

Evolutionary Dynamics in a Simple Model of Self-Assembly

Iain G. Johnston,¹ Sebastian E. Ahnert,² Jonathan P. K. Doye,³ and Ard A. Louis¹

¹*Rudolf Peierls Centre for Theoretical Physics, University of Oxford, 1 Keble Road, Oxford OX1 3NP, UK*

²*Theory of Condensed Matter, Cavendish Laboratory,*

University of Cambridge, JJ Thomson Avenue, Cambridge CB3 0HE, UK

³*Physical & Theoretical Chemistry Laboratory, Department of Chemistry,
University of Oxford, South Parks Road, Oxford OX1 3QZ, UK*

(Dated: March 1, 2011)

We investigate the evolutionary dynamics of an idealised model for the robust self-assembly of two-dimensional structures called polyominoes. The model includes rules that encode interactions between sets of square tiles that drive the self-assembly process. The relationship between the model's rule set and its resulting self-assembled structure can be viewed as a genotype-phenotype map and incorporated into a genetic algorithm. The rule sets evolve under selection for specified target structures. The corresponding, complex fitness landscape generates rich evolutionary dynamics as a function of parameters such as the population size, search space size, mutation rate, and method of recombination. Furthermore, these systems are simple enough that in some cases the associated model genome space can be completely characterised, shedding light on how the evolutionary dynamics depends on the detailed structure of the fitness landscape. Finally, we apply the model to study the emergence of the preference for dihedral over cyclic symmetry observed for homomeric protein tetramers.

PACS numbers: 61.46.Bc, 87.10.Mn, 87.23.Kg, 81.16.Dn

I. INTRODUCTION

Self-assembly processes, in which constituent components reliably assemble into a complete structure without external control, are ubiquitous in nature, providing the means by which sophisticated biological machinery such as protein complexes are formed within organisms [1]. A key question then arises: how did the interactions that drive these self-assembly processes evolve over billions of years to form the optimised systems we observe today [2–4]? Bioinformatic studies of protein complexes [5] suggest that a number of observed trends in protein quaternary structure are caused not only by the biological function under selection, but also by the details of the evolutionary dynamics. Some of these trends have recently been explained by using computer simulations of a simple continuous patchy particle model [6] for globular proteins [7, 8]. However, such models are computationally expensive because a detailed simulation of the assembly process is required at each step in evolutionary time.

In this paper we study the evolutionary dynamics of a highly idealised coarse-grained model for the evolution of self-assembling systems, for which the assembly process can be simulated quickly and straightforwardly. The model consists of an ‘alphabet’ of square tiles that self-assemble into *polyominoes*: unions of connected cells on a 2D square lattice. The alphabet of available tiles, which we term the *assembly rule set*, contains a description of the interactions that drive the assembling system towards a final structure [9]. A physical interpretation of the model consists of a structure assembling on a 2D substrate in contact with a suspension of tiles, as shown in Fig. 1. These tiles can form many kinds of structures,

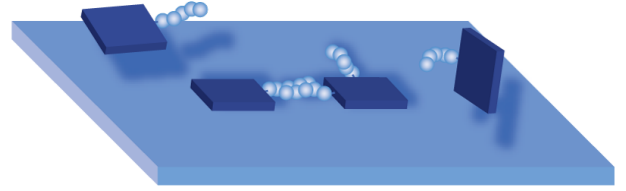


FIG. 1: (colour online) Illustration of a possible realisation of physical polyomino assembly. Square tile building blocks interact with each other through complementary bonding between edges, here illustrated with interacting polymer chains. In addition, tiles experience an attractive interaction to a flat substrate, leading to growing polyomino structures on a surface.

both bounded and unbounded. We focus on *deterministic* rule sets that always assemble into the same bounded 2D structures, a class of behaviour that is analogous to the monodisperse self-assembly observed for example for many kinds of protein quaternary structures.

These models may also be relevant for experimental systems such as 2D self-assembled systems that have been made of RNA [10] and DNA [11] tiles. Each tile can be tailored to interact with its neighbours through complementary bonding. Patterns and grids of varying geometries on the nanoscale have been produced by changing these design rules, with some examples being circuit patterns [12] and Sierpinski triangles [13]. The variety of structures that can be produced using DNA tiling [14] and DNA-linked particles [15] is rapidly increasing. The evolutionary design of polyomino structures may shed light on the design of these synthetic systems.

Pioneering work by Wang [16] demonstrated that tiles could be used to specify a Turing machine. In an important development, Winfree showed that DNA nanotechnology could be used to create molecular Wang tiles [11]. Self-assembling tile sets can thus perform computational tasks such as binary counting, and a measure of the complexity of assembly sets required for such algorithmic applications have been computed [17]. This theoretical work has been extended to study the details of tile assembly nucleation [18] and the effects of errors in the assembly process [19].

In this study, we use genetic algorithms (GAs) [20, 21] that search through the space of all possible rule sets to find those that generate the deterministic assembly of desired polyomino structures. Despite its simplicity, and resulting computational tractability, the model produces rich evolutionary behaviour. The assembly process can be viewed as a mapping that transforms an assembly rule set into an assembled polyomino structure. This mapping is reminiscent of the *genotype-phenotype map* in evolutionary biology, whereby information in the genome (the genotype) is used to develop the physical form of a biological structure (the phenotype).

We investigate how the evolutionary dynamics of our model system depends on parameters such as population size, mutation rate and recombination. In GAs, mutation rate has been shown to dramatically affect the speed of evolution, with populations evolving at higher rates around an optimal mutation rate that is roughly the reciprocal of the genome length [22, 23]. Biological organisms also often have mutation rates around this optimal value [24, 25]. Recombination has also been shown to increase the speed of evolution on a simple fitness landscape [26]. We study how these evolutionary variables affect the adaptation and discovery times of our self-assembling systems.

An important property of our model is that it is simple enough to allow, in some cases, an exhaustive search of the associated search space, yielding a fully-characterisable but highly non-trivial *fitness landscape* [27, 28] that facilitates a detailed analysis of the underlying evolutionary dynamics.

In addition, we aim to explore the emergence of symmetry in evolving self-assembling systems. It has been observed, for example, that homomeric tetramer protein complexes show a strong preference for dihedral (D_2) symmetry over cyclic (C_4) symmetry [5, 7]. We study this preference as a function of various evolutionary parameters with our simplified polyomino system, for which a complete characterisation of the fitness landscape can be achieved.

This paper is structured as follows. In Section II we describe our model of self-assembling polyominoes, and our implementation of genetic algorithms. In Section III we exhaustively study the search space defined by a particular parameterisation of our model. Section IV analyses how evolutionary variables including mutation rate, population size and search space size affect the dynamics of

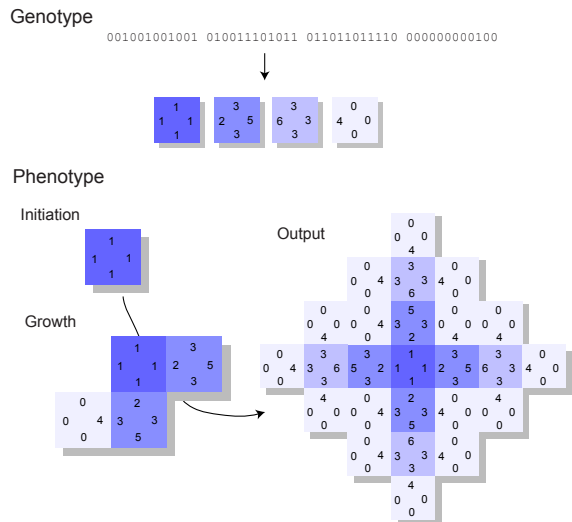


FIG. 2: (colour online) Illustration of polyomino assembly, for a rule set with $n_t = 4$, $n_c = 7$, and with the nucleus and interaction conventions described in the text. A binary representation of a rule set is translated to nucleus, tile and interaction information. The nuclei are placed on a lattice in the initiation stage, and growth progresses stochastically, to a final output possessing shape and tile, but not orientational, determinism: the diagonal neighbours of the central tile have two possible orientations, as their 4 edge can bond to either of the two adjacent 3 edges.

polyomino evolution. In Section V we apply our model to study the evolution of homomeric tetramer protein complexes, and we list our conclusions in Section VI.

II. MODEL & METHODS

A. Model Implementation

Our model uses interacting square tiles to model the self-assembly of 2D polyomino structures on a square lattice [9]. The interactions between adjacent tiles are defined by the nature of each tile's edges, which are assigned 'colours', with any two colours either experiencing no interaction or an attraction. In this conceptual model, there is no energy or temperature scale, so two edges are either non-interacting or have an effectively infinite attractive interaction, making bonding irreversible.

A given assembly scenario will consist of n_t tile types, and an alphabet of n_c available colours. Each tile is entirely specified by a description of its four edge colours. We will denote a tile as an ordered set of four colours, with the first element corresponding to the top edge and subsequent elements corresponding to the edges reached in clockwise order, for example, $\{1, 2, 3, 4\}$. An $n_c \times n_c$ binary interaction matrix A describes the interaction between colours, with colours i and j experiencing an attractive interaction if $A_{ij} = 1$, and no attraction

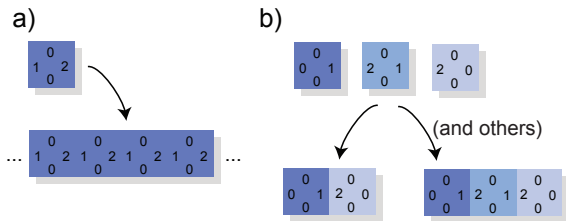


FIG. 3: (colour online) (a) Unbound and (b) non-deterministic rule sets. The interactions in this case follow the convention in Eqn. 1, so edge types 1 and 2 experience an attractive interaction and 0 is neutral. The rule set in (a) consists of one block that is capable of bonding to itself, thus creating an endless chain of repeated blocks. The rule set in (b) contains a block capable of bonding to more than one other, leading to tile non-determinism depending on which bonding block is added first. In this case, the two different bonding tiles create different structures, leading to shape non-determinism.

otherwise. The generalised case of varying interaction strengths has been studied analytically [29], but for simplicity we consider binary interactions.

The tiles are similar to Wang tiles [30], with two important differences: interactions between colours are not limited to each colour bonding only with itself, and the tiles may be rotated to any of the four possible orientations allowed by C_4 symmetry (for example, $\{1, 2, 3, 4\} \equiv \{3, 4, 1, 2\}$). The sides of a tile therefore comprise what is termed ‘an n_c -ary fixed necklace’ of length 4 [9, 31]. The generalisation to free necklaces [31], in which tiles may also be ‘flipped’ ($\{1, 2, 3, 4\} \equiv \{1, 4, 3, 2\}$ and its cyclic variants), will be visited in Section V.

Assembly progresses on an infinite square lattice, and takes place in two phases: initiation and growth (see Fig. 2). The initiation phase involves one or more tiles being placed on the (initially empty) lattice at prescribed positions and orientations: these are the *nucleus tiles*, each of which is described by the tile type of the nucleus, its co-ordinates on the lattice, and its orientation. The combined instruction set representing nucleus, tile edge, and interaction matrix data is the *rule set* for a particular assembly scenario.

There are several alternative schemes for nucleating assembly in this model. Assembly may progress from a single initial tile, laid down at the start of the assembly process. In this case, the single tile may be of a fixed, specific tile type — which we will term a single fixed nucleus (SFN) — or of a tile type arbitrarily chosen from the rule set — which we will term a single general nucleus (SGN). It has been shown that to guarantee deterministic assembly from an arbitrary nucleus tile, considerably more information content is often required within genomes [9].

The question of nucleating tile-based self-assembly has been addressed theoretically [32] and experimentally [18] in the context of algorithmic DNA assembly. In these

studies, seed particles constructed of DNA form the nucleus of a structure and contain information to regulate the assembly process. This approach effectively corresponds to an SFN setup.

We will adopt conventions for the nucleus tiles and the structure of the interaction matrix A , allowing us to simplify the representation of a rule set. We will use an SFN, and take the nucleus tile to be of the tile type first described in the rule set. Furthermore, we fix the orientation of the nucleus tile, so that the edge specified first in the rule set is taken to be the upper edge of the tile when first placed on the grid. Under our convention, the position of the nucleus tile is arbitrary, and polyominoes that differ only by translations are counted as equivalent.

We will usually (with an exception in Section V, which allows the incorporation of self-interacting colours) fix the interaction matrix by defining the interaction between colours i and j (represented by non-negative integers) as:

$$A_{ij} = (1 - i \bmod 2)\delta_{i(j+1)} + (i \bmod 2)\delta_{i(j-1)} \quad (1)$$

so that each colour only interacts with one partner, $1 \leftrightarrow 2$, $3 \leftrightarrow 4, \dots$ and 0 provides a neutral edge, which does not interact with any other edge type.

The combination of conventions for assembly nucleation (SFN, with the first tile specified in the rule set as the nucleus) and the interaction matrix (Eqn. 1) allows us to represent a given rule set by specifying the edges of the tiles involved in assembly alone. Rule sets can then be represented straightforwardly by a binary string (see Fig. 2), by writing each numerical parameter in the rule set (each tile edge) as its binary counterpart and concatenating all the binary variables into one long string. This resulting ‘genome’ is then suitable for processing with genetic algorithms (see Section II C).

Growth progresses stochastically in the following manner. A tile type is chosen randomly from a uniform distribution over the available tiles. A position on the lattice is selected randomly with the constraint that it must be adjacent to a previously laid tile. The chosen tile is cycled in random order through its four possible orientations at the chosen point. If during this cycling the tile experiences an attractive interaction to any of its four neighbouring lattice points, it bonds immediately in that configuration at the chosen site, as illustrated in Fig. 2. In this way, bonding occurs irreversibly, but the model can be generalised to allow reversible interactions by introducing a temperature scale, relaxing the binary constraint on interaction matrix A , and allowing assembly to proceed within a simulation that includes thermal effects.

B. Classes of Assembly Behaviour

Rule sets in our model may result in *unbound* structures: those where the assembly process proceeds in at

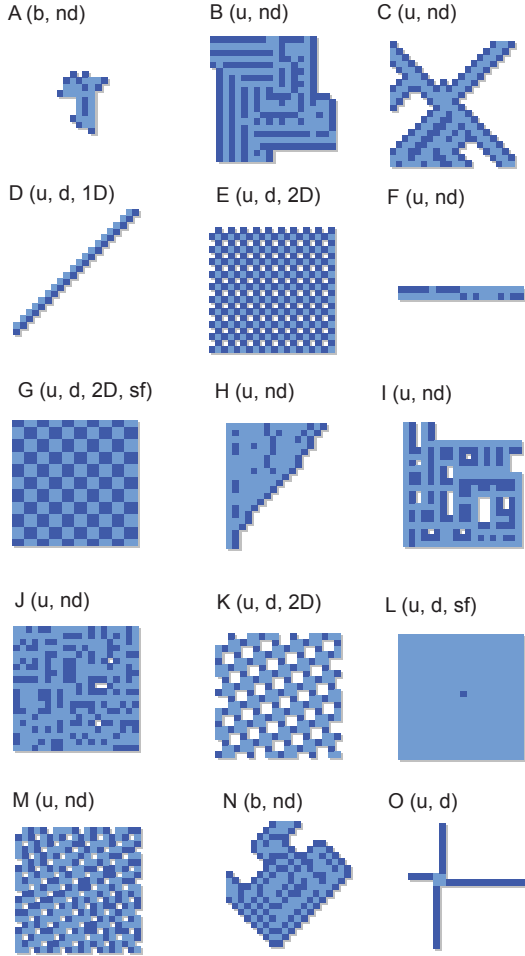


FIG. 4: (colour online) Illustration of some UND polyomino types resulting from growth of genomes with $n_t = 2$, $n_c = 8$. UND polyominoes form the majority of achievable structures. The two colours label the two different tile types that may be involved in assembly. Letters in brackets denote whether the structures are bound (b), unbound (u), deterministic (d), non-deterministic (nd), space-filling (sf) and periodic in one (1D) or two (2D) dimensions.

least one direction without termination. Unbound structures may result, for example, from a set of one or more tiles that bonds to itself repeatedly, forming an endless chain of repeated units, as illustrated in Fig. 3(a).

Self-assembly in biology may also yield unbound structures. Proteinaceous structures consisting of extended sets of repeated units include helical protein filaments such as microtubules [33], actin filaments [34] and tobacco mosaic virus [35]; two-dimensional arrays such as S-layers [36] and purple membranes [37]; and even three-dimensional crystals [38], although some biological mechanism must usually be present to regulate the size of these assemblies and prevent them being truly unbound [39].

Another assembly feature that may result from our model is *non-determinism*, whereby the same set of rules may lead to different structures forming in the growth phase. This non-determinism is due to the inherent

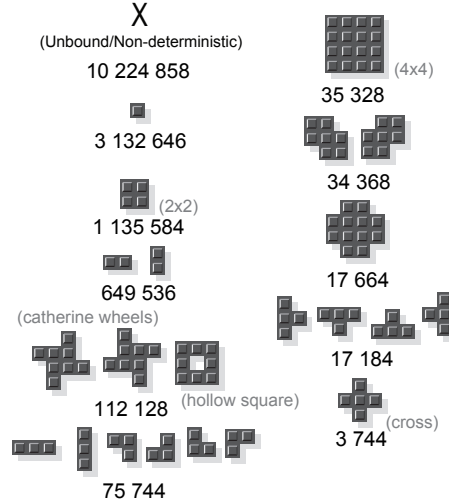


FIG. 5: 61.1% of the polyominoes that can be grown from genomes with $n_t = 2$, $n_c = 8$ are UND structures (X). The rest are bound, deterministic polyominoes. The number of genomes that encode for each polyomino are given – for sets of structures with identical values, each structure occurs the given number of times in genome space. Some structures are given names for ease of reference in the text.

stochasticity in the assembly process. Non-determinism may arise when a tile edge is capable of bonding to more than one other tile edge, which may occur, for example, when the partner to a given edge type appears on more than one tile within the rule set, as in Fig. 3(b).

Non-determinism may occur in several ways. *Shape non-determinism* is the least subtle form, whereby the overall shape of the produced structure (disregarding any detail of tile types, sides and orientations) differs stochastically in different assembly runs (see Fig. 3 (b) for example). *Tile non-determinism* occurs when the same overall structure is achieved for all runs, but sites within the structure are occupied by different tile types stochastically. *Oriental non-determinism* occurs when the structure is both shape- and tile-deterministic, but tiles within the structure differing stochastically in orientation between assembly runs (an example of this is the structure in Fig. 2). Another type of non-determinism, *steric non-determinism*, may also occur as a result of the different speeds of growth in two directions that converge on the same point: if two arms of a structure pass through the same lattice point, the structure will differ depending on which arm arrives there first and hinders growth of the other. This type of non-determinism does not require the multiple bonding edges mentioned above, and is thus hard to detect through observation of the genome.

In biology, non-determinism can also occur in a number of ways. Some closely related proteins coassemble into complexes of well-defined size and shape, but in which the identity of the protein at any position is random. An example of this phenomenon is in the seeds

of pea plants [40] where the legumin protein is formed by a number of paralogous genes, which result in hexamers containing randomly assorted subunits of similar but distinct polypeptide sequences. This example would correspond to tile non-determinism in our model. There also exist examples of shape non-determinism in biology, where proteins, such as certain heat shock proteins, assemble into clusters with a polydisperse distribution of sizes [41, 42].

Finally, our assembly model may yield structures that are *bound* (of finite size) and *deterministic* (in which the self-assembly process always forms the same structure, with a specific shape). The majority of protein quaternary structures fall into this category [39].

For completeness, we note that there is incomplete overlap between the sets of non-deterministic and unbound structures: rule sets may code for outputs that are unbound but deterministic, bound but non-deterministic, bound and deterministic or unbound and non-deterministic. In this study, we will focus on bound, deterministic structures, and will refer to structures not meeting these criteria as UND structures (unbound or non-deterministic). Some examples of these structures are shown in Fig. 4.

All these forms of non-determinism can in theory be detected by running each growth phase a large number k times and comparing the output each time. We shall employ $k = 10$, a value that was confirmed through preliminary investigation to detect most non-deterministic structures while retaining computational speed. In this investigation, we choose tile and orientational determinism as our desirable criterion.

C. Genetic Algorithm Details

Genetic algorithms (GAs) are a class of optimisation procedures that employ operators based on evolutionary biology to reach a solution to some problem [20, 21]. Typically, GAs involve a *population* of N individuals, each representing a trial solution to a problem. A *fitness function* quantitatively measures the performance of an individual at solving the required problem.

GAs take place over a number of time steps or *generations*. Each generation, the fitness function is used to assign a fitness value f_i to each genome i in a population of N individuals. A selection operator is then applied, selecting genomes for reproduction according to their fitness values, with high-fitness genomes being preferentially selected. The N rule sets comprising the next generation are then formed from selected genomes. We employ the roulette-wheel selection method [43], where the probability $P(i)$ of a genome i being selected is proportional to its fitness: $P(i) = f_i / \sum_j f_j$.

A common practise in the implementation of GAs is to preserve a certain number of the fittest individuals in a population from one generation to the next. This approach is termed *elitism*, with a proportion ϵ of fit

individuals preserved, immune to the effects of mutation [21]. We will explore the use of elitism in Section IV F but will generally set $\epsilon = 0$.

GAs may employ *crossover*, modelling recombination. Without crossover, in the asexual regime, new individuals begin as cloned copies of selected genomes. With crossover, modelling sexual reproduction, new individuals are formed by selecting two ‘parent’ rule sets from the old generation, forming a new rule set by combining the rule sets of these parents. The crossover scheme we employ is single-point crossover, where the first L_R bits from one parent and the last $L - L_R$ bits from the other are combined to form a new individual, and L_R is chosen randomly from $[0, L]$.

The implementation of crossover in a simulation is controlled by the crossover rate R , giving the proportion of new genomes that are formed through crossover. For simplicity, we will only employ values of $R = 0$ (corresponding to asexual reproduction) and $R = 1$ (corresponding to sexual reproduction).

Another genetic operator used in GAs is *mutation*, whereby individuals in a generation undergo stochastic changes to their rule sets. We employ *point mutation*, whereby each bit in the genome is flipped with probability μ .

Genomes may contain redundant information, with a tile type being coded for more than once in the binary string. In addition, information on tiles and edges that do not play a role in the assembly of the final structure may be included in the genome. This unused information in genomes allows neutral mutation to progress. A genome may also, in the aforementioned non-deterministic case, code for many different polyomino structures, and the same structure may be produced by more than one genome, providing a many-to-many mapping.

III. SEARCH SPACE ANALYSIS

The process of evolution can be viewed as an optimisation process on the high-dimensional search space defined by all possible genomes [27, 28]. An advantage of our self-assembly model is that the search space for simple structures can be fully characterised. We first investigate the structure of the search space for a polyomino model with two tiles ($n_t = 2$) and up to eight colours ($n_c = 8$), allowing three bond types ($1 \leftrightarrow 2, 3 \leftrightarrow 4, 5 \leftrightarrow 6$), with colours 0 and 7 corresponding to neutral edge types. Each of the 8 tile edges can be represented by $\log_2 8 = 3$ bits, giving a binary genome of length $L = 24$. The search space therefore consists of $2^{24} \simeq 1.6 \times 10^7$ individuals. We will refer to search spaces as \mathcal{S}_{n_t, n_c} , labelled by the number of blocks (tiles) n_t and number of colours n_c , so that the aforementioned search space is $\mathcal{S}_{2,8}$.

We adopt the convention that the first tile encoded in the genome is the assembly nucleus, and its initial orientation is specified by the order in which its edges are encoded, with the top edge first and others following

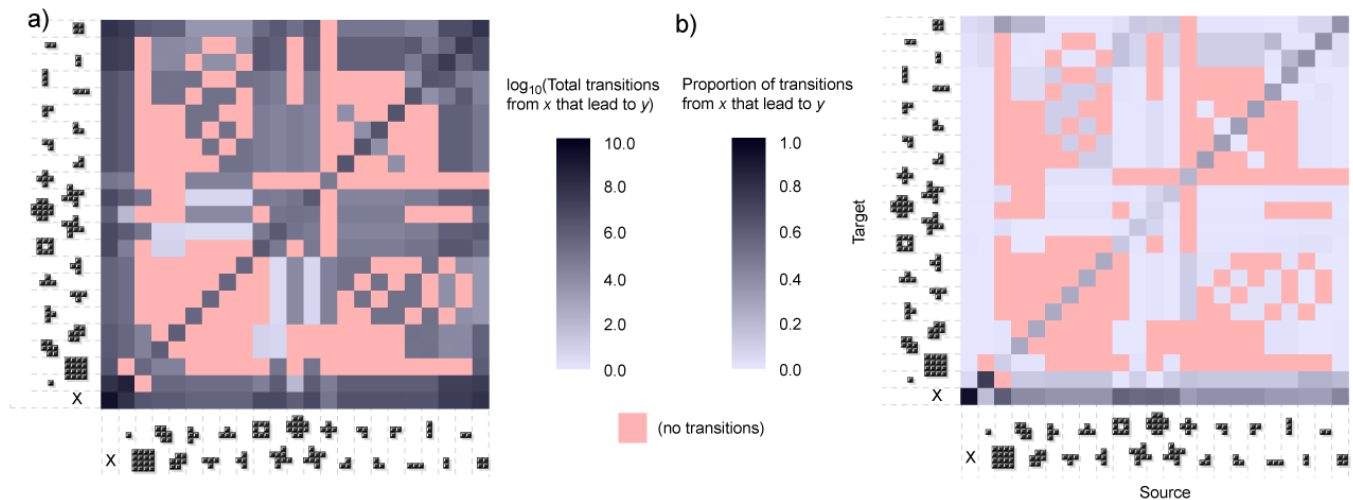


FIG. 6: (colour online) Transitions between polyominoes in the $\mathcal{S}_{2,8}$ ($n_t = 2, n_c = 8$) system. (a) The value of a pixel denotes the total number of single-point mutations that result in a change from phenotype x to phenotype y , over all genotypes in $\mathcal{S}_{2,8}$ that encode x and y . (b) The value of a pixel denotes the average proportion of mutations that cause an $x \rightarrow y$ transition, where the average is taken over all occurrences of x in $\mathcal{S}_{2,8}$. Pink pixels denote transitions between phenotypes that cannot be accomplished with a single mutation.

in a clockwise direction. We then exhaustively evaluate all polyomino structures that may be constructed in this system. The majority are UND structures, some examples of which are shown in Fig. 4 to illustrate the diversity of achievable forms. These structures include non-deterministic, bound structures (for example, A in Fig. 4), deterministic structures that are translationally periodic in one (D) or two dimensions (E, K), some of which may be space-filling (G). Unbound structures displaying shape- but not tile-determinism order also exist (F, M).

The resulting structures are illustrated, along with the volume of search space they occupy, in Fig. 5. Sets of genomes encoding the same phenotype form the *neutral network* of a given phenotype. The large differences in neutral network size corresponding to different phenotypes are related to the differing amounts of information required to specify bonds for different phenotypes. For example, the single tile phenotype only requires an absence of any bonding edges rather than any specific interaction pairs, and correspondingly occupies a large volume of genome space. By contrast, the 4×4 block phenotype requires two interacting pairs of edges, at specific positions relative to each other, and the number of genomes fulfilling these criteria is much smaller.

In addition, all single mutation transitions were recorded, identifying the effect of every possible single mutation on every possible genome — which may change the phenotype or be neutral (with no phenotypic effect). Fig. 6(a) depicts the number of possible single-mutation transitions between different phenotypes, whilst Fig. 6(b) depicts the probability of a transition to another structure given an initial structure. In (b), the total number of transitions between two phenotypes are normalised by the number of genomes encoding the x -axis phenotype

(see Fig. 5). The resulting quantity measures the average number of mutations in a genome that encodes phenotype x that cause a transition to phenotype y .

The Fiedler eigenvalue method [44] was used to arrange the phenotypes in Fig. 6 to maximise the “blockiness” of the resulting matrix by clustering rows and columns whose elements follow similar trends. This method noticeably groups modularly-related polyominoes — for example, the 2×1 and 3×1 structures are clustered together, as several single-mutation changes allow transitions between these structures through the addition or subtraction of a single block. This clustering reflects the fact that pairs of polyominoes that share modules (tiles or bond sequences) are more closely connected in genome space than unrelated structures.

Fig. 6(b) shows that the majority of single mutations from a given phenotype are either *neutral*, preserving the phenotype — leading to high diagonal values in the plot — or cause a transition to a UND or single-tile phenotype. The fraction of neutral mutations is noticeably smaller for larger polyominoes (for example, the ‘catherine wheel’ structures and the 4×4 block have diagonal values under 0.3) than smaller ones (for example, the single tile, 2×1 blocks and the 2×2 block have diagonal values over 0.6), partially because genomes encoding small polyominoes contain more redundant information than those encoding large polyominoes.

Another observable feature of the search space is that, for several phenotypes, the most common result of mutations that are not neutral and do not result in a UND phenotype is a loss of part of the structure associated with the phenotype. For example, a significant proportion of mutations lead from the 4×4 block to the 2×2 block, removing the outer ‘shell’ of tiles. The T-shaped tetrominoes also show many transitions to the L-shaped

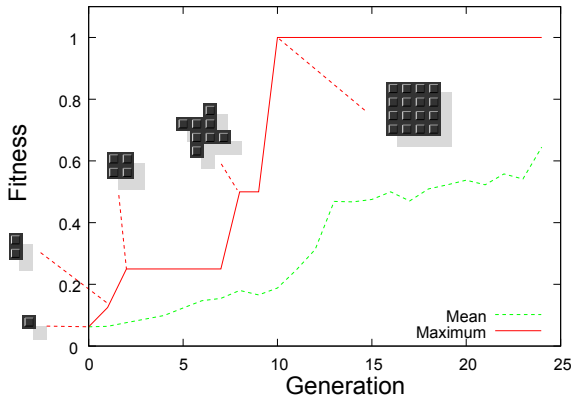


FIG. 7: (colour online) Fitness curves during a typical evolutionary run. A population of genomes is evolved towards a structure of size $s \geq 16$ using the fitness function in Eqn. 2, at $\mu L = 0.5$, $R = 0$, $N = 10$. The plot shows the mean (dashed) and maximal (solid) fitness within a population as time progresses.

triominoes, as one tile is lost from the phenotype. These triominoes in turn show many transitions to the 2×1 blocks, from the loss of another bonded tile.

Fig. 6(b) gives a measure of the average *robustness* and *evolvability* [45] of a given phenotype. The diagonal values give the phenotypic robustness, measuring the average (over all genomes that encode a given phenotype) number of possible mutations that preserve phenotype. This averaging gives a mean phenotype robustness rather than the robustness value for any individual genome [46]. Phenotypic evolvability can be measured in two different ways. Firstly, a sum over off-diagonal values gives the number of mutations that result in a useful (non-UND) phenotypic change. Secondly, the number of non-zero off-diagonal values in a column give the number of different phenotypes that can be accessed from the source phenotype. The first measure can be used to describe the probability that a non-neutral mutation will result in a useful phenotype. The second is more closely related to Wagner’s definition of phenotype evolvability [46]: it measures the diversity of phenotypes accessible from the neutral network of a given phenotype. In our model, robustness and evolvability are related differently in different phenotypes: the catherine wheel structures are highly evolvable according to both the above definitions, but have low robustness (about 0.3), whereas the 2×2 square has high evolvability and high robustness (about 0.6).

IV. EVOLUTIONARY DYNAMICS

A. Evolving Polyomino Size

In evolution, selection drives a system towards high-fitness phenotypes (analogous to a thermodynamic drive

towards low-energy structures), and entropic effects favour those structures that occupy a large proportion of search space. This interplay of fitness and entropic terms is analogous to the concept of free energy in thermodynamics, and indeed several studies have analysed evolution using a ‘free fitness’ quantity [47, 48]. It may be expected that the importance of a given phenotype in evolutionary dynamics is related to several factors, including the fitness of the phenotype and how frequently genomes that produce it occur in the search space. For example, if fitness is defined as proportional to polyomino size, we may expect large structures that occupy a large volume of search space (i.e. with relatively large neutral networks)—like the ‘catherine wheels’ and the 4×4 block in Fig. 5—to play important roles in evolutionary pathways.

Having characterised the $\mathcal{S}_{2,8}$ search space in detail, we now proceed to simulate evolution on a fitness landscape in this search space, with a particular aim being to relate the evolutionary dynamics back to the structure of the underlying search space. We use a specific fitness function to drive evolution towards a given target with a GA. A simple example is evolution towards a bound, deterministic polyomino matching or exceeding a certain size, using the fitness function:

$$F(P_1, P_2, \dots, P_k; s^*) = \begin{cases} 1, & s(P_1) \geq s^* \text{ and all } P_i \\ & \text{identical and bound;} \\ s(P_1)/s^*, & s(P_1) < s^* \text{ and all } P_i \\ & \text{identical;} \\ 0, & P_i \text{ unbound or } P_i \neq P_j \\ & \text{for any } i, j. \end{cases} \quad (2)$$

Here the fitness function takes a set of polyominoes $\{P_1, \dots, P_c\}$ produced through k repeats of the assembly process, and a desired size s^* . The function returns a zero fitness value if the set of polyominoes is UND, and a fitness value proportional to polyomino size for bound, deterministic structures. A value of one means that a solution matching the size criterion has been found.

We note that the previous section suggests that only a small minority of the possible mutations to a genotype lead to a phenotype of larger size. However, it may be expected that on the rare occasions that such mutations do take place, selection will allow these phenotypic changes to be retained and propagate through the population.

Fig. 7 shows the time evolution of a population of polyominoes towards the target $s^* = 16$. On the $\mathcal{S}_{2,8}$ landscape, only one phenotype fulfills this criterion: the 4×4 block. We employ what we will term *zero initial conditions*, in which every bit in every genome at the start of the simulation is set to zero. In the self-assembly implementation described above, this approach means every initial genome encodes a single tile phenotype, which is laid down and incapable of further bonding.

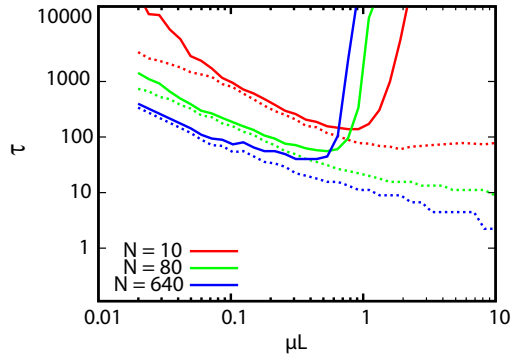


FIG. 8: (colour online) Adaptation time τ (solid lines) and discovery time τ_D (dashed lines) in generations, in $S_{2,8}$ evolving to $s^* \geq 16$, with mutation rate μ , at different N and with $R = 0$.

The simulation begins with the trivial, single-tile phenotype, then quickly ‘discovers’ beneficial interactions, increasing the size of the largest phenotype in the population first to two then to four, with the 2×2 square structure being discovered. The mean fitness lags behind the maximal fitness, as many members of the population will still possess lower fitness values – the mean fitness rises only gradually above the value corresponding to the single tile phenotype, as the information for the 2×2 square structure does not immediately propagate through the whole population. The slow spread of information is due to both the finite fitness advantage resulting from the larger size of the square structure, and the possibility of further mutations leading to UND structures. After several generations, a further beneficial interaction is discovered, creating the ‘catherine wheel’ octomino, and in the next generation this structure is expanded upon to form the 4×4 structure. Note that the catherine wheel structure is one of only a few phenotypes exhibiting a single-mutation transition to the 4×4 block (see Fig. 6).

The discovery of the 4×4 block leads to a sharp rise in the mean fitness, which lasts several generations before flattening. This flattening is due to the non-zero mutation rate and the high transition probability between the 4×4 block and other structures of lower fitness. This mutational entropy means that for a finite μ , perfect adaptation is not reached in this system. The simulation is terminated when more than half the population has maximal fitness.

B. Varying Mutation Rate

In GA experiments, the *discovery time* τ_D measures how long a system takes to produce a single copy of a maximally fit solution, giving an indication of the speed at which evolution progresses. Specifically, τ_D is the first generation in which at least one genome encoding a max-

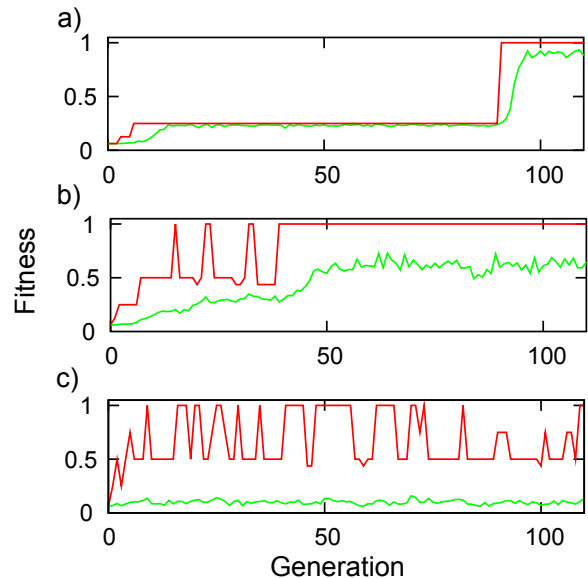


FIG. 9: (colour online) Fitness curves with time for simulations in $S_{2,8}$ evolving to $s^* \geq 16$, with $N = 80$, $R = 0$ and (a) $\mu L = 0.1$, (b) $\mu L = 0.5$, (c) $\mu L = 2$. Maximum fitness in the population is shown in red (upper curves) and mean fitness is shown in green (lower curves). Adaptation, defined as the point where 50% or more of the population has maximal fitness, occurred at generations 95 for (a) and 51 for (b). (c) failed to adapt within the 20 000 generation cutoff.

imally fit solution is present.

The distribution of τ_D in an ensemble of GA experiments is generally observed to be long-tailed, with infrequent occurrences of very high discovery times. Due to computational limitations, we generally employ a cutoff of 20 000 generations in our GA runs. As these rare, high values can skew the mean of such a distribution, we use the median of the distribution as a measure for τ_D , as this statistic is less prone to skew from the rare events and artefacts from the imposed cutoff. 1 000 GA runs were performed for each data point in the following plots.

We measured the value of τ_D in the $S_{2,8}$ system, as a function of mutation rate μ at a range of population sizes N . We set $R = 0$ and use zero initial conditions. Fig. 8 shows the results. τ_D decreases monotonically with μ except in the case of low N , where a slight increase at high μ is observed. The decrease in τ_D at high μ is due to the allowed larger steps across search space and a more explorative search. The slight increase in τ_D at high μ in the low N case may be due to the inability of completely random search to efficiently explore the search space with a small population – in other words, either some memory of previously discovered information or a large population is required for optimal search. We will see in Section IV C that the monotonic decrease in τ_D for larger population sizes is due to the small size of the $S_{2,8}$ search space, and that τ_D exhibits an optimum with μ in larger search spaces.

Another timescale of interest in evolutionary simulations is the *adaptation time* τ of a system, measuring how long a solution, designated as maximally fit, takes to dominate the population. We measure this quantity as the first generation in which more than half the population has maximal fitness. The reason this criterion was chosen is that, due to the high proportion of deleterious mutations that decrease fitness (see Section III), full adaptation is unlikely to occur in reasonably sized populations at finite μ due to the likelihood of at least one phenotype-changing mutation occurring in a population.

Fig. 8 shows τ values for the $\mathcal{S}_{2,8}$ system, with $R = 0$. A general observation is the presence of an optimal mutation rate μ^* , at which τ is a minimum. The optimal mutation rate arises from the following competition. At very low μ , τ increases divergently as μ decreases. This increase in τ at low μ is steeper at low N than at high N . The reason is simply that at low mutation rates, it takes a long time for the system to discover new phenotypes, and this is made worse in smaller populations. On the other hand, arguments from population genetics [49] suggest that full adaptation of a population becomes increasingly difficult for $\mu \gtrsim 1/L$, due to mutational entropy. Thus one expects an optimal mutation rate for adaptation around $\mu \approx 1/L$.

Fig. 9 shows examples of the time evolution of the fitness during simulations at a range of μ values (low: $\mu L = 0.1$, intermediate: $\mu L = 0.5$, high: $\mu L = 2$). At low μ , the mean fitness closely tracks the maximal fitness, as diversity is low and the population is confined around a small region of genome space. The behaviour is due to the high correlation between generations: as little change is introduced to the gene pool through mutation, diversity in the population is low.

At high μ , the mean fitness fluctuates around a low value, dominated by the entropic drive towards common, low-fitness structures (as most mutations are deleterious — see Fig. 6). In this regime, the population is decorrelated, and highly genetically diverse — resembling a random search across genome space.

Behaviour at $\mu L = 1$ is intermediate between these regimes, with some diversity resulting in a rather lower mean fitness than maximal fitness, but a clear relationship between the two showing that information is not being lost through decorrelation.

The relationship between mean and maximal fitness also depends on the robustness of the phenotypes within a population. In Fig. 9 (b), the mean and maximal fitness values are closer in magnitude for a local optimum (around generation 40) than for the global optimum (generation 43 onwards). This difference suggests that the robustness of the global optimum is lower than that of the local optimum, as the population has more difficulty adapting to the fitter phenotype.

We will use the terms *exploration* and *exploitation* to refer to the two regimes observable at high and low μ , respectively. Exploration refers to the random search regime at high μ , where genome space is explored uni-

formly and randomly, and the entropic effect of mutation is too high for the population to become localised and adapt. Exploitation refers to the highly-correlated regime at low μ , where evolution progresses through small changes made to existing information, resembling a “hill-climbing” process with a low diversity. The intermediate μ regime may be thought of as providing a combination of these two effects, with enough exploration to allow escape from local optima and enough exploitation to experience a drive to higher fitness values.

C. Comparing Search Spaces

To investigate the effect of changing the search space for the system, we next considered the $\mathcal{S}_{6,8}$ space, involving $n_t = 6$ blocks rather than the $n_t = 2$ used previously. Genome length is now $L = 72$, with $2^{72} \simeq 4.7 \times 10^{21}$ points in search space, more than 14 orders of magnitude larger than the $\mathcal{S}_{2,8}$ space. We used a sampling approach, investigating 10^8 points in $\mathcal{S}_{6,8}$, to investigate how the structure of this new search space may affect the search for an $s \geq 16$ structure. Firstly, a larger number of genomes in the new space encode for such a structure, with many possible ways of achieving the 4×4 square and other, more diverse structures with $s \geq 16$. However, the associated exponential increase in the overall size of the search space means that a smaller *proportion* of genomes encode a structures with $s \geq 16$, with many more genomes now producing small or UND polyominoes.

Fig. 10 shows the τ and τ_D behaviour with μ in $\mathcal{S}_{6,8}$. In this plot, we see first of all that even though the search space is many orders of magnitude larger, the optimal adaptation and discovery times are at most an order of magnitude larger. The qualitative behaviour of the discovery time τ_D also shows an important difference from the simpler $\mathcal{S}_{2,8}$ system. This measure now exhibits an optimum with μ , generally around $\mu L \geq 1$. At higher μ values, τ_D increases, indicating that the large steps performed by high- μ search in this case are not beneficial. This optimum arises from a tradeoff between exploration and exploitation: the system must have a high enough μ to successfully explore a range of genome space, but must have a low enough μ so that useful information is not lost.

At high μ , the gene pool decorrelates significantly from generation to generation, resulting in loss of information about intermediate-fitness structures that have been discovered. In the smaller $\mathcal{S}_{2,8}$ system, Fig. 8 suggests that this loss of information is not an important effect, as the highly random search afforded by high μ has a finite chance of discovering a suitable solution through exploration alone. However, in the exponentially-larger $\mathcal{S}_{6,8}$ space, random search has a very low probability of discovering a suitable solution, and exploitation of existing information is important in the discovery of better solutions.

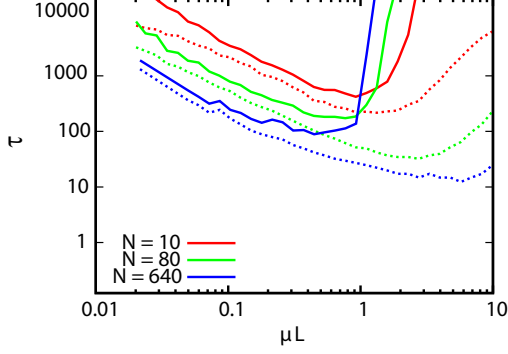


FIG. 10: (colour online) Adaptation time τ (solid lines) and discovery time τ_D (dashed lines) in $\mathcal{S}_{6,8}$ evolving to $s^* \geq 16$, with mutation rate μ , at different N and with $R = 0$.

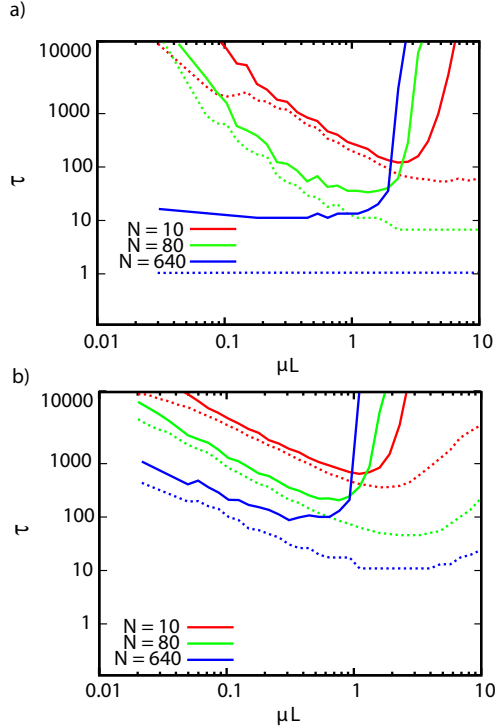


FIG. 11: (colour online) Adaptation time τ (solid lines) and discovery time τ_D (dashed lines) with random initial conditions in (a) $\mathcal{S}_{2,8}$ and (b) $\mathcal{S}_{6,8}$, evolving to $s^* \geq 16$, with mutation rate μ , at different N and with $R = 0$.

D. Initial Conditions

Many studies of evolution employ *random initial conditions*, where the initial population is randomised before numerical simulation [26, 50, 51]. While this picture is appropriate for the modelling of randomly distributed alleles in a population, it is of dubious biological

relevance when bits in a genome represent more fundamental units of genetic information, as it corresponds to an interbreeding population with entirely different, randomised genomes. In considering the evolution of a self-assembling system such as protein quaternary structure [5, 7], it may be that the uniform population of trivial phenotypes afforded by our aforementioned zero initial conditions is more biologically relevant.

To compare the two scenarios, we ran simulations of the $\mathcal{S}_{2,8}$ and $\mathcal{S}_{6,8}$ systems with random, rather than zero, initial conditions. The results (Fig. 11) show a significant departure from our results with zero initial conditions. The difference is particularly pronounced at high N , where the diversity provided by a large population of random genomes will lead to very low discovery times, as space can be explored very quickly from this start point before any adaptation takes place. In fact, the $N = 640$ $\mathcal{S}_{2,8}$ system shows a discovery time of one, as the proportion of search space corresponding to a solution ($35\,328/2^{24} \simeq 2.1 \times 10^{-3}$) is more than $1/N$ ($1/640 \simeq 1.6 \times 10^{-3}$), making it likely that at least one random genome in the initial population will already be a suitable solution. By contrast, this random search effect has little impact in the much larger search space of the $\mathcal{S}_{6,8}$ system.

E. Recombination

We next set $R = 1$, modelling sexual reproduction. This parameterisation was observed to have little effect on the behaviour of τ values in the $\mathcal{S}_{2,8}$ and $\mathcal{S}_{6,8}$ systems with zero initial conditions, leading only to a slight increase in adaptation times for given μ . The effect of setting $R = 1$ with random initial conditions was much more pronounced. In this case, discovery times were significantly reduced and adaptation times were raised in both systems, suggesting that recombination may act to increase the ‘effective mutation rate’ experienced by a genome.

In this picture, recombination may act to decorrelate an offspring from both its parents if the genetic diversity in the population is high. This effect may be, to first order, absorbed into an effective mutation rate dependent on the diversity in the population. Random initial conditions ensure that this diversity is high, particularly for large N , and hence the steps across genome space caused by crossover may be large. This ‘genetic drift’ acts in cohort with the bare mutation rate μ , facilitating rapid discovery of solutions on the small $\mathcal{S}_{2,8}$ search space, but acting to hinder adaptation at higher μ .

F. Elitism

Optimisation-oriented applications of GAs often employ *elitism*. In a population of N individuals with elitism ϵ (where $\epsilon \in [0, 1]$), the fittest ϵN individuals are

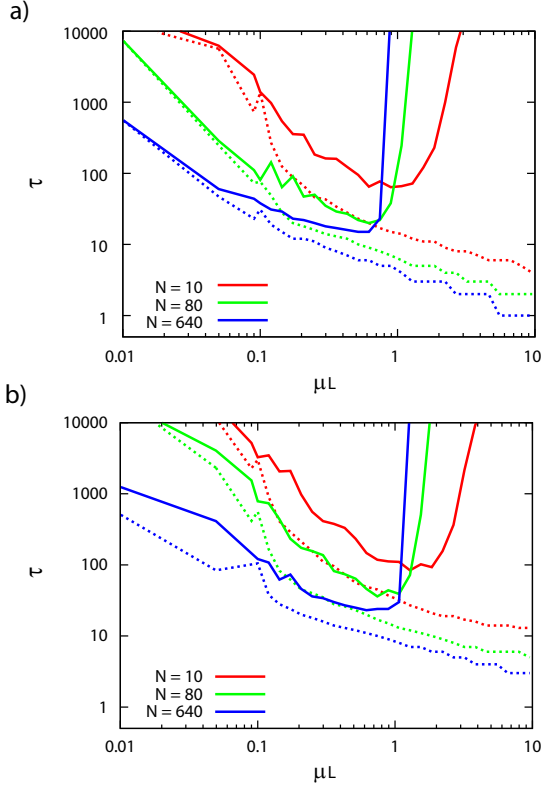


FIG. 12: (colour online) Adaptation time τ (solid lines) and discovery time τ_D (dashed lines) with $\epsilon = 0.1$ for (a) $S_{2,8}$, (b) $S_{6,8}$, with zero initial conditions. The increase in τ_D with high μ for $n_t = 6$ has vanished, and all τ values are lower than the $\epsilon = 0$ equivalents.

preserved totally intact from one generation to the next, immune to the action of mutation and recombination. In this way, the information within the fittest individuals — the location of the highest peak thus far discovered — is preserved, so that decorrelation from this point progresses more slowly and can never be complete. This approach is often beneficial for optimisation as it allows larger μ values to be used — increasing exploration efficiency — without loss of information about the current best solution.

The biological relevance of elitism is questionable. The problem arises from the immunity of the fittest individuals to mutation (and crossover, in a sexually reproducing population). This situation essentially corresponds to a number of extremely long-lived individuals which continually reproduce through their lifetimes, dying only when a fitter solution is found.

Elitism can have a profound effect on the evolutionary dynamics of a model. Fig. 12 shows (μ, τ) curves for a range of evolutionary scenarios with $\epsilon = 0.1$. These effects include a general reduction in τ values, showing that elitism is a useful tool in pure optimisation application of GAs. The increase in τ_D with high μ on $S_{6,8}$ is no longer

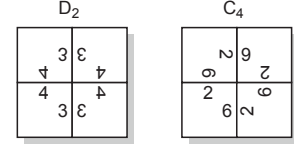


FIG. 13: Illustration of D_2 and C_4 symmetries in homomeric tetramers.

observed, as elitism retains information from one generation to the next, meaning that the search never becomes fully random. In experiments with recombination (not pictured), elitism also acts to stabilise the population, with adaptation observed in $\epsilon = 0.1$ simulations in some regimes that struggled to adapt with $\epsilon = 0$.

V. HOMOMERIC PROTEIN TETRAMERS

It has been estimated that between 50 and 70% of proteins form homomeric clusters *in vivo* [52]. These complexes are usually symmetrical, with each protein in an identical environment. Homomeric tetramers, for example, may display cyclic symmetry (C_4) or dihedral symmetry (D_2). The C_4 geometry involves only one type of interaction, whereas the D_2 complex involves at least two self-complementary interactions. In an important recent study by Levy *et al.* [5], it was shown that dihedral complexes are over 10 times more abundant than cyclic complexes with the same number of subunits. Moreover, these authors found that the evolutionarily older interactions are typically stronger than the more recently evolved patches, and that the clusters dissassembled in a hierarchical fashion, with the newer (and weaker) bonds breaking first.

The relationship between the strength of the patches and their evolutionary history, as well as the observed hierarchical dissassembly can be rationalized with simple statistical mechanical models [7]. Similarly, the preference for dihedral over cyclic symmetry has been linked to the fact that for D_2 structures, two pairs of identical edges bond (requiring self-complementary interactions or *homointeractions*), whereas in C_4 structures, one pair of different edges bond (using non-self-complementary interactions or *heterointeractions*). Statistical models of the formation of homointeractions and heterointeractions have shown that the former have a wider distribution of energies than the latter. It has been suggested that this wide distribution makes stable low-energy bonds easier to evolve using homointeractions than heterointeractions which may result in a biological preference for D_2 structures [53–55]. Another reason for the preference for D_2 may be that evolution does not need to proceed to a tetramer structure in a single step, but can go through a dimeric intermediate.

Our simple polyomino model cannot be used in its current form to study the strength of patches, and by extension, the hierarchical assembly/disassembly. However, it can be used to investigate the effect of homo/heterointeractions and evolutionary intermediates on the evolutionary preference for D_2 over C_4 . In order to model this system, we must generalise our model to allow tetrameric structures to form in both symmetry configurations, as shown in Fig. 13. To do this, we allow building block tiles to ‘flip’, so that, for example, tiles $\{1, 2, 3, 4\}$ and $\{1, 4, 3, 2\}$ are equivalent. The sides of building blocks now correspond to free, rather than fixed, necklaces [31]. This condition reflects the fact that homointeraction interfaces require a rotation by π radians with respect to each other to form a bond.

We first investigate the case where heterointeractions are equally easy to evolve as homointeractions. To achieve this, we choose a new interaction matrix such that the bonding pairs are: $3 \leftrightarrow 3, 4 \leftrightarrow 4, 2 \leftrightarrow 6$, with all other colours neutral. This setup was chosen so that, given zero initial conditions, the formation of two self-interacting edges involves the same number of mutations as the formation of a non-self-interacting bonding pair. Specifically, the discovery of colours 3 and 4 (011 and 100) or 2 and 6 (010 and 110) are equally likely, each requiring three beneficial mutations. We label this new search space \mathcal{S}'_{18} , with a characteristic number of self-interactions $n_{si} = 2$. We note that, given that $n_t = 1$ for this system, there is no distinction between the SFN and SGN cases mentioned in Section II A.

In a similar manner to that used for the $\mathcal{S}_{2,8}$ system in Section III, we can evaluate all possible structures in this new search space and the possible transitions between phenotypes (see Fig. 14). There are 4096 different possible genotypes, which are distributed among the possible phenotypes as shown in Table I. A completely random search would thus display a D_2 structure frequency of 0.68. While the interactions are chosen so that the minimal number of mutations required to reach a D_2 structure from zero initial conditions is the same as that required to reach a C_4 structure, the redundancy available to D_2 genomes (which may contain, for example, one unpaired heterointeraction in addition to their homointeractions) gives D_4 a higher search space volume than that of the less redundant C_4 structures. We can also map out all the pathways between different phenotypic states, as shown in Fig. 14.

To study the dynamic effects of the structure of search space, we simulated a population of 10^5 random walkers in genome space. Each walker started from zero initial conditions, and then took mutational steps until a genome encoding one of the two tetrameric states was reached. A mutational step involved an application of the mutation operator from a GA, rather than enforcing exactly one mutation per step. Walks were terminated and ignored if they reached the UND state (something that mirrors what might happen in nature where this usually would be lethal for the organism).

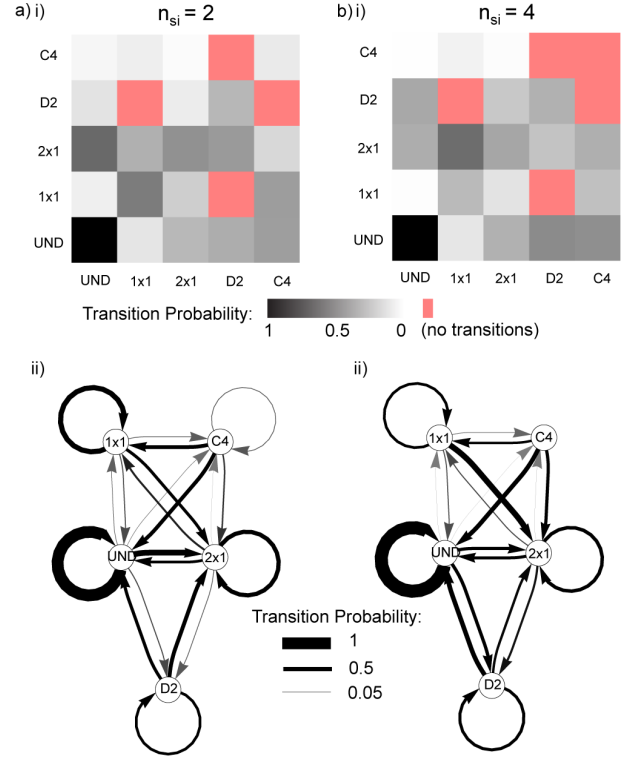


FIG. 14: (colour online) Transition probabilities for $\mathcal{S}'_{1,8}$. (a) Number of self-interacting colours $n_{si} = 2$ and (b) $n_{si} = 4$. (i) Transition probabilities between phenotype x and phenotype y . (ii) Transition probabilities represented in a network between phenotypes. The edge widths are proportional to their probability.

	UND	Monomer	Dimer	D_2	C_4
$n_{si} = 2$	1 214	994	1 488	272	128
$n_{si} = 3$	1 829	431	1 212	552	72
$n_{si} = 4$	2 510	146	736	672	32

TABLE I: Number of genomes in the $\mathcal{S}'_{1,8}$ search spaces that encode different structures.

A similar random walker analysis is possible in phenotype space, on the network in Fig. 14 a) ii). Here, each random walker occupies a node in the network, and may, at each time-step, undergo a transition between nodes according to the weight of the connecting edge. A population of walkers was initialised at the monomer node and allowed to walk, with UND encounters being terminated and ignored. The results of both these walker simulations are shown in Table II.

To test the effect of a dimer intermediate on the probability of obtaining a D_2 or a C_4 structure, we also used a GA to run an evolutionary simulation. We employed two different fitness functions, representing two situations: dimers possessing either no fitness advantage or a large fitness advantage over monomers. As the only pos-

	SSP	GW	PW	GA, FF A	GA, FF B
$n_{si} = 2$	0.68	0.58	0.48	0.55	0.67
$n_{si} = 3$	0.88	0.75	0.67	0.69	0.80
$n_{si} = 4$	0.95	0.86	0.75	0.96	0.98

TABLE II: Proportion of random walker and GA simulations on $S'_{1,8}$ that result in a D_2 structure being discovered before a C_4 structure. Columns are Search Space Proportion (SSP), defined as the number of genomes encoding D_2 structures divided by the total number of tetramer genomes, Genotype Walker (GW), Phenotype Walker (PW), and GAs, with FF denoting fitness function, as described in the text. n_{si} is the number of self-interacting colours in the rule set. GAs were run with $N = 80, \mu L = 0.5$.

sible phenotypes in this landscape are UND and $s = 1$, $s = 2$, $s = 4$, we represent a fitness function with the values awarded to these four cases respectively. Fitness function A gives no advantage to dimer formation: $F(\text{UND}) = 0, F(1) = 0.1, F(2) = 0.1, F(4) = 1$. Fitness function B gives a large fitness advantage to dimer formation: $F(\text{UND}) = 0, F(1) = 0.1, F(2) = 0.9, F(4) = 1$. We ran 10^4 simple GAs for each case, with $N = 80$ and $\mu L = 0.5$, and measured the proportion of times a run discovered (rather than adapted to) either a C_4 or a D_2 phenotype. Table II shows the results of simulations with these fitness functions.

We then introduced an evolutionary bias towards homointeractions by allowing more colours to self-interact. To this end, we first include $1 \leftrightarrow 1$ (giving $n_{si} = 3$) and then $5 \leftrightarrow 5$ bonds (giving $n_{si} = 4$). This addition of homointeractions changes the evolutionary landscape dramatically (see Fig. 14 b)). The distribution of phenotypes is shown in Table I. The number of UND genotypes is observed to increase with n_{si} , due to the greater number of genomes that encode extended, unbound structures in systems with large numbers of self-interactions.

Table II shows a comparison of the $D_2 : C_4$ ratio expected from search space structure with results for walker and GA simulations on these systems. A number of trends can be observed in Table II. Although the interactions are chosen so that the minimum number of mutations required to reach a D_2 structure from zero initial conditions is the same as that for a C_4 structure, D_2 structures appear more frequently, which is commensurate with the fact that they occupy a larger proportion of search space.

However, the proportion of runs that first discover D_2 structures is lower than expected from the search space structure for genotype walkers, and lower still for phenotype walkers. The slightly lower proportion for genotype walkers is due to the starting point of the simulations: monomers, of all possible phenotypes, display the highest transition probability to C_4 structures, so C_4 discovery is more likely from the zero initial conditions we employ (encoding a monomer) than from a random start point.

The significantly lower D_2 proportion from phenotype

walkers is due to the shorter length of the monomer \rightarrow tetramer pathways, which requires only one transition, whereas monomer \rightarrow dimer $\rightarrow D_2$ requires two. In this case, the phenotype representation has masked the genetic detail whereby the minimal number of steps required to reach D_2 and C_4 structures from zero initial conditions are identical. The steps in the minimal monomer $\rightarrow C_4$ pathway involve one neutral monomer step ($0000 \rightarrow 0002$) and one phenotype-changing monomer $\rightarrow C_4$ step ($0002 \rightarrow 0062$), whereas both steps to reach a D_2 structure are phenotype-changing ($0000 \rightarrow 0003 \rightarrow 0043$). The observed difference is an illustration of the influence of a complex genotype-phenotype map. In this case, information is lost when mutational steps across a neutral network are disregarded.

The proportion of GA runs with fitness function A that identify a D_2 structure is lower than expected in comparison to the genotype walkers for $n_{si} = 2$ and $n_{si} = 3$. This difference arises from the different amounts of time required for a GA to identify D_2 and C_4 structures. For $n_{si} = 2$ and 3, it is observed that the mean discovery time for C_4 structures is lower than the mean discovery time for D_2 structures. A GA reports the structure it first discovers, whereas a set of genotype walkers reports the proportion of structures discovered regardless of the relative time taken to reach these structures. The lower C_4 mean discovery time for low n_{si} GAs therefore results in more C_4 structures being reported than in the genotype walkers. For $n_{si} = 4$, the mean discovery time for D_2 structures is lower than that for C_4 in GAs, reflected in the higher observation of D_2 structures in these GA simulations. Note that if the GWs were run in parallel sets, and the set was stopped at the first discovery of a tetramer, this would also favour C_4 for $n_{si} = 2, 3$ and D_2 for $n_{si} = 4$.

Another effect that acts to change the expected $D_2 : C_4$ ratio arises from UND structures. In a GA, genomes encoding UND structures will be replaced (due to their zero fitness) by a copy of another genome chosen by selection. This replacement genome will be either a monomer or a dimer, according to the current state of the GA population. As n_{si} increases, or if fitness function B is used, the population becomes more likely to contain dimers, due respectively to their increased presence in search space and their increased fitness. If UND genotypes are replaced by monomers, C_4 discovery will be more likely (the case at low n_{si}). If they are replaced by dimers, D_2 discovery will be more likely (the case at high n_{si}).

Another noticeable result is that conferring a fitness advantage to dimers increases the proportion of D_2 structures discovered in GAs. This increase is due to selection favouring dimers in the evolving population, from which situation the dimer $\rightarrow D_2$ transition is most likely.

The above GA results concern the discovery of tetramers rather than adaptation of the population to tetramers. When adaptation was considered, the $n_{si} = 2$ and 3 trends remained very similar. The $n_{si} = 4$ sys-

tem became 100% dominated by D_2 tetramers, as the genomes encoding C_4 structures in this system were individual and isolated. In other words, they exhibit low phenotypic robustness and adaptation proved impossible with such small neutral network sizes.

We note that the coarse-grained nature of our model greatly simplifies the description of protein surfaces. In proteins, interacting sites consist of multiple amino acid residues, rather than a single colour type as we employ. Point mutations in reality will normally alter not more than one constituent amino acid of a bonding site, rather than entirely changing the bonding characteristics of an interaction site. In addition, the spatial structure of protein complexes is vastly more complicated than the simple 2D tile geometry we employ here. However, this simple system nonetheless displays interesting dynamic behaviour. We show that favouring homo-interactions in the search space, and favouring dimers in the fitness function, can both significantly enhance the proportion of D_2 tetramers over C_4 tetramers. By performing a complete enumeration of the fitness landscape, we can also uncover some subtle questions related to the underlying structure of the landscape. For example, considering only the phenotype structure can mask important genotypic structure that influences the evolutionary dynamics.

VI. CONCLUSIONS

We have studied the evolutionary dynamics of self-assembling polyominoes. We focussed on deterministic self-assembly – where a given rule set always leads to the same polyomino structure – because an analogy can be made with monodisperse self-assembly seen in nature, for example in protein quaternary structure.

Although our model is simple enough to be easily tractable with modest computational resources, it exhibits rich evolutionary behaviour that is linked to its non-trivial genotype-phenotype mapping. The evolutionary dynamics can be viewed as a search performed by a population of individuals on a complex fitness landscape. An advantage of the polyomino system is that in some cases this landscape can be fully enumerated and classified in terms of adjacent structures and the transitions between them. Such information helps explain some of the detailed behaviour observed in GA simula-

tions. Properties like robustness and evolvability [45] can easily be extracted from the fully enumerated landscapes.

We also investigated the effect of changing the mutation rate, the population size, and the size of the search space on adaptation and discovery times for the evolution of certain classes of polyominoes. We find that there is an optimal, intermediate mutation rate value for adaptation. For smaller μL the system takes longer to discover the desired phenotypes, whereas for larger μL the mutational entropy prevents it from adapting to the right phenotype.

For smaller spaces and larger populations the discovery time keeps decreasing with increasing mutation rate, but for larger spaces, there is also an optimal mutation rate for the discovery time. This effect can be cast into the language of *exploration and exploitation* [56–58]. For low μL , the system can only take small steps across the search space, leading to confinement of the gene pool around fitness optima [27], low diversity, and slow *exploration* of surrounding space. At high μL , the system de-correlates very quickly, reducing its ability to *exploit* beneficial mutants through further small changes, raising diversity to almost the level expected for a randomised population. The search’s hill-climbing ability is decreased as large steps randomise the gene pool very quickly.

The modelling of evolutionary processes with genetic algorithms is complicated by the fact that the number of parameters that can be varied is very large. One advantage of our polyomino system is that the effects of varying the GA parameters can be easily quantified. We studied some popular parameter choices, and argue that, for example, the use of random initial conditions or elitism may not be the most biologically relevant way to parameterise a genetic algorithm.

Finally, we studied the evolution of polyomino tetramers, inspired by recent work on the structure and evolution of homomeric protein tetramers [5, 7]. In nature there is a strong preference of D_2 over C_4 symmetries, and we show that both an increase in the probability of homointeractions as well as a fitness advantage of dimeric intermediates can strongly favour the formation of D_2 symmetry. Our simplified model shows that the outcome of evolutionary dynamics is affected by the topology of the search space, including emergent properties like phenotypic robustness.

-
- [1] D. S. Goodsell, *Bionanotechnology: Lessons from Nature* (Wiley-Liss, Hoboken, NJ, 2004).
 - [2] J. B. Pereira-Leal, E. D. Levy, and S. A. Teichmann, *Phil. Trans. Roy. Soc. B* **361**, 507 (2006).
 - [3] A. E. Hirsh and H. B. Fraser, *Nature* **411**, 1046 (2001).
 - [4] J. P. K. Doye, A. A. Louis, and M. Vendruscolo, *Phys. Biol.* **1**, 9 (2004).
 - [5] E. D. Levy, E. B. Erba, C. V. Robinson, and S. A. Teichmann, *Nature* **453**, 1262 (2008).
 - [6] A. W. Wilber, J. P. K. Doye, A. A. Louis, E. G. Noya, M. A. Miller, and P. Wong, *J. Chem. Phys.* **127**, 085106 (2007).
 - [7] G. Villar, A. W. Wilber, A. J. Williamson, P. Thiara, J. P. K. Doye, A. A. Louis, M. N. Jochum, A. C. F. Lewis, and E. D. Levy, *Phys. Rev. Lett.* **102**, 118106 (2009).
 - [8] J. Andreani, A. W. Wilber, J. P. K. Doye, and A. A. Louis, in preparation (2011).

- [9] S. E. Ahnert, I. G. Johnston, T. M. A. Fink, J. P. K. Doye, and A. A. Louis, Phys. Rev. E **82**, 026117 (2010).
- [10] A. Chworos, I. Severcan, A. Y. Koyfman, P. Weinkam, E. Oroudjev, H. G. Hansma, and L. Jaeger, Science **306**, 2068 (2004).
- [11] E. Winfree, F. Liu, L. A. Wenzler, and N. C. Seeman, Nature **394**, 539 (1998).
- [12] M. Cook, P. W. K. Rothmund, and E. Winfree, Lect. Notes Comput. Sci. **2943**, 1979 (2004).
- [13] P. W. K. Rothmund, N. Papadakis, and E. Winfree, PLoS Biology **2**, e424 (2004).
- [14] C. Lin, Y. Liu, S. Rinker, and H. Yan, ChemPhysChem **7**, 1641 (2006).
- [15] D. B. Lukatsky, B. M. Mulder, and D. Frenkel, J. Phys.: Condens. Matt. **18**, S567 (2006).
- [16] H. Wang, in *Proceedings of the Symposium on Mathematical Theory of Automata, at Polytechnic Institute of Brooklyn* (Polytechnic Press of the Polytechnic Institute of Brooklyn; distributors: Interscience Publishers, New York, 1963), p. 23.
- [17] P. W. K. Rothmund and E. Winfree, in *Proceedings of the thirty-second annual ACM symposium on Theory of computing* (ACM, 2000), pp. 459–468.
- [18] R. D. Barish, R. Schulman, P. W. K. Rothmund, and E. Winfree, Proc. Natl. Acad. Sci. USA **106**, 6054 (2009).
- [19] E. Winfree and R. Bekbolatov, Lect. Notes Comput. Sci. **2943**, 1980 (2004).
- [20] D. E. Goldberg, *Genetic Algorithms in Search and Optimization* (Addison-Wesley, Reading, MA, 1989).
- [21] M. Mitchell, *An Introduction to Genetic Algorithms* (MIT Press, Cambridge, MA, 1998).
- [22] K. A. De Jong and W. M. Spears, Parallel Prob. Solv. Nature **1**, 38 (1990).
- [23] T. Bäck, in *Proceedings of the 5th International Conference on Genetic Algorithms* (Morgan Kaufmann, San Francisco, CA, 1993), p. 2.
- [24] D. Metzgar and C. Wills, Cell **101**, 581 (2000).
- [25] J. W. Drake, B. Charlesworth, D. Charlesworth, and J. F. Crow, Genetics **148**, 1667 (1998).
- [26] E. Cohen, D. A. Kessler, and H. Levine, Phys. Rev. Lett. **94**, 098102 (2005).
- [27] S. Wright, Proc. 6th Intl. Congress on Genetics p. 356 (1932).
- [28] J. Maynard Smith, Nature **225**, 563 (1970).
- [29] D. Soloveichik and E. Winfree, SIAM J. Comput. **36**, 1544 (2007).
- [30] H. Wang, AT&T Tech. J. **40**, 1 (1961).
- [31] *Necklace. From MathWorld—A Wolfram Web Resource.* <http://mathworld.wolfram.com/Necklace.html>.
- [32] R. Schulman and E. Winfree, SIAM J. Comput. **39**, 1581 (2009).
- [33] L. A. Amos, Org. Biomol. Chem. **2**, 2153 (2004).
- [34] E. Reisler and E. H. Egelman, J. Biol. Chem. **282**, 36133 (2007).
- [35] W. K. Kegel and P. Van Der Schoot, Biophys. J. **91**, 1501 (2006).
- [36] M. Sára and U. B. Sleytr, J. Bacteriol. **182**, 859 (2000).
- [37] M. P. Krebs and T. A. Isenbarger, BBA-Bioenergetics **1460**, 15 (2000).
- [38] J. P. K. Doye and W. C. K. Poon, Curr. Opin. Colloid In. **11**, 40 (2006).
- [39] D. S. Goodsell and A. J. Olson, Annu. Rev. Biophys. Biomol. Struct. **29**, 105 (2000).
- [40] E. Stoger, M. Parker, P. Christou, and R. Casey, Plant Physiol. **125**, 1732 (2001).
- [41] A. Laganowsky, J. L. P. Benesch, M. Landau, L. Ding, M. R. Sawaya, D. Cascio, Q. Huang, C. V. Robinson, J. Horwitz, and D. Eisenberg, Protein Sci. **19**, 1031 (2010).
- [42] J. L. P. Benesch, J. A. Aquilina, A. J. Baldwin, A. Rekas, F. Stengel, R. A. Lindner, E. Basha, G. L. Devlin, J. Horwitz, E. Vierling, et al., Chem. Biol. **17**, 1008 (2010).
- [43] D. E. Goldberg and K. Deb, in *Foundations of Genetic Algorithms* (Morgan Kaufmann, San Mateo, CA, 1991), vol. 1, p. 69.
- [44] C. P. Massen and J. P. K. Doye, Arxiv preprint cond-mat/0610077 (2006).
- [45] A. Wagner, *Robustness and evolvability in living systems* (Princeton University Press, Princeton, NJ, 2005).
- [46] A. Wagner, Proc. Roy. Soc. B **275**, 91 (2008).
- [47] Y. Iwasa, J. Theor. Biol. **135**, 265 (1988).
- [48] G. Sella and A. E. Hirsh, Proc. Natl. Acad. Sci. USA **102**, 9541 (2005).
- [49] M. A. Nowak, *Evolutionary Dynamics: Exploring the Equations of Life* (Harvard University Press, Cambridge, MA, 2006).
- [50] N. Kashtan, E. Noor, and U. Alon, Proc. Natl. Acad. Sci. USA **104**, 13711 (2007).
- [51] P. Oikonomou and P. Cluzel, Nature Phys. **2**, 532 (2006).
- [52] E. D. Levy, J. B. Pereira-Leal, C. Chothia, and S. A. Teichmann, PLoS Comput. Biol. **2**, e155 (2006).
- [53] D. B. Lukatsky, K. B. Zeldovich, and E. I. Shakhnovich, Phys. Rev. Lett. **97**, 178101 (2006).
- [54] I. André, C. E. M. Strauss, D. B. Kaplan, P. Bradley, and D. Baker, Proc. Natl. Acad. Sci. USA **105**, 16148 (2008).
- [55] G. E. Schulz, J. Mol. Biol. **395**, 834 (2010).
- [56] J. March, Organ. Sci. **2**, 71 (1991).
- [57] A. Eiben and C. Schippers, Fund. Inform. **35**, 35 (1998).
- [58] J. H. Holland, *Adaptation in natural and artificial systems* (MIT Press Cambridge, MA, USA, 1992).