BIOINFORMATICS ORIGINAL PAPER Vol. 26 no. 17 2010, pages 2145–2152
doi:10.1093/bioinformatics/btq353

Genetics and population analysis Advance Access publication July 17, 2010

Predictive models for population performance on real biological fitness landscapes

William Rowe[1,*,†], David C. Wedge[2,3,†], Mark Platt[4], Douglas B. Kell[2,3] and Joshua Knowles[2,5]

[1]Faculty of Life Sciences, The University of Manchester, Michael Smith Building, Manchester M13 9PT, [2]Manchester Interdisciplinary Biocentre, The University of Manchester, 131 Princess Street, Manchester M1 7DN, [3]School of Chemistry, The University of Manchester, Oxford Road, Manchester, M13 9PL, UK, [4]School of Chemistry and Chemical Biology, University College Dublin, Belfield campus, Dublin 4, Ireland and [5]School of Computer Science, University of Manchester, Kilburn Building, Oxford Road, Manchester, M13 9PL, UK

Associate Editor: Jeffrey Barrett

ABSTRACT

Motivation: Directed evolution, in addition to its principal application of obtaining novel biomolecules, offers significant potential as a vehicle for obtaining useful information about the topologies of biomolecular fitness landscapes. In this article, we make use of a special type of model of fitness landscapes—based on finite state machines—which can be inferred from directed evolution experiments. Importantly, the model is constructed only from the fitness data and phylogeny, not sequence or structural information, which is often absent. The model, called a landscape state machine (LSM), has already been used successfully in the evolutionary computation literature to model the landscapes of artificial optimization problems. Here, we use the method for the first time to simulate a biological fitness landscape based on experimental evaluation.

Results: We demonstrate in this study that LSMs are capable not only of representing the structure of model fitness landscapes such as NK-landscapes, but also the fitness landscape of real DNA oligomers binding to a protein (allophycocyanin), data we derived from experimental evaluations on microarrays. The LSMs prove adept at modelling the progress of evolution as a function of various controlling parameters, as validated by evaluations on the real landscapes. Specifically, the ability of the model to 'predict' optimal mutation rates and other parameters of the evolution is demonstrated. A modification to the standard LSM also proves accurate at predicting the effects of recombination on the evolution.

Contact: william.rowe@manchester.ac.uk

Received on March 26, 2010; revised on June 8, 2010; accepted on June 29, 2010

1 INTRODUCTION

A fitness landscape is a conceptual visualization of the topographic relationship between a genotype and a static fitness function, a property that will determine reproductive rate (Wright, 1932). The properties of the fitness landscape relate to the behaviour of evolving populations and as such they have become integral tools in both evolutionary biology and evolutionary computation. Despite their long term and widespread use, the information pertaining to real biological fitness landscapes is currently limited. As a consequence, researchers interested in studying the dynamics of evolution are often forced to rely on *in silico* landscape models instead. Many artificial fitness landscapes have attempted to model the properties of real biomolecular fitness landscapes. These have ranged from John Maynard Smith's simple word game metaphor for protein space (Maynard Smith, 1970), to more complex models, which incorporate properties such as ruggedness and neutrality (Barnett, 1998; Kauffman and Levin, 1987; Mitchell *et al*., 1992). The real fitness landscapes of proteins are poorly understood and so the accuracy of these simplistic models is unknown.

At the molecular level adaptive evolution is routinely studied and exploited during the process of directed evolution. The steps within a directed evolution experiment mimic Darwinian processes within a lab and closely resemble the structure of many standard genetic algorithms, methods used widely in computer science for optimization. Iterative cycles of mutation (and/or recombination) and selection have led to the development of a plethora of novel or improved biological entities (Alexeeva *et al*., 2003; Ellington and Szostak, 1990; Giver *et al*., 1998). This has been achieved without any guidance from a molecular level understanding of protein function.

Sequencing the intermediates produced during the course of a directed evolution experiment reveals the series of modifications that have led to the new functionality. This approach has provided new insights into the relationship between the sequence and function of proteins and how proteins may adapt during natural evolution (Romero and Arnold, 2009). Determining how properties such as selection pressure, mutation rate and library size (population) will affect the success of future experiments, however, would require extrapolation beyond these limited evaluations.

Recently, a study was undertaken to map the complete interaction profile of every possible 10-base oligonucleotide with a fluorescent protein (allophycocyanin), using the ability to perform highly multiplexed assays afforded by microarrays (Rowe *et al*., 2009). Based on over one million evaluations the data set reveals a complete sequence-fitness landscape. This still represents a simple biomolecular interaction; ideally, a resource is desired that represents more complex biological systems. No doubt, with the continuing

*To whom correspondence should be addressed.
†The authors wish it to be known that, in their opinion, the first two authors should be regarded as joint First authors.

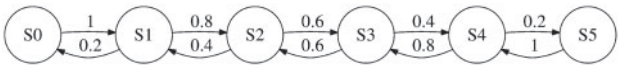

Fig. 1. Representation of an exact LSM for the MaxOnes problem where $L=5$ (Corne *et al.*, 2003).

development and availability of high-throughput sequencers and/or microarrays, even more extensive datasets will emerge over the coming months and years. This potential development, while exciting, is only part of the battle, however. In order to understand something useful about the landscapes and the dynamics of evolution on them, appropriate models or abstractions from the raw data must be inferred.

In computer science, in the field of evolutionary computation, characterizing fitness landscapes and how their features might affect evolutionary progress and dynamics is an area of intense study (Jones and Forrest, 1995; Kallel *et al.*, 2001; Merz, 2004). Grefenstette (1995) and Altenberg (1995) independently developed the idea of modelling/predicting evolutionary progress from fitness distribution data using, respectively, simple regressions and 'search kernels' to represent these distributions. The Landscape Machine (LSM; Corne *et al.*, 2003) was developed independently, but is a concrete implementation of Altenberg's search kernel. It models the fitness landscape of a particular abstract optimization problem as a finite state machine, representing landscapes purely as collections of phenotypic states and describes transitions between these states as a consequence of mutation. Consider S is the collection of all possible fitness values within a landscape, and s ($s \in S$) represents a value associated with an individual (for simplicity a single sequence). When s (the parent's fitness) is transformed by mutation, a new value e ($e \in S$, the child) is generated with a probability t_{se}. The probabilities of transformation between all points within S can be stored within a transition matrix, T, and used to 'replace' the real landscape.

In some cases, the LSM can model these transitions exactly. Figure 1 represents a simple MaxOnes problem. The states s_{0-5} represent values of solutions to the problem, in which the aim is to optimize the number of 1 s within a binary string {0,1} of length $L=5$ (the states match the number of 1 s within the string). Arcs indicate the transition probabilities between these states when a single random bit is flipped ($0 \rightarrow 1$ or $1 \rightarrow 0$) within a string. Such a model can act as a substitute for the real problem when assessing the performance of a genetic algorithm, in which candidate solutions (variations of the string) have been replaced by these states. The success and speed of several different algorithms (meaning different population sizes, different methods of biased selection, different selection pressures and so forth) can then be assessed without the use of any genotypic information.

For more complex landscapes, calculating these transitions or even storing a matrix of transitions between all points within S becomes intractable. If this is the case, points within S are pooled together into equally spaced fitness intervals (states) and transition probabilities are inferred from values generated by a sampling algorithm run on the real landscape. Using this approach, landscape state machines have shown a high degree of accuracy and specificity in predicting the performance of different evolutionary algorithms on a range of optimization problems studied in Computer Science (Corne *et al.*, 2003; Rowe *et al.*, 2006).

LSMs were proposed not only for the study of computer science optimization problems, but also with the intention of future use in directed evolution (Corne *et al.*, 2003). In this article, we begin to examine, empirically, their potential in this area for the first time.

Until now, no attempt has been made to model recombination events using LSMs. This task is not a trivial extension to the scheme since populating a three dimensional transition matrix (for the fitnesses of two parents and one child) would require a much greater number of evaluations on the real landscape. In other words, the sampling will be at an even greater order of sparseness than with the standard LSM. The LSM will also be limited in the algorithms it can assess because the sampling algorithm will only sample combinations of parents from a range dictated by the selection pressure employed and the properties of the fitness landscape. We attempt to correct for these problems using a simple heuristic, described in 'Methods' section.

The performances of a range of genetic algorithms (including those that incorporate recombination) were assessed on both real landscapes and LSM abstractions of these landscapes. NK-landscapes (Kauffman and Levin, 1987) with different levels of epistasis and a DNA–protein interaction landscape derived from real experimental evaluations were both investigated. The LSMs proved adept at modelling the features of the landscapes, with strong correlations between the relative performances of the LSMs and the corresponding real-world algorithms. The ability to model real biological fitness landscapes with sufficient fidelity to allow predictions of the most appropriate (near-optimal) parameters of evolution (mutation rates, recombination, selection pressures, etc.) using fitness information alone seems to be an encouraging prospect for future progress in this area.

2 METHODS

2.1 Artificial fitness landscape: the NK Model

The NK-model is a Boolean structure conceived by Kauffman (Kauffman and Levin, 1987) that assigns continuous values to combinations of bits in a binary string. The model has been used to represent the combinatorial effects of genes in a whole organism but can equally be applied to the interaction of amino acids in a protein structure. The fitness of each individual position (gene), f_i, is determined by the state of K coupled residues as well as its own state. N is the dimensionality of the landscape, which is equal to L, the length of the string (chromosome). The total fitness F of a string is the average fitness across all f_i, as given in Equation (1), in which i is the position assessed and α_n are the states (1 or 0) of the $K+1$ coupled residues.

$$F = \frac{1}{N}\sum_{i=1}^{N} f_i(\alpha_{i1}, \alpha_{i2}, \ldots \alpha_{ik+1}) \ \alpha \in \{0,1\}^N \quad (1)$$

The properties of the landscape can be tuned by varying K. When K is zero the landscape is smooth resulting in a single peak of the Mount Fuji type. When K is increased the landscape becomes increasingly rugged resulting in many fitness peaks, making it more difficult for algorithms with high-selection pressure to find the global optimum. Two NK-landscapes were assessed in this study, with $K=2$ and $K=3$. Both had $N=100$.

2.2 Experimentally-derived DNA–protein interaction landscape

The second landscape was obtained from measured experimental data (Rowe *et al.*, 2009). These data comprise every DNA sequence ten bases in length and their corresponding affinities with the fluorescent protein

allophycocyanin (as measured in duplicate through hybridisation on a microarray chip). The properties of the resultant landscape were studied using various statistical measures in comparison to known NK-landscapes with noise added at a similar level to that observed in the experimental replicates. A comprehensive data set allows us to evaluate genetic algorithms' performance on the data as they would on the real landscape without the need to perform a single measurement. The data set can be downloaded from *http://dbkgroup.org/direvol.htm*.

2.3 Algorithm design

The algorithms that are assessed in this study are based upon a standard (μ, λ) algorithm (Back, 1995), chosen as it resembles the approach employed by many experimentalists when performing directed evolution. The fittest μ individuals within a population of size λ are selected and reproduced iteratively over a number of generations, with diversity introduced through bit-flip point mutations and recombination. Point mutation rates refer to the probability that each bit in a chromosome will be flipped ($0->1$ or $1->0$), or in the case of the experimental landscape that a base will be switched to one of the three alternatives.

The performances of multiple algorithms with the following properties were assessed on both the experimental and real landscapes (L is the length of the sequences under investigation). The algorithms based on the NK-landscape were chosen to match those previously used to model directed evolution (Wedge *et al*., 2009).

NK-landscapes

- $\mu = 2, 4, 40$ and 400, $\lambda = 40\,000$
- Mutation rates {0.2, 0.4, 0.8, 1.0, 2.0, 3.0, 4.0, 5.0, 10.0, 15.0, 20, 25.0 and 30.0}/L
- With and without recombination.

Experimental-landscapes

- $\mu = 2, 5, 10$ and 100, $\lambda = 1000$
- Mutation rates {0.2, 0.4, 0.8, 1.0, 2.0, 3.0, 4.0, 5.0 and 10.0}/L
- With and without recombination.

Recombination, when present, was applied at a rate of 0.6, with the two parents selected randomly, to produce a single offspring in the next generation, using standard uniform crossover (Sywerda, 1989). Performance was measured as the mean best fitness observed during the course of each evolution based on 100 replicates.

2.4 Construction and assessment of LSMs

In order to construct an LSM, it is necessary to sample transitions (mutation or recombination steps) from the problem landscape. In this work, transition probabilities for LSMs were inferred from the NK-landscapes and experimental landscape using genetic algorithms with the following properties:

NK-landscapes

- A standard (μ, λ) algorithm, with $\mu = 400$ and $\lambda = 40\,000$.
- A mutation rate of 0.2/L.
- Uniform crossover at a rate of 0.6.
- 21 generations
- 10 replicates

Experimental-landscape

- A standard (μ, λ) algorithm, with $\mu = 10$ and $\lambda = 1000$.
- A mutation rate of 0.2/L.
- Uniform crossover at a rate of 0.6.
- 21 generations
- 50 replicates

Fig. 2. A mutation that is included in the transition probabilities of an LSM when applying a transition matrix repeatedly to model a mutation rate higher than that sampled by the LSM. This mutation cannot occur during the equivalent standard mutation on the real landscape.

All parent and child fitness values were recorded and stored. Scores were then partitioned into 200 evenly spaced states based upon the highest and lowest values recorded by the sampling algorithm. The partitioned parent and child data were then used to populate a 200×200 transition matrix. The corresponding states of individuals from the first generation of the sampling algorithm were stored separately. Individuals were selected randomly from these states to form the starting populations of all algorithms tested using the LSMs.

Genetic algorithm performance was then assessed using the LSM as it would be on the real landscape. Mutations were represented by states changes, with the new states determined by fitness proportionate selection from the transition probabilities associated with the parent's state. The algorithm performance on the LSMs was evaluated over 1000 replicates by measuring the highest state achieved or, if two algorithms achieved equal values, the number of generations taken to reach this value.

2.5 Modelling mutation rates

Higher mutation rates can be approximated by raising the transition matrix, T, to the appropriate power. For example, to model a mutation rate of 1/L, based upon a sampling algorithm using a mutation rate of 0.2/L, T would be applied five times (represented as T^5). This approach approximates the higher mutation rates but can become inexact under certain circumstances. First, if the LSM is extrapolated to a high mutation rate from an extremely low mutation rate, any errors in sampling (where the true transition probabilities have not converged) will become magnified. To assess this effect, a sampling algorithm was also run at 1/L and the performance at higher mutation rates was compared to that of the sampling algorithm run at 0.2/L.

A second problem arises when modelling landscapes where L is small. An LSM modelling a mutation rate of 2/L derived from a sampling algorithm run at 1/L is essentially modelling two successive rounds of mutations. When L is small the probability that a bit (or a base) will be mutated and then mutate back to the original base in the second round of mutations is high (see Figure 2). This will have the effect of distorting the transition probabilities. As L increases, the probability of these 'nullifying mutations' decreases and so they have less effect on the transition probabilities. It is simple to compute these errors exactly. Consider a sequence of length 5 and a mutation rate of 2/L. The probability that all five bits flip from 0 to 1 is $(2/L)^L = 0.01024$. The probability that all five bits are flipped $P(\alpha_{1t} = 1, \alpha_{2t} = 1 \ldots \alpha_{Lt} = 1)$ by two consecutive applications of a 1/L mutation is a product of the probability that a 1 is generated (by one of two routes $0 \rightarrow 1 \rightarrow 1$ or $0 \rightarrow 0 \rightarrow 1$) at each position i in L. This is given by Equation (2), in which $\alpha \in \{0,1\}$.

$$P(\alpha_{1t} = 1, \alpha_{2t} = 1 \ldots \alpha_{Lt} = 1) = \prod_{i=1}^{L} (P(\alpha_{it} = 1 | \alpha_{it-1} = 1 | \alpha_{it-2} = 0) + P(\alpha_{it} = 1 | \alpha_{it-1} = 0 | \alpha_{it-2} = 0)) \quad (2)$$

where α_{it} is the state (1 or 0) after i successive mutations. When $L = 5$ this equals 0.00336, about a third as much. Thus, for a short sequence and an extreme event, probabilities are underestimated.

These errors could be corrected by using transition matrices for single-mutation events rather than for a fixed mutation rate. These matrices

would be applied x times, where x is determined by the expected probability of $1, 2, \ldots, L$ mutations. However, this would incur some cost to the simplicity of the method and would require the collection of sequence information that is not commonly available during directed evolution experiments. An alternative approach, which we adopt here, is to retain the simpler fixed mutation rate model, but to assess its accuracy. We note in advance, however, that the NK-landscape is much less likely to be affected by this problem than is the experimental landscape as $L = 100$ in the former and $L = 10$ in the latter.

2.6 Modelling recombination

To model the consequence of crossover on genetic algorithm performance, a separate recombination transition matrix was constructed. In this 3D array, an element e_{ijk} represents the (sample) probability of obtaining offspring of fitness k from parents of fitnesses i and j (where here fitnesses are understood to have been discretized and normalized), where the array is symmetric and i is the fitter parent. Due to considerations of the sparseness of this array, a GA evaluated on the LSM may require transitions from a parental combination that was not encountered by the sampling algorithm. To address this issue, a heuristic was implemented such that instead of using row $e_{\_ij}$ of the matrix, e_{im} was used, where $m \leq i$ was the closest row to j in which at least one crossover event had been encountered during sampling.

A single sampling algorithm was used to populate the mutation and crossover matrices: when individuals were selected for crossover this event was used to populate the crossover LSM (fitness values were assessed before mutation was applied); when individuals were mutated these operations were stored in the mutation only LSM. To implement the effect of crossover the two matrices were used in series, with the crossover-modelling matrix applied before the mutation-modelling matrix. Compatibility between the crossover and mutation matrices was ensured by parameterizing the data into the same intervals within both matrices. Keeping the two processes separate enables the assessment of a wider range of algorithms using the LSMs.

The ability of the LSMs to represent the different landscapes was measured in three different ways:

- Pearson correlation between the algorithm performances on the real landscape and the LSM, at different mutation rates. For example this could represent the correlation between the performance of a (μ, λ) algorithm with $\mu = 2$ and $\lambda = 1000$ over a range of mutation rates (0.2, 0.4, 0.8, 1.0, 2.0, 3.0, 4.0, 5.0 and 10.0/L) on the experimental landscape and on the LSM.
- The ability of an LSM to determine the optimal mutation rate for each algorithm.
- The overall rank correlation between the performances of the LSM and (μ, λ) search on a real landscape as measured by the Kendall tau correlation coefficient. This represents the correlation of the performance rankings on the LSM and the real landscape across all algorithms, taking into account variations in μ, the presence or absence of crossover, and the range of mutation rates, and therefore represents a single statistic for each landscape.

3 RESULTS

3.1 NK-Landscapes

Plots of the performance of each of the algorithms assessed on the real landscapes and the LSMs are displayed in Figures 3–6; the corresponding Pearson correlation coefficients are listed in Table 1.

The correlations between the values produced on the LSM and the real landscapes are high for both NK-landscapes, ranging from 0.94 to 1.00. While these values are strikingly high, in real terms they only relate to the general trends in performance of each of the algorithms (with increasing mutation rates) in isolation. The relative performance at each mutation rate was ranked for all algorithms and

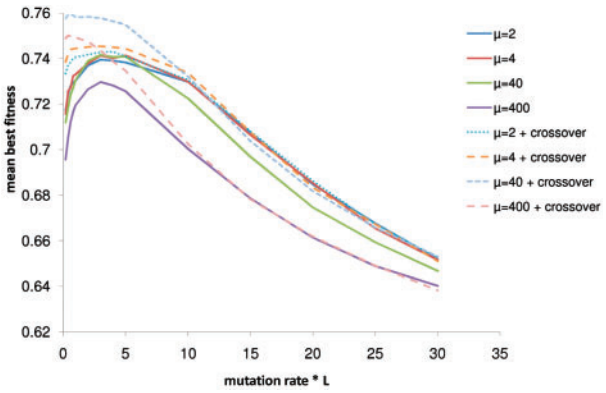

Fig. 3. Displaying plots of differing GA performance with increasing mutation rate based on evaluations on the NK-landscape, where $N = 100$ and $K = 2$ ($\lambda = 40000$).

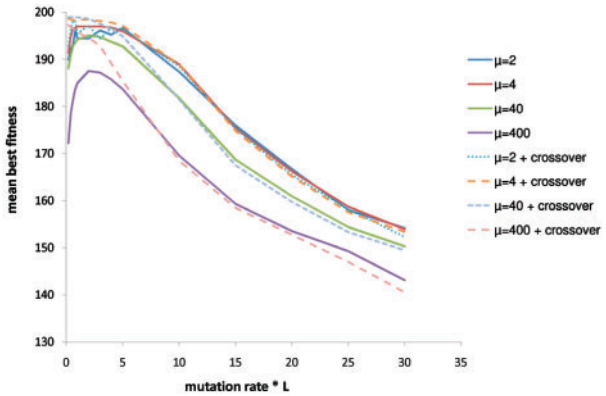

Fig. 4. Displaying plots of differing GA performance with increasing mutation rate based on evaluations on an LSM derived from an NK-landscape, where $N = 100$ and $K = 2$ ($\lambda = 40000$) (sampled at 0.2/L).

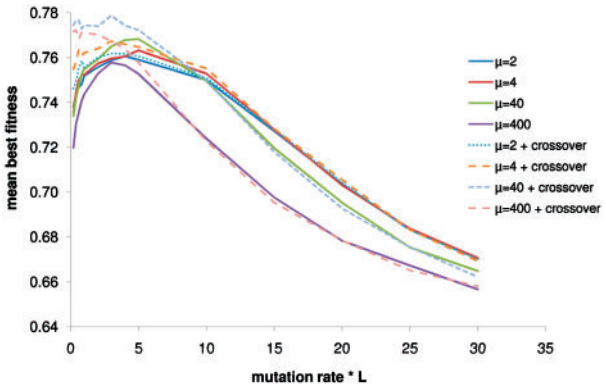

Fig. 5. Displaying plots of differing GA performance with increasing mutation rate based on evaluations on the NK-landscape, where $N = 100$ and $K = 3$ ($\lambda = 40000$).

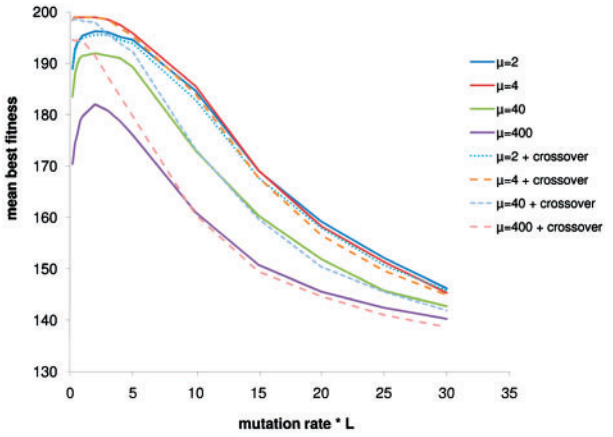

Fig. 6. Displaying plots of differing GA performance with increasing mutation rate based on evaluations on an LSM derived from an NK-landscape, where $N=100$ and $K=3$ ($\lambda=40\,000$) (sampled at $0.2/L$).

Table 1. Pearson correlation coefficients observed between average best fitness values observed on two NK-landscapes and those predicted using the LSM for different μ, λ algorithms (where $\lambda=40\,000$)

Algorithm	$K=2$	$K=3$
$\mu=2$	0.98	0.97
$\mu=4$	0.97	0.94
$\mu=40$	0.97	0.95
$\mu=400$	0.99	0.97
$\mu=2$+crossover	0.99	0.98
$\mu=4$+crossover	0.99	0.97
$\mu=40$+crossover	1.00	0.98
$\mu=400$+crossover	1.00	0.99

the degree of correspondence between the rankings on the LSMs and the (μ, λ) algorithm was determined using the Kendall tau correlation coefficient (Kendall, 1938).Values of tau were 0.82 and 0.64 for the landscapes where $K=2$ and $K=3$, respectively.

Performance can also be measured as the ability of the LSMs to determine the best parameters such as mutation rate for further optimizations in experiments such as directed evolution. In practical terms, the optimal mutation rate will vary according to many factors, including the properties of the landscape and construction of the algorithm assessed (Cervantes and Stephens, 2009). It has for instance been observed that high mutation rates are superior when libraries are large in directed evolution (Drummond *et al.*, 2005). Amongst the evolutionary computing community there has, however, been almost a prescriptive trend to use the $1/L$ heuristic (Mühlenbein, 1992), to set the mutation rate.

Of the 16 algorithms assessed, the LSMs predict 3 optimal mutation rates correctly (see electronic supplementary information). On average the predictions disagree with the real data by $1.79/L$, with seven predictions disagreeing by more than $1/L$. Nevertheless, the use of LSM-predicted mutation rates results in a significant improvement over the use of a fixed $1/L$ rate. A fixed $1/L$ mutation rate is on average inaccurate by $2.2/L$. None of the optimal mutation rates correspond to $1/L$ and 10 algorithms have optimal

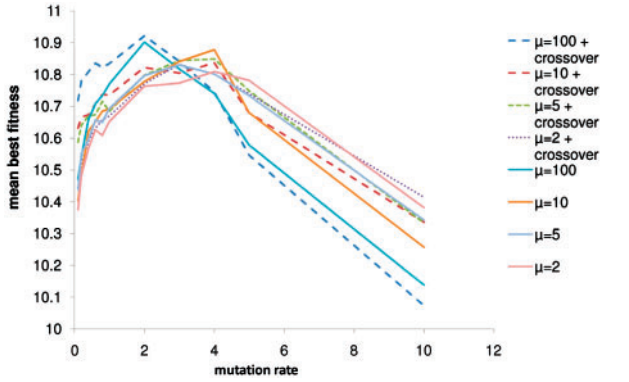

Fig. 7. Displaying plots of differing GA performance with increasing mutation rate based on evaluations using the experimental landscape data ($L=10$, $\lambda=1000$).

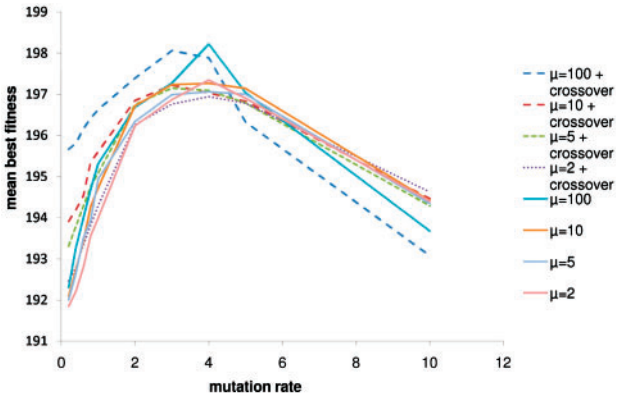

Fig. 8. Displaying plots of differing GA performance with increasing mutation rate based on evaluations on an LSM derived from the experimental landscape using a sampling algorithm run at $0.2/L$ ($L=10$, $\lambda=1000$).

mutation rates during search on the real landscape that differ from this value by more than $1/L$.

3.2 Experimental landscape

Plots of the performance of the different algorithms on the experimental landscapes and the resultant LSMs are displayed in Figures 7 and 8. There are many differences between the experimental landscape and the NK-landscapes listed previously. In particular:

- The experimental landscape has been shown to be filled with local optima from noise. However, there is still a high underlying correlation between neighbouring (in terms of sequence similarity) sequences (Rowe *et al.*, 2010).
- The sequences are shorter than those that comprise the NK-landscapes.

Despite these two factors there is an overall consistency shown by the genetic algorithms, all of which performed better at mutation rates lower than $10/L$ (a level at which new sequences are essentially generated randomly).

Table 2. Pearson correlation coefficients observed between average best fitness values observed on the experimental fitness landscape and those predicted using the LSM for different μ, λ algorithms (where $\lambda = 40000$)

Algorithm	Correlation
$\mu = 2$	0.76
$\mu = 5$	0.67
$\mu = 10$	0.62
$\mu = 100$	0.49
$\mu = 2$+crossover	0.74
$\mu = 5$+crossover	0.67
$\mu = 10$+crossover	0.59
$\mu = 100$+ crossover	0.79

The Pearson correlation coefficients of the average best fitness achieved on the real landscape and the LSM are much lower for the experimental landscape than for the NK-landscapes (Table 2). Correlations between 0.49 and 0.79 are observed for each individual algorithm and a value of tau of 0.61 from the Kendall tau correlation test was recorded, based on the overall predicted and observed values. Although these values are not as high as those for the NK-landscapes, there are still very clear trends in algorithm performance on the real landscape that are identified by the LSM. This is best exemplified by the clear differentiation of performance of the two algorithms where $\mu = 100$ (with and without crossover) from the other algorithms.

In terms of predicting the optimum mutation rate the LSM performs well. The predictions differ from the actual landscape on average by only $0.88/L$, with only one algorithm differing by more than $1/L$. In contrast the $1/L$ heuristic performs poorly on this landscape, with optimal mutations rates differing on average by $2.25/L$.

3.3 Recombination

To determine how accurately the LSM is modelling the effect of crossover, the specificities of the predictions were assessed. To assess whether the LSMs could accurately depict the difference between algorithms with and without crossover a simple metric was applied. The performance (mean best score) of each algorithm without crossover was deducted from the performance of the same algorithm with crossover at each mutation rate. Specificity was recorded as the Pearson correlation between values derived from evaluations on the LSM and those on the real landscape.

From the results in Table 3, it can be seen that there is a general trend between the performance of the LSMs at modelling the effect of crossover and the value of μ for each algorithm. At low values of μ the correlation between the LSMs and the real landscapes is low. As μ increases the correlation increases, reaching 0.99 in each of the NK-landscapes where $\mu = 400$ and 0.97 in the experimental landscape where $\mu = 100$. At low values of μ, there is little difference in the performance of the algorithms with and without crossover on all of the real landscapes assessed (Table 4). When the performances of the two sets of algorithms are so similar it is difficult to model the difference between the two. As μ increases, the algorithms that include crossover increasingly outperform those without and consequently the specificity metric increases.

Table 3. Pearson correlations representing the difference in performance of algorithms with and without crossover on the real landscape and the performance of algorithms with and without crossover on the LSM

μ	NK, $K = 2$	μ	NK, $K = 3$	μ	Experimental landscape
2	0.75	2	−0.03	2	0.35
4	0.90	5	−0.02	4	0.77
40	0.98	10	0.93	40	0.75
400	0.99	100	0.99	400	0.97

Table 4. Pearson correlation coefficients between the performances of algorithms with and without crossover with increasing values of μ on the NK and experimental landscapes

μ	NK, $K = 2$	NK, $K = 3$	μ	Experimental landscape
2	0.99	0.99	2	0.98
4	0.98	0.98	5	0.95
40	0.95	0.96	10	0.92
400	0.92	0.93	100	0.90

3.4 Effect of sampling

Sampling the landscapes with an algorithm using a mutation rate of $0.2/L$ means relying on a huge extrapolation of the sampled probabilities when predicting performance at $30/L$. Figures 7 and 8 represent the performance of algorithms on the experimental landscape and the corresponding LSM. For this landscape, L is low (10). From these figures, it may be seen that the relative performances of the algorithms at low and high mutation rates are incorrectly predicted by the LSMs. Specifically, in the real landscape the performances of algorithms at low mutation rates ($0.2/L$) are better than those at high mutation rates ($10/L$), whereas for the LSM this trend is reversed. This is most likely a result of inaccuracies in the transition probabilities when L is low owing to the inherent bias in the simulation of high mutation rates due to the effect that mutations can be 'undone' (see discussion above). While the LSM has been raised to the power of 10 the effective mutation rate is actually much lower.

Table 5 shows the correlation of performance of genetic algorithms on the real landscapes and two LSMs, derived from sampling algorithms employing a $1/L$ mutation rate and a $0.2/L$ mutation rate (only genetic algorithms run at a mutation rate of $1/L$ or greater were assessed). For each of the three landscapes—NK-landscape with $K = 2$, NK-landscape with $K = 3$ and the experimental landscape—8 different mutation rates were assessed. An increase in accuracy when sampling at $1/L$ was seen in, respectively, 8, 5 and 7 cases. In the four cases where there was a loss in accuracy the drop was small (0.036 maximum). In contrast, the gains in accuracy were in some cases high, especially in the experimental landscape. In Supplementary Figures 1 and 2 it can be seen, for instance, that the performance of a genetic algorithm is predicted to be better at $10/L$ than at $1/L$. However, with the LSM sampled at $0.2/L$ this is not the case.

Sampling at the higher mutation rate also greatly improves prediction of the optimal mutation rate compared to the use of the $1/L$ heuristic. Of the 16 algorithms assessed on the NK-landscapes the

Table 5. Pearson correlation coefficients between average best fitness values observed using the NK-landscapes ($K=2$, $K=3$) and the experimental fitness landscape and those predicted using LSMs sampled at $1/L$ and $0.2/L$ for different μ, λ algorithms (where $\lambda=40000$) with varying mutation rate (X denotes crossover)

	NK, $K=2$		NK, $K=3$			Experimental landscape	
Algorithm	$1/L$	$0.2/L$	$1/L$	$0.2/L$	Algorithm	$1/L$	$0.2/L$
$\mu=2$	0.993	0.990	0.971	0.982	$\mu=2$	0.966	0.927
$\mu=4$	0.992	0.990	0.965	0.976	$\mu=5$	0.989	0.878
$\mu=40$	0.990	0.989	0.975	0.971	$\mu=10$	0.959	0.915
$\mu=400$	0.996	0.994	0.989	0.982	$\mu=100$	0.891	0.767
$\mu=2$+X	0.997	0.995	0.980	0.982	$\mu=2$+X	0.982	0.715
$\mu=4$+X	0.996	0.994	0.982	0.981	$\mu=5$+X	0.969	0.784
$\mu=40$+X	0.996	0.995	0.991	0.983	$\mu=10$+X	0.987	0.837
$\mu=40$+X	0.999	0.997	0.996	0.988	$\mu=100$+X	0.891	0.787

LSMs sampled at $1/L$ predict five optimal mutation rates correctly. Most importantly, the predictions of the LSM sampled at $1/L$ are better than (or at least as good as) the $1/L$ heuristic on every single landscape and at every selection-pressure setting, both with and without crossover.

4 DISCUSSION

Predicting the performance of genetic algorithms using limited sampling is not a new concept and a multitude of statistics have been developed to characterise the features of landscapes and the consequent effects on algorithm performance (Naudts and Kallel, 2000). These statistics tend to be self validating in that they are correct only in their own terms and require prior knowledge of the landscape they are characterising. In practice these predictors of genetic algorithm hardness often underperform and are incapable even of determining whether a hillclimber (an algorithm that accepts only beneficial changes) will be more effective than a genetic algorithm. Identifying whether a landscape is 'easy' or 'hard' to search using these measures gives almost no indication of how to tackle it. Currently, the only way to tune algorithms for optimal performance on a landscape is to assess the performance of these algorithms through trial and error. LSMs are abstractions of real landscapes based on limited sampling data, allowing genetic algorithm performance to be assessed as it would be on the real landscapes. The ability of the LSM to characterize the landscape of interest can be directly measured by comparison of the performance of the genetic algorithms on the landscapes and on the LSMs.

The LSMs in this study are capable of closely replicating algorithm performance on the landscape from which they are derived. The cost of generating the models described here is high, with the number of evaluations sampled from the experimental landscape (1 050 000) similar to the number required to map the entire 10-mer landscape (1 048 576). In contrast no sequence information is required to construct the LSMs. This makes LSMs cheap to construct as by-products of directed evolution experiments, which are more than capable of producing quantities of data of this magnitude.

It has previously been shown that accurate LSMs can be constructed with far fewer real evaluations and an effective sampling algorithm can be beneficial to LSM accuracy (Rowe *et al*., 2006). In this study, a sampling algorithm with a low mutation rate was selected not for its performance on the landscape, but because from this low mutation rate a much greater range of mutation rates can be assessed by the LSM through matrix multiplication. This comes at a price: higher mutation rate transitions may not be sampled. The predictions made on the performance of algorithms with much higher mutation rates show only a small loss in accuracy, which is remarkable given that any error within the transition matrix is multiplied with every matrix multiplication.

The most important finding of this study has been that the simple structure of the LSM is capable of mimicking the features of a real biological landscape derived from real experimental data, despite the noise that is inherent to such measurements and the complex sequence/structural interactions. It would be interesting to determine to what extent LSMs can replicate real biological fitness landscapes, and whether they can be extended to model non-static and multiple fitness functions. It may be necessary to modify the structure of the LSMs (as we have attempted here to model recombination). For instance higher order Markov models may be more appropriate for highly neutral and epistatic landscapes. Further, landscape state machines are not limited to studying static mutation rates but could also be applied to dynamic mutation rates and could feasibly be implemented in memetic algorithms (Merz, 2004).

Directed evolution experiments give an insight into the process of molecular evolution that is unobtainable at the level of whole organisms. Parent and child fitness values have previously been used to make quantitative assessments of biomolecular fitness landscapes through measures such as autocorrelation. LSMs go beyond simple landscape metrics, providing an abstraction of the real systems upon which further experiments can be performed (Altenberg, 1995). With the onset of high-throughput methodologies for the assessment of modified/transformed peptides (Drummond *et al*., 2005), proteins and cells, LSMs can make use of this deluge of data in a tractable manner independent of any sequence or structural information. The resulting LSMs may serve as a resource to researchers studying molecular evolution, providing a platform beyond simplistic and often unrealistic model landscapes.

In this study, we extended LSMs to generate 3D structures capable of modelling recombination in genetic algorithms (Altenberg, 1995). Using several matrices representing different mutation rates serially can permit the assessment of algorithms with dynamic mutation rates and there is no reason to suggest that the structure of the LSMs could not be extended to model multiple fitness functions simultaneously. Experiments performed using NK-landscapes and an experimental landscape indicates LSMs represent a promising tool in both evolutionary computation and evolutionary biology.

Funding: We acknowledge sponsorship by the Biotechnology and Biological Sciences Research Council PBB/D000203/1. D.W. was funded by the Home Office 'Low Cost Sensor Arrays Using Organic Semiconductors' project.

Conflict of Interest: none declared.

REFERENCES

Alexeeva,M. *et al*. (2003) Directed evolution of enzymes: new biocatalysts for asymmetric synthesis. *Org. Biomol. Chem.*, **1**, 4133–4137.

Altenberg,L. (1995) The schema theorem and Price's theorem. In *Foundations of Genetic Algorithms*. Morgan Kaufmann, San Francisco, pp. 23–49.

Bäck,T. (1995) Generalized convergence models for tournament- and (mu, lambda)-selection. In *Proceedings of the 6th International Conference on Genetic Algorithms*. Morgan Kaufmann Publishers Inc., San Francisco.

Barnett,L. (1998) Ruggedness and neutrality - the NKp family of fitness landscapes. In *Alive VI: Sixth International Conference on Artificial Life*, MIT-Press, Cambridge, MA, USA, pp. 18–27.

Cervantes,J. and Stephens,C.R. (2009) Limitations of existing mutation rate heuristics and how a rank GA overcomes them. *IEEE Trans. Evol. Comput.*, **12**, 369–397.

Corne,D. *et al*. (2003) Landscape state machines: tools for evolutionary algorithm performance analyses and landscape/algorithm mapping. *Lecture Notes Computer Science*, Springer, Berlin, pp. 197–198.

Dayhoff,M.O. *et al*. (1978) A model of evolutionary change in proteins. In Dayhoff, M.O. (ed), *Atlas of Protein Sequence and Structure*. National Biomedical Research Foundation, Silver Spring, MD, pp. 345–352.

Drummond,D.A. *et al*. (2005) Why high-error-rate random mutagenesis libraries are enriched in functional and improved proteins. *J. Mol. Biol.*, **350**, 806–816.

Ellington,A.D. and Szostak,J.W. (1990) In vitro selection of RNA molecules that bind specific ligands. *Nature*, **346**, 818–822.

Giver,L. *et al*. (1998) Directed evolution of a thermostable esterase *Proc. Natl Acad. Sci. USA*, **95**, 12809–12813.

Grefenstette,J.J. (1995) Predictive models using fitness distributions of genetic operators. In *Foundations of Genetic Algorithms 3*. Morgan Kaufmann, San Mateo, CA, pp. 139–161.

Jones,T. and Forrest,S. (1995) Fitness distance correlation as a measure of problem difficulty in genetic algorithms. In *Proceedings of 6th International Conference on Genetic Algorithms*. Morgan Kauffman, Pittsburgh, PA, pp. 184–192.

Kallel,L. *et al*. (2001) Properties of fitness functions and search landscapes. In *Theoretical Aspects of Evolutionary Computing*. Springer, London, UK, pp. 175–206.

Kauffman,S. and Levin,S. (1987) Towards a general-theory of adaptive walks on rugged landscapes. *J. Theor. Biol.*, **128**, 11–45.

Kendall,M. (1938) A new measure of rank correlation. *Biometrika*, **30**, 81–89.

Merz,P. (2004) Advanced fitness landscape analysis and the performance of memetic algorithms. *Evol. Comput.*, **12**, 303–325.

Mitchell,M. *et al*. (1992) The royal road for genetic algorithms: fitness landscapes and GA performance. In *Proceedings of European Conf. on Artificial Life,* Varela, F.J. and Bourgine, P. (eds), MIT Press, Cambridge MA, pp. 245–254.

Maynard Smith, J. (1970) Natural selection and the concept of a protein space. *Nature*, **225**, 563–564.

Mühlenbein,H. (1992) How genetic algorithms really work: 1. Mutation and Hillclimbing *Parallel Problem Solving from Nature II*. Elsevier Science Amsterdam, 15–25

Naudts,B. and Kallel,L. (2000) A comparison of predictive measures of problem difficulty in evolutionary algorithms. *IEEE Trans. Evol. Comput.*, **4**, 1–15.

Romero,P.A. and Arnold,F.H. (2009) Exploring protein fitness landscapes by directed evolution. *Nat. Rev. Mol. Cell Biol.*, **10**, 866–877.

Rowe,L.A. *et al*. (2003) A comparison of directed evolution approaches using the beta-glucuronidase model system. *J. Mol. Biol.*, **332**, 851–860.

Rowe,W. *et al*. (2006) Predicting stochastic search algorithm performance using landscape state machines. *IEEE Congress on Evolutionary Computation (CEC 2006), Vancouver*. IEEE Press, Piscataway NJ, pp. 9849–9856.

Rowe,W. *et al*. (2010) Analysis of a complete DNA-protein affinity landscape. *J. Roy. Soc. Interface*, **7**, 397–408.

Stemmer,W.P. (1994) DNA shuffling by random fragmentation and reassembly: in vitro recombination for molecular evolution. *Proc. Natl Acad. Sci. USA*, **91**, 10747–10751.

Sywerda,G. (1989) Uniform crossover in genetic algorithms. *Proceedings of the Third International Conference on Genetic algorithms*. Morgan Kaufmann, George Mason University, USA.

Voigt,C.A. *et al*. (2000) Rational evolutionary design: the theory of in vitro protein evolution. *Adv. Protein Chem.*, **55**, 79–160.

Wedge,D.C. *et al*. (2009) In silico modelling of directed evolution: Implications for experimental design and stepwise evolution. *J. Theor. Biol.*, **257**, 131–141.

Wright,S. (1932) The roles of mutation, inbreeding, crossbreeding, and selection in evolution. *Proceedings of the Sixth International Congress on Genetics*, **1**, 355–366.