, Roy A, Govindarajan S, Minshull J, Gustafsson C, et al. (2003) Optimizing the search algorithm for protein engineering by directed evolution. *Protein Eng* 16: 589–597.
124. Heinemann M, Panke S (2006) Synthetic biology—putting engineering into biology. *Bioinformatics* 22: 2790–2799.
125. Leonard E, Nielsen D, Solomon K, Prather KJ (2008) Engineering microbes with synthetic biology frameworks. *Trends Biotechnol* 26: 674–681.
126. Baker M (2011) The next step for the synthetic genome. *Nature* 473: 403, 405–408.
127. Merali Z (2010)...Error. *Nature* 467: 775–777.
128. Holland JH (1992) *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*: MIT Press.
129. Collins RJ, Jefferson DR (1991) Selection in massively parallel genetic algorithms. In: Belew RK, Booker LB, editors. *Morgan Kaufmann*. 249–256.
130. Goldberg DE, Richardson J (1987) Genetic algorithms with *sharing* for multimodal function optimization. *Lawrence Erlbaum*. 41–49.
131. Michalski RS (2000) Learnable evolution model: Evolutionary processes guided by machine learning. *Machine Learning* 38: 9–40.
132. Llorà X, Goldberg DE (2003) Wise Breeding GA via Machine Learning Techniques for Function Optimization. *Proc GECCO*: 1172–1183.
133. Wojtusiak J, Michalski RS, Kaufman KA, Pietrzykowski J (2006) The AQ21 natural induction program for pattern discovery: initial version and its novel features. *Proc International Conference on Tools with Artificial Intelligence (ICTAI'06)* 523–526.
134. Ossowski S, Schneeberger K, Lucas-Lledo JJ, Warthmann N, Clark RM, et al. (2010) The rate and molecular spectrum of spontaneous mutations in *Arabidopsis thaliana*. *Science* 327: 92–94.
135. Conover WJ (1980) *Practical Nonparametric Statistics*. New York: Wiley.
136. Broadhurst D, Kell DB (2006) Statistical strategies for avoiding false discoveries in metabolomics and related experiments. *Metabolomics* 2: 171–196.
137. Shannon CE, Weaver W (1949) *The mathematical theory of communication*. Urbana, IL: University of Illinois Press.
138. Goldberg DE (1989) *Genetic algorithms in search, optimization and machine learning*: Addison-Wesley.
139. Fox RJ, Davis SC, Mundorff EC, Newman LM, Gavrilovic V, et al. (2007) Improving catalytic function by ProSAR-driven enzyme evolution. *Nat Biotechnol* 25: 338–344.
140. Fox R (2005) Directed molecular evolution by machine learning and the influence of nonlinear interactions. *J Theor Biol* 234: 187–199.
141. Wold S, Trygg J, Berglund A, Antti H (2001) Some recent developments in PLS modeling. *Chemometr Intell Lab Syst* 58: 131–150.
142. Yonezawa K, Yamagata H (1977) Optimum mutation rate and optimum dose for practical mutation breeding. *Euphytica* 26: 413–426.
143. Zaccolo M, Gherardi E (1999) The effect of high-frequency random mutagenesis on *in vitro* protein evolution: A study on TEM-1  $\beta$ -lactamase. *J Mol Biol* 285: 775–783.
144. Feinberg AP, Tycko B (2004) The history of cancer epigenetics. *Nat Rev Cancer* 4: 143–153.
145. Barrett CL, Kim TY, Kim HU, Palsson BØ, Lee SY (2006) Systems biology as a foundation for genome-scale synthetic biology. *Curr Opin Biotechnol* 17: 488–492.
146. Andrianantoandro E, Basu S, Karig DK, Weiss R (2006) Synthetic biology: new engineering rules for an emerging discipline. *Mol Syst Biol* 2: 2006 0028.
147. Purnick PE, Weiss R (2009) The second wave of synthetic biology: from modules to systems. *Nat Rev Mol Cell Biol* 10: 410–422.
148. Mühlenbein H, Schlierkamp-Voosen D (1993) The science of breeding and its application to the breeder genetic algorithm (BGA). *Evolutionary Computation* 1: 335–360.
149. Horn J, Nafpliotis N. Multiobjective optimisation using the niched Pareto genetic algorithm; 1994; Piscataway. 82–87.
150. Mahfoud SW (1995) *Niching methods for genetic algorithms*. PhD thesis, University of Illinois at Urbana-Champaign IlliGAL Report 95001.
151. Mengshoel OJ, Goldberg DE (2008) The crowding approach to niching in genetic algorithms. *Evol Comput* 16: 315–354.
152. Tomassini M (2005) *Spatially structured evolutionary algorithms*. Heidelberg: Springer.
153. Schlierkamp-Voosen D, Mühlenbein H (1994) Strategy adaptation by competing subpopulations. *Parallel Problem Solving from Nature (PPSN III)*: 199–208.
154. Wineberg M, Chen J (2004) The shifting balance genetic algorithm as more than just another island model GA. *Proc Genet Evol Comput Conf (GECCO 2004)*: 318–329.
155. Skolicki Z, de Jong K (2005) The influence of migration sizes and intervals on island models. *Proc Genetic Evol Comput Conf (GECCO 2005)*: 1295–1302.
156. Corne DW, Oates MJ, Kell DB (2003) *Landscape State Machines: tools for evolutionary algorithm performance analyses and landscape/algorithm mapping*. In: Cagnoni S, Cardalda JRJ, Corne DW, Gottlieb J, Guillot A, et al., editors. *EvoWorkshops 2003*. Berlin: Springer. 187–198.
157. Rowe W, Corne DW, Knowles J (2006) Predicting Stochastic Search Algorithm Performance using Landscape State Machines. *IEEE Congress on Evolutionary Computation (CEC 2006)*: 9849–9856.
158. Hutter M (2001) Fitness uniform selection to preserve genetic diversity. Technical Report IDSIA-01-01 <http://ftp.idsia.ch/pub/techrep/IDSIA-01-01psgz>.
159. Hutter M, Legg S (2006) Fitness Uniform Optimization. *IEEE Trans Evol Comput* 10: 568–589.
160. Moscato P (1999) Memetic algorithms: a short introduction. In: Corne D, Dorigo M, Glover F, editors. *New Ideas in Optimisation*. London: McGraw-Hill. 219–243.
161. Knowles JD, Corne DW (2004) Memetic algorithms for multiobjective optimization: issues, methods and prospects. In: Krasnogor N, Smith JE, Hart WE, editors. *Recent Advances in Memetic Algorithms*. Berlin: Springer. 313–352.
162. Birattari M, Stützle T, Paquete L, Varrentrap K (2002) A racing algorithm for configuring metaheuristics. *Proc Genet Evol Comput Conf (GECCO 2002)*: 11–18.
163. Yuan B, Gallagher M (2004) Statistical racing techniques for improved empirical evaluation of evolutionary algorithms 81. *Proc 8th Int Conf on Parallel Problem Solving from Nature LNCS 3242*.
164. Jourdan L, Corne D, Savic DA, Walters GA (2005) Preliminary investigation of the ‘Learnable Evolution Model’ for faster/better multiobjective water systems design. *Proc Conf on Evolutionary Multiobjective Optimisation 2005*: 841–855.
165. Jones DR, Schonlau M, Welch WJ (1998) Efficient global optimization of expensive black-box functions. *J Global Opt* 13: 455–492.