Chapter 4

Evolutionary Design of a Model of Self-Assembling Chemical Structures

Andrew Buchanan[a,b], Gianluca Gazzola[a,b], Mark A. Bedau[a,b,c,1]

[a] *ProtoLife Srl, Via della Libertà 12, Marghera, Venice 30175, Italy*
[b] *European Center for Living Technology, Ca' della Zorza, Dorsoduro 3859, Venezia 30125, Italy*
[c] *Reed College, 3203 SE Woodstock Blvd., Portland, OR 97202, USA*

Abstract. We introduce a new variant of dissipative particle dynamics (DPD) models that include the possibility of dynamically forming and breaking strong bonds. This model exhibits different forms of self-assembled processes; some like micelle formation involve only weak bonds, and others like the ligation of oligomers involve both weak and strong bonds. Complex self-assembly processes are notoriously difficult to design and program. We empirically demonstrate an evolutionary algorithm that optimizes self-assembly processes like micelle formation and template-directed ligation.

1. Design of Self-Assembling Chemical Systems

Self-assembly processes involve a population of independently acting components that are each governed only by simple, usually local, rules. The systems exhibit emergent collective behavior that produces characteristic kinds of stable aggregate structures. These structures include "soft" materials like micelles and vesicles formed by weak associative forces such as hydrogen bonds, but also structures like polymers produced with "hard" covalent bonds. This chapter illustrates an evolutionary approach to designing self-assembling structures formed through the operation of both weak associative forces and covalent bonds.

Life forms exhibit particularly rich examples of such self-assembled chemical systems. The emergence of life from pre-biotic chemical systems involved harnessing chemical self-assembly processes for specific functions. Examples include the chemical coupling of amphiphile systems that self-assemble into cellular containers, and chemical self-replicating systems of genetic polymers that self-assemble through templating. According to some scenarios, early amphiphiles self-assembled into vesicles, which eventually encapsulated replicating chemical microsystems [38]. Such cooperating self-assembling chemical systems were then filtered by a selection process, in which those that prolonged vesicle

[1] E-mail: mark@protolife.net.

STUDIES IN MULTIDISCIPLINARITY
VOLUME 5 ISSN 1571-0831/DOI: 10.1016/S1571-0831(07)00004-4

lifetime or caused it to quickly grow and divide, would eventually dominate the population. Achieving this complementary reproduction of amphiphilic containers and genetic polymers is also a key milestone in the development of artificial cells [32].

The traditional method of designing self-assembling chemical systems relies on rational design by chemical experts on the basis of chemical first principles, applied to the simplest and purest compounds. But this method breaks down for inherently complicated chemical systems, including many of those involving self-assembly. Design of self-assembling systems is inherently difficult because their behavior is emergent. Desired emergent properties cannot be directly programmed into the system but must arise indirectly, as a result of directly programming lower level interactions. There is no method to deduce the resulting emergent behaviors short of actually examining the system's behavior. So, a general ability to program self-assembling chemical systems would have immense practical significance.

Evolutionary design with an evolutionary algorithm has been shown to provide greater flexibility, lower cost and feasibility in areas in which traditional rational design is ineffective [28,37,45,46]. Hence, we here employ an evolutionary algorithm to design self-assembling chemical systems. Earlier we empirically demonstrated evolutionary design of self-assembling structures in a computer simulation of amphiphiles in water [3]. Here we present evolutionary design of simulations of two kinds of chemical self-assembly: micelle formation and template-directed ligation. The simulation framework we employ is notable for its realistic portrayal of certain self-assembly processes.

The evolutionary algorithm that here optimizes computer simulations of chemical structures has also optimized real self-assembling chemical structures in a wet lab [39]. Together these results illustrate the broad applicability of our evolutionary design method.

Evolutionary design of a self-assembling system is somewhat similar to certain other design processes, specifically combinatorial chemistry, *in vitro* or directed evolution, and evolutionary design of physical systems through a proxy model. But the similarities are only superficial.

Combinatorial chemistry [8,12,34] does not employ true evolutionary search. Vast libraries of chemicals are synthesized and then screened for matching some target, but the chemical possibilities are determined beforehand by the size and composition of the library which is determined in advance. If the library does not contain a match, then combinatorial chemistry will fail. Combinatorial chemistry is useful for mining existing chemical databases for new uses. However, it breaks down in the face of larger polynomial search spaces or spaces for complex chemical interactions. By contrast, an evolutionary algorithm can find good solutions in a range of possibilities after testing only a small fraction of the possibilities.

In vitro or directed evolution [9,11,22,23,35,42,47] searches for desired functionality in combinatorial libraries of DNA or RNA. The target functionality is often the production of desired enzymes. These DNA or RNA libraries continually evolve, through random mutations of selected molecules. This truly evolutionary search is a more creative than combinatorial chemistry screening. However, *in vitro* evolution can be applied only to chemical functionality encoded in nucleic acids. The method we demonstrate here can be used to design any sort of desired chemical functionality, in principle.

Indirect design of target systems via explicit evolutionary design of a simulation of the target system is often used to design physical objects such as robots [10,13,28,31]. This method suffers from an inherent “reality gap” because the simulation will typically lack some relevant characteristics of the target system. In contrast, our method always measures fitness directly in the target system (here, the target is itself a computer model). So no “reality gap” can afflict our method.

2. Dynamic-Bonding Dissipative Particle Dynamics (dbDPD)

For our investigation we used a model of chemical reaction systems based on the well-studied dissipative particle dynamics (DPD) framework [18,21,30,36, 41,43]. The DPD framework is a mesoscopic system simulator meant to bridge the gap between molecular dynamics (MD) models and continuous substance models. The extreme computational demands of MD models make them appropriate only for simulating small systems for brief intervals—orders of magnitude smaller than the time and length scales of interest here. Continuous substance models are inappropriate as models of molecular scale systems in which the discrete nature of particles impacts the dynamics of the system.

In DPD, the equations of motion are of second order, with explicit conservation of momentum, in contrast to Langevin or Brownian dynamics. Solvent molecules may be represented explicitly, but random and dissipative forces are included in the dynamics to compensate for the dynamical effects of replacing the hard short-range potentials of MD by softer potentials in DPD simulations. This procedure allows a major acceleration of the simulation compared with MD.

In traditional DPD models, all bonds are specified initially, and subsequently cannot form or break. One limitation imposed on the DPD simulations discussed here is that each element can have at most two strong bonds at a given time. The particles also interact in a manner corresponding to weak forces such as van der Waals forces or hydrogen bonds. In contrast to real systems, weak forces are not limited to a pair of elements, but may simultaneously occur between a single element and many others. Orientation of individual elements also plays no role, as DPD elements are radially symmetrical.

All the elements move in a two- or three-dimensional continuous space, according to the influences of four pairwise forces: the conservative "weak" forces between pairs of particles, a dissipative force between nearby particles, a spring-like "strong" bond force if bonded and a random force:

$$f_i = \sum_j \left(F_{ij}^{\mathrm{C}} + F_{ij}^{\mathrm{D}} + F_{ij}^{\mathrm{B}} + F_{ij}^{\mathrm{R}}\right).$$

All of these forces are considered to operate only within a certain local cutoff radius, r_0. The cutoff radius is a main mechanism for improving model feasibility. Our simulation follows the standard of setting $r_0 = 1$ for convenience, and as such it is omitted from the formulas below. The conservative forces between particles i and j are given by a linear approximation of the Lennard-Jones potential following Besold *et al.* [6]:

$$F_{ij}^{\mathrm{C}} = \alpha_{IJ}(1 - \beta_{IJ} r_{ij})$$

where α_{IJ} and β_{IJ} are specific to the types of i and j and r_{ij} is the Euclidean distance between the particles. The dissipative force, F_{ij}^{D}, causes the kinetic energy of elements to move toward equilibrium with other nearby elements:

$$F_{ij}^{\mathrm{D}} = (v_i - v_j)(1 - r_{ij})^2 \gamma$$

where v_i is the velocity vector of i, γ is a weighting factor given by $\gamma = \sigma^2/2$, and σ is a balancing factor between dissipative and random forces which serves to maintain the temperature of the system around a more or less fixed point. The bonds are represented as Hookean springs:

$$F_{ij}^{\mathrm{B}} = k(r_{ij} - l)$$

where k is the bond strength and l the relaxed bond length. The random force is given by

$$F_{ij}^{\mathrm{R}} = \sigma w^{\mathrm{R}}(1 - r_{ij})u$$

where w^{R} is an independent random factor and u is a uniform random number chosen from the interval $(-1, 1)$. For all our experiments, $\sigma = 3$, $w^{\mathrm{R}} = \sqrt{3}$, with a density of 10 DPD particles per unit volume.

The work reported here uses a DPD implementation of a model of monomers and oligomers in water. Some elements in the model represent bulk water, with one model element representing many water molecules. Other elements could represent hydrophilic or hydrophobic monomers. In some cases those elements are connected by explicit bonds, which are represented as springs that rotate freely about their ends. These complexes explicitly but very abstractly represent the three-dimensional structure of oligomers. For example, amphiphilic molecules can be created by explicitly bonding a hydrophilic monomer "head" onto a hydrophobic "tail" (chain of hydrophobic monomers).

DPD thermodynamic forces can create self-assembled structures held together by the weak associative forces. For example, a DPD system with amphiphiles in water can exhibit a wide variety of known supramolecular amphiphilic phases, including monolayers, bilayers, micelles, rods, vesicles, and bicontinuous cubic structures [26,27,36,48,49]. But since strong bonds never form or break in the traditional DPD framework, that framework is unable to represent self-assembled structures that involve the dynamics of covalent bonds.

To achieve self-assembly processes that involve forming and breaking strong bonds, the DPD framework must be extended by making strong bonds dynamic. To this end, we created dynamic-bonding DPD (or dbDPD), which is a DPD that is augmented with the following two rules:

- Bonds form if $r_{ij} < r^{\mathrm{f}}_{IJ}$, where r^{f}_{IJ} is the bond-forming radius for types I and J.
- Bonds break if either $r_{ij} > r_0$ or $r_{ij} > r^{\mathrm{b}}_{IJ}$, where r^{b}_{IJ} is the bond-breaking radius for types I and J.

The strong bond strength parameter, k, governs the strength of all strong bonds, whether or not they were present in the initial conditions. These bonding rules allow emergent chemical reaction networks, including emergent side-reactions, to interact with self-assembly processes [15].

Note that the temperature of the system changes when bonds form and break. However, the momentum in the system is constant, since the changes in the momentum of individual elements due to bonding events are always symmetrical with respect to the bonded particles.

3. Genetic Algorithm for Chemical Structures

We now describe an evolutionary algorithm (EA) for designing chemical systems. We use the EA to optimize the parameters of dbDPD models, where the parameters are given a wide range of possible values. Specifically, a genome, $\mathbf{g}$, is a vector of chemical systems parameters: $\mathbf{g} = (g_1, \ldots, g_N)$. The evolutionary algorithm we use is simple, but differs from a standard genetic algorithm in important respects. In deference to the expense of testing fitness, all previously tested genomes are candidate parents, rather than only those of last generation, and children are not allowed to repeat previously tested genomes. In an attempt to better mimic real evolutionary processes, mutation, the sole genetic operator, is limited to a subset of the space near the parent's value, rather than the whole range of the gene. Obvious possibilities for future work include extending our results with various more complicated evolutionary algorithms [14,17,20], such as those using crossover, and comparison with a standard genetic algorithm.

The EA proceeds with discrete generations of size P_{g}. Parents are selected from the entire population of genomes tested in any previous generation, not just those tested in the immediately preceding generation. Selection is done via truncation: at the end of each generation the least fit members of the population are culled until the group of next parents reaches a pre-determined size, P_{t}. For all experiments discussed below, $P_{\mathrm{t}} = P_{\mathrm{g}} = 10$.

Each parent produces one child genome $\mathbf{g}_{\mathrm{c}}$ by mutation, with the limitation that $\mathbf{g}_{\mathrm{c}}$ not duplicate a previous $\mathbf{g}$ tested by the EA. The mutation rate per locus is governed by a global parameter, μ (set to 0.5 for all experiments reported here). The mutation operator is given by:

$$g_i^{\mathrm{c}} = u\left(2g_i^{\mathrm{p}} - \frac{g_i^{\mathrm{p}}}{2}\right) + \frac{g_i^{\mathrm{p}}}{2}$$

where u is a uniform random number chosen from the interval (0, 1), and g_i^{c} and g_i^{p} are the ith child and parent gene. Mutation thus has an asymmetric Hamming distance around $\mathbf{g}_{\mathrm{p}}$ such that $g_i^{\mathrm{p}}/2 < g_i^{\mathrm{c}} < 2g_i^{\mathrm{p}}$. The mutation rate is relatively high relative to balance other differences between ours and a standard EA. Because mutations are limited to the range defined by the Hamming distance, even high mutation rates do not lead to mutant children being overly distant in genomic space from their parents. Additionally, because children may not duplicate earlier genomes, much of the genome may be effectively immune to mutation due to previous sampling, particularly in combination with the limit imposed by the Hamming distance.

4. Results

We optimized dbDPD parameters for a handful of different kinds of self-assembled structures. Some involve only weak associative forces, while others also involve strong bonds. This section reports the typical and notable results from a large sample of optimizations of these structures by evolutionary algorithms.

4.1. *Optimization of Micelle Size*

One set of experiments focussed on systems consisting of amphiphile aggregations in water. Amphiphiles in aqueous solution spontaneously assemble into a number of structures, depending on the type of amphiphile and the pH and temperature of the solution. Amphiphilic self-assembled structures typically arise solely because of weak associative forces.

The smallest amphiphile self-assembled structures are micelles. In micelles, amphiphiles aggregate with their hydrophobic tails at the center and the hydrophilic heads around the outside (see Fig. 4.1, right). The size of a micelle is affected by a number of factors, but can be considered in the abstract to depend

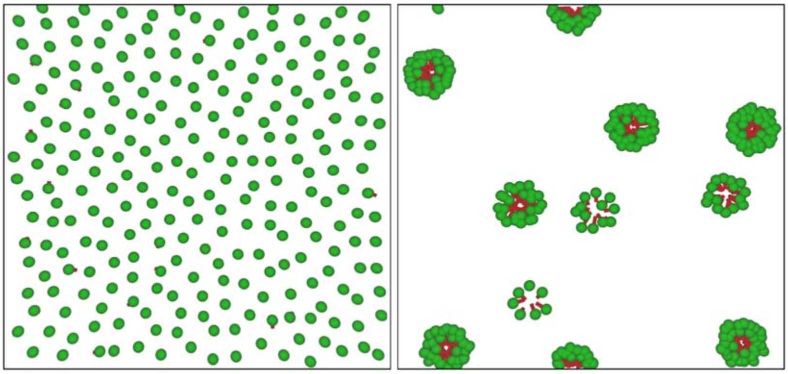

Fig. 4.1. The equilibrium state of the dbDPD system before (left) and after (right) evolutionary design of dbDPD parameters for the micelle size. Amphiphile heads are shown as light-colored spheres, tails as darker columns; water is present but not shown. The system on the left includes amphiphile tails, but most are too short to be seen. In the system on the right, note the self-assembly of ten different micelles of a certain characteristic size. Two micelles wrap around the toroidal edge of the space. For color, see Color Plate Section.

primarily on the length of tail and the "width," in terms of inter-molecular forces, of the head. If the tail is short and the head wide, small micelles composed of low numbers of amphiphiles will be formed.

The EA tested dbDPD parameters in systems consisting of 1000 particles: half water (W) and half tail–head (T–H) dimers. The initial system parameters were such that T–H pairs were strongly bonded and attracted each other, T–H bonds could break but could not reform, Hs repelled each other, and Ts repelled each other. The genome in this EA consists of genes for the bond strength, k, and for the repulsion forces, α_{IJ}:

$$\mathbf{g} = (k, \alpha_{\mathrm{WT}}, \alpha_{\mathrm{WH}}, \alpha_{\mathrm{HH}}, \alpha_{\mathrm{TH}}, \alpha_{\mathrm{TT}})$$

where all are integers in $[1, 400]$.

In this experiment, we use an evolutionary algorithm to tune dbDPD parameters so that micelles spontaneously self-assemble and their size is maximized. The existence and size of micelles is determined by the spatial distribution of amphiphiles. So, the fitness function needs to detect micelles and to measure their size. In the fitness function used here, a micelle-detecting algorithm locates all the amphiphile structures made up of a core of tails surrounded by heads. The algorithm proceeds by (1) listing all the tail particles in the system, (2) choosing a tail particle from the list at random, (3) drawing a circle whose radius is the distance between that tail particle and the nearest head or water particle, (4) removing from the list all the tail particles located inside that circle, and then repeating steps (2)–(4) until the list of tail particles is empty. The algorithm then groups any overlapping circles (where overlap is transitive), and considers each group of overlapping circles as a single micelle. A micelle's radius is considered to be the radius of the largest circle in the group corresponding to that micelle.

The fitness of a given genome (vector of dbDPD parameters) is dependent on the size of the micelles (if any) that form after a set number of model updates when the dbDPD system is started from those dbDPD parameters. More precisely, the fitness, F, of a given genome, $\mathbf{g}$, is the mean size of each micelle it contains,

$$F = \bar{f_i}$$

where f_i is the size of an individual micelle. An individual micelle's size is given by

$$f_i = \left(\frac{s_i}{r_i}\right)^{r_i}$$

where s_i is the number of tails in the micelle core and r_i is the micelle's radius. The fitness function thus increases in the size of both core and radius.

The evolutionary algorithm typically had no difficulty creating large micelles. Different runs of the EA would often take different routes to similar final solutions. In some cases, a good solution is found in less than 20 generations; see Fig. 4.2. But other times the process is more gradual, as in Fig. 4.3, which illustrates a series of discrete adaptations. The fixed dbDPD parameters in these two contexts were similar but not exactly the same. In both cases, $\alpha_{\mathrm{WW}} = 1$, $\beta_{\mathrm{TT}} = 3$ and $\beta = 1$ for all other particle pairs, and for all particle pairs $r^{\mathrm{f}} = 0$ (no bonds form). In the EA shown in Fig. 4.2, $k = 250$, $l = 0.05$, and $r^{\mathrm{b}} = 0.7$. In the EA shown in Fig. 4.3, $l = 0.01$, $r^{\mathrm{b}} = 0.9$, k was added as a gene and allowed to evolve.

These separate adaptations can be identified in the corresponding evolutionary activity plot. Evolutionary activity statistics are a method for visualizing and quantifying the dynamics of evolutionary adaptations [2,4,5,33]. Here, we focus on the activity of only the parents, for those genotypes will contain the significant adaptations. In the present context, the evolutionary activity $A(a, t)$ at time t of a given allele a of a given parental gene g_i is the sum of a's concentration in past parental generations:

$$A(a, t) = \sum_{0}^{t} c(a, t)$$

where $c(a, t)$ is the concentration of the a allele in the parental population at generation t if the allele is present, and 0 otherwise. A discrete number of different allele types are formed from continuous parameters by dividing the range of legal allele values into some finite number of bins (40 bins, in the present study).

We can see the effects of two main adaptations in the evolutionary activity waves in Fig. 4.4. The first adaptation around generation 20 is lower amphiphile

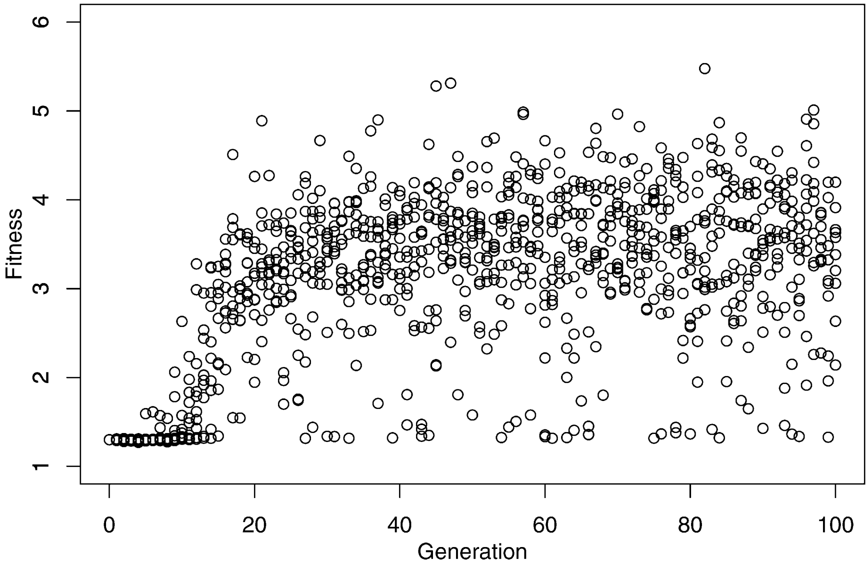

Fig. 4.2. Time series of fitness scatter plot of a micelle size EA, in which a good solution is found in less than 20 generations. Each point represents the fitness of a genome in the EA with all the genomes in a single generation at the same point on the x-axis.

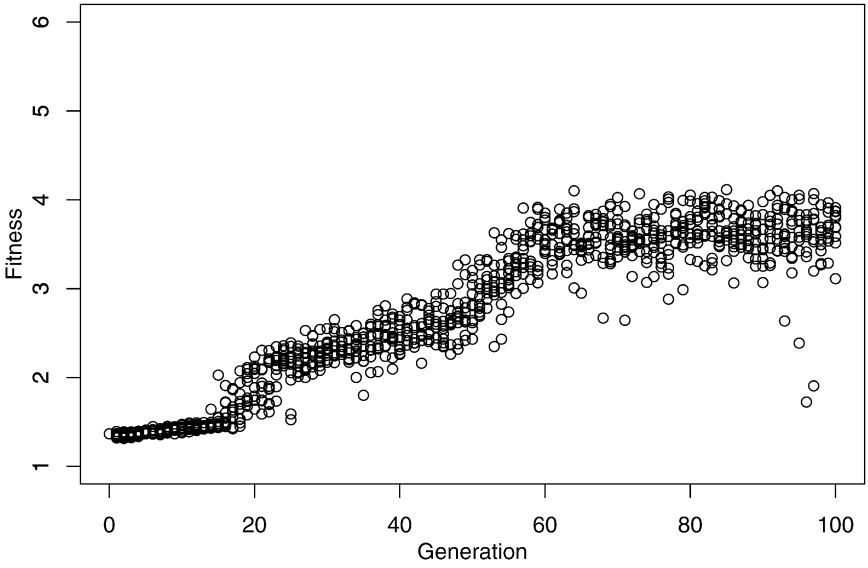

Fig. 4.3. Time series of fitness scatter plot for a micelle size EA, illustrating successive adaptations.

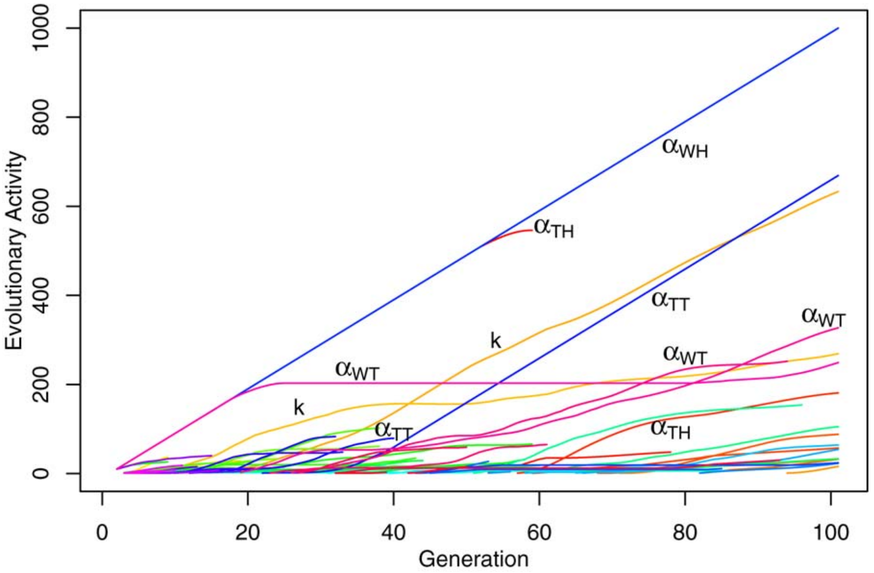

Fig. 4.4. Evolutionary activity waves of all parental loci in the EA shown in Fig. 4.3. Note the rise of a number of adaptations such as the dominant gene in k, α_{TT} and α_{WT}. Also notable is the quiescence of one α_{WT} gene only to see a revival towards the end of the study. A gene at locus α_{TH} can be seen dying out around generation 60 to be replaced by a number of dominant neutral variants. For color, see Color Plate Section.

tail repulsion. This allows amphiphiles to become packed into micelles. One can see the creation of new types of α_{TT} genes, one of which quickly dominates the population (the big α_{TT} wave in the middle of the diagram). This first adaptation also involves increasing the repulsion between heads, but the activity waves for these innovations appear among those in the lower half of the wave diagram. The second main adaptation around generation 50 is increased repulsion between heads and tails. This increases the amphiphile's length, and so increases the size of the micelles. One can see the creation of new activity waves for α_{TH} genes at around this time. (The persisting allele waves involving α_{WH} and k illustrate that these genes are not undergoing significant innovations.)

If we compare the micelles produced by the parents at successive stages in the evolutionary algorithm in Fig. 4.2, we observe a clear directional change. Figure 4.5 compares the distribution of mean core size and mean radius of the micelles made by the parents in generation 10, 15, and 30. One can see that the values of both variables increase over the course of the EA.

Figure 4.6 compares the most fit genomes in the EAs shown in Figs. 4.2 and 4.3. It is evident that both EAs produce roughly similar genomes, though they differ in some details. The fittest genomes from the EA shown in Fig. 4.2 produce micelles with a more regular shape than those from the EA shown in Fig. 4.3.

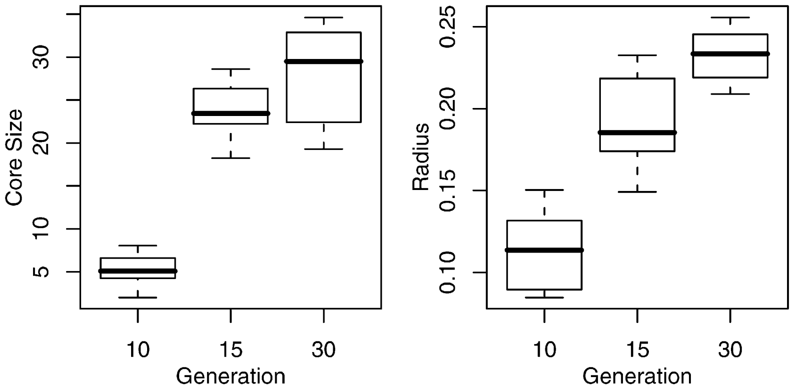

Fig. 4.5. Box plots of the mean values of core size (left) and radius (right) of micelles formed by parents of generation 10, 15, and 30, in the EA shown in Fig. 4.2. A clear directional change is evident. The bottom of the box marks the first quartile ($x_{0.25}$), the top, the third ($x_{0.75}$) and the thicker line, the median. The height of the box defines the interquartile range (*IQR*). A data point is considered an outlier, and shown as an empty dot, if it is smaller than the lower fence ($LF = x_{0.25} - 1.5IQR$) or bigger than the upper fence ($UF = x_{0.75} + 1.5IQR$). The bottom line marks the smallest data point bigger than *LF*, and the top line the biggest data point smaller than *UF*.

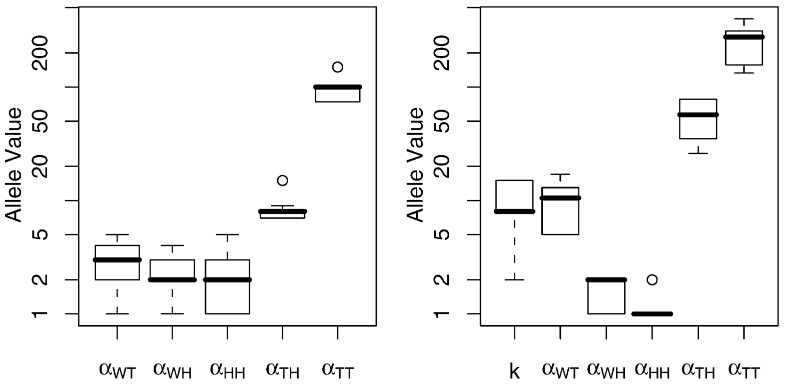

Fig. 4.6. Logarithmically scaled distribution of the most fit genomes in the EAs shown in Figs. 4.2 and 4.3. The bond strength, k, was a gene in the EA shown on the right, but not in the EA shown on the left. Note that the two sets of evolved genomes are similar, though not the same.

We also investigated the structure of the fitness landscape in the micelle size task by making a number of “fitness slices.” In these fitness slices, all parameters were kept constant except for one parameter which was smoothly varied between its minimum and maximum values, and we measured fitness of a number (typically 20) of replicate dbDPD systems. The fitness slices (Fig. 4.7) reveal that the fitness landscape has an interesting, sometimes multi-modal, structure in at least

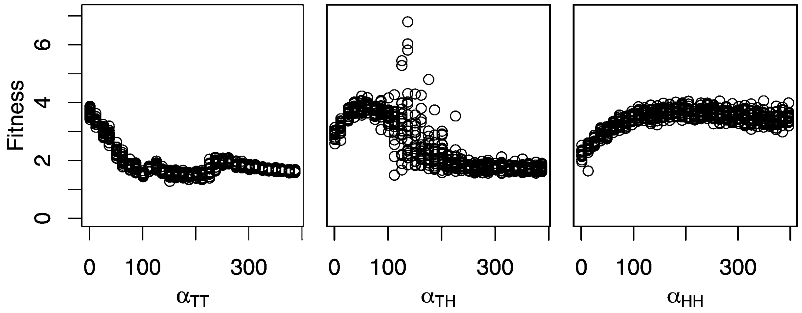

Fig. 4.7. Three slices of the fitness landscape for the micelle size EA. Each slice is a series of scatter plots of twenty fitness measurements in dbDPD parameter space, showing how fitness varies as a single parameter is varied over its whole range of possible values, while all other parameters are held constant. The parameters varied are α_{TT} (left), α_{TH} (middle), and α_{HH} (right). Some significant topographical variation is evident in each slice.

three of its dimensions (α_{TT}, α_{TH} and α_{HH}). The presence of outliers which we observe on the fitness slices for α_{TH} is due to a chaotic point in the parameter space which yielded large but extremely unstable micelles that produced widely differing fitness measurements depending on the momentary state of the system in which the measurement was made.

These fitness slices corroborate our explanation above of the adaptations seen in the activity wave diagram (Fig. 4.4). The first significant adaptation involves lowering the TT repulsion and increasing the HH repulsion, and Fig. 4.7 reveals that for at least one slice through parameter space fitness increases as TT repulsion is lowered (left) and increases as HH repulsion increases (right). The second significant adaptation involves increasing the TH repulsion, and we see that fitness increases as TH repulsion increases from its minimal value (Fig. 4.7, middle).

4.2. Template-Directed Ligation of Uniform Oligomers

Chemical amplification via templating is the basic mechanism of DNA replication, and also of simpler replicator systems such as von Kiedrowski's autocatalytic replicator system [44] and peptide replicators [40]. Monomers of a given type may participate in a weak interaction with monomers of a complementary type, and each may form strong bonds with a monomer of any type if the two are in the correct proximity and orientation. Given a template oligomer made up of different types of monomers and a reservoir of free floating monomers, each monomer of the template oligomer can associate weakly with a complementary free monomer. If the weak forces of the template oligomer bring the free monomers into the correct orientation and proximity, strong bonds form between the monomers producing a complementary oligomer through the process of ligation. In this way, the self-assembly process of template-directed ligation involves both weak and strong chemical bonds.

If the paired complementary oligomers are separated by a mechanism such as duplex melting due to temperature change or protein action, then each oligomer may repeat the process, creating more templates and complements. By this means, the overall number of oligomers in the population increases. Although this process results in the chemical amplification of oligomers, the focus of the present work is simply ligation, and the optimization of parameters that result in the organization and ligation of monomers into oligomers. Unlike a DNA system we attempt to simulate a system in which template oligomers directly catalyze the formation of their complements without any facilitating proteins consistent with RNA world theories of the origin of life.

The ligation systems we subject to evolutionary design are analogous to the chemical system of non-enzymatic template-directed synthesis [1,7,19,24,25,29,50]. Like template-directed systems of ligation *in vivo* and *in vitro*, our system is supplied with a template molecule and an excess of monomers. It then evolves so as to optimize the assembly of monomers on the template to produce a ligated copy of the template.

The simplest form of template-directed ligation involves only *uniform* oligomers, *i.e.*, oligomers composed of a single species of monomer. The template for a uniform oligomer is another uniform oligomer composed of the complementary monomer. Pairs of opposite type units attract each other, while like type units are unlikely to become associated by weak forces. This is roughly analogous to complementary base pairing in the context of nucleotides. In this section we present results of evolutionary design of ligation of uniform oligomers. Results on evolutionary design of the more general form of ligation involving non-uniform oligomers are reported in Section 4.3. It is possible to measure the catalytic efficiency of the templating process by fitting rate constants and comparing with background rates, but this is not done here.

When designing ligation of uniform oligomers, we augmented dbDPD with angular stiffness of covalent bonds, following Shillcock and Lipowsky [36]. The preferred angle was 0, tending towards parallel bonds, and the bending constant was 200. Our genes in this EA are just three dbDPD parameters: the bond strength, the relaxed length of the bond, and the strength of the attractive conservative force between A and B particles:

$$\mathbf{g} = (k, l, \beta_{\mathrm{AB}})$$

where k is an integer in $[1, 1000]$, l is a real number in $[0.01, 0.75]$, and β_{AB} is a real number in $[1, 2]$. The other dbDPD parameters were fixed as follows: $\alpha_{\mathrm{WW}} = \alpha_{\mathrm{WT}} = \alpha_{\mathrm{WH}} = 3$ and all other α values were set to 30, β values for all particle pairs except AB were set to 1, $r^{\mathrm{f}}_{\mathrm{AA}} = r^{\mathrm{f}}_{\mathrm{BB}} = 0.25$ and r^{f} values for all

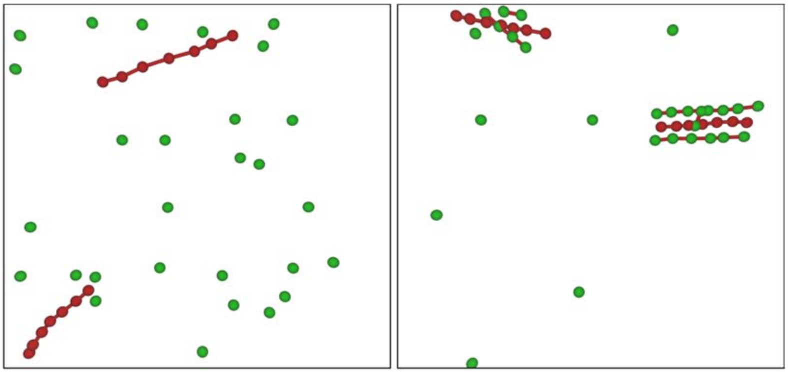

Fig. 4.8. The dbDPD system before (left) and after (right) evolutionary design of uniform oligomer ligation. Template oligomers are dark, monomers and target oligomers lighter. Water is present but not shown. After evolutionary design, it is evident that the template successfully catalyzes the production of complementary oligomers. For color, see Color Plate Section.

other particle pairs were set to 0, and $r^{\mathrm{b}} = -1$ for all particle pairs (bonds do not break).

A genome's fitness is measured by starting the dbDPD with the genome's parameters and seeding the system with one kind of free monomers and a template that is a uniform oligomer formed from the complementary monomer. The fitness of a genome is given by:

$$F = \sum_{l=l_{\min}}^{l_{\max}} n_l \times k^l$$

where l is the length of the oligomer, $l_{\min}$ is the length of the shortest oligomer which counts towards fitness, $l_{\max}$ is the length of the longest oligomers allowed in the system, n_l is the number of oligomers of a given length l, and k is a scaling factor that governs the relative importance of forming longer oligomers. For all our experiments $k = 1.5$. Many generalizations and modifications of our search algorithm and fitness function could be explored.

As can be seen in Fig. 4.8 the EA was successful at templating oligomers in the uniform case. Fitness improved from zero to over eighty percent of the maximum possible. The twenty percent shortfall from perfect performance is from incomplete ligation due in large part to lack of sufficient diffusion of monomers in the time allotted for evaluating the system.

4.3. Template-Directed Ligation of Non-Uniform Trimers

Ligation of uniform oligomers is a limiting case of ligation of non-uniform oligomers. In this section, we report results on evolutionary design of the more general form of ligation. To simplify the chemistry as much as possible, we focus solely on the ligation of trimers composed out of two types of monomers.

Our genes are eight chemical system parameters: the bond strength, the relaxed length of the bond, the strength of the repulsive conservative force between A particles, the strength of the repulsive conservative force between B particles, the strength of the attractive conservative force between A and B particles, the bond-forming radius for two A particles, the bond-forming radius for two B particles, and the bond-forming radius for A and B particles:

$$\mathbf{g} = \left(k, l, \alpha_{\mathrm{AA}}, \alpha_{\mathrm{BB}}, \beta_{\mathrm{AB}}, r^{\mathrm{f}}_{\mathrm{AA}}, r^{\mathrm{f}}_{\mathrm{BB}}, r^{\mathrm{f}}_{\mathrm{AB}}\right)$$

where k, α_{AA} and α_{BB} are integers in [5, 200], β_{AB} is an integer in [5, 100], $r^{\mathrm{f}}_{\mathrm{AA}}$, $r^{\mathrm{f}}_{\mathrm{BB}}$ and $r^{\mathrm{f}}_{\mathrm{AB}}$ are real numbers in [0.05, 0.75] and l is a real number in [0.01, 0.75]. The other dbDPD parameters were fixed as follows: α and β values for all particle pairs not controlled by genes were set to 1, and r^{b} for all particle pairs were set to 0.8.

Here, a genome's fitness has three components, corresponding to the ability to ligate three different classes of trimers. Each fitness component is measured by starting the dbDPD with the genome's parameters and seeding the system solely with free monomers and one of three kinds of template trimers—AAA, AAB, and ABA—and letting dbDPD run for a fixed number of model updates. These templates and their complements cover all possible trimers that can be formed from the monomers A and B. The components of a genome's fitness are the number of correct template and complementary trimers formed when seeded with each kind of trimer template. Formally, a genome's fitness is:

$$F = n_{\mathrm{AAA}} \times n_{\mathrm{AAB}} \times n_{\mathrm{ABA}}$$

where n_X is the number of X trimers and their complements produced after seeding the system with templates of type X.

Figure 4.9 shows a scatterplot of the fitness of each genome over a typical EA run. In addition to the fitness of the genomes in each generation, the fitness of the parents of each generation is also plotted. This makes more apparent the mechanism of the EA as well as allowing more detailed investigation of performance in terms of the component fitnesses. We observed a steady increase in fitness over the course of one hundred generations.

Figure 4.10 shows the evolutionary activity of the parents in the EA shown in Fig. 4.9. (Recall definitions from Section 4.1.) We can see significant adaptations in the genes for different types of bonding rules: β_{AB} first adapts, then $r^{\mathrm{f}}_{\mathrm{AA}}$ and $r^{\mathrm{f}}_{\mathrm{BB}}$. (We also see nearly constant domination of the parent population by alleles for genes k and l, presumably because they are not undergoing significant change.)

Examining the changes in each component of the fitness function underlines the implications of the activity wave data. Figure 4.11 shows a scatterplot of

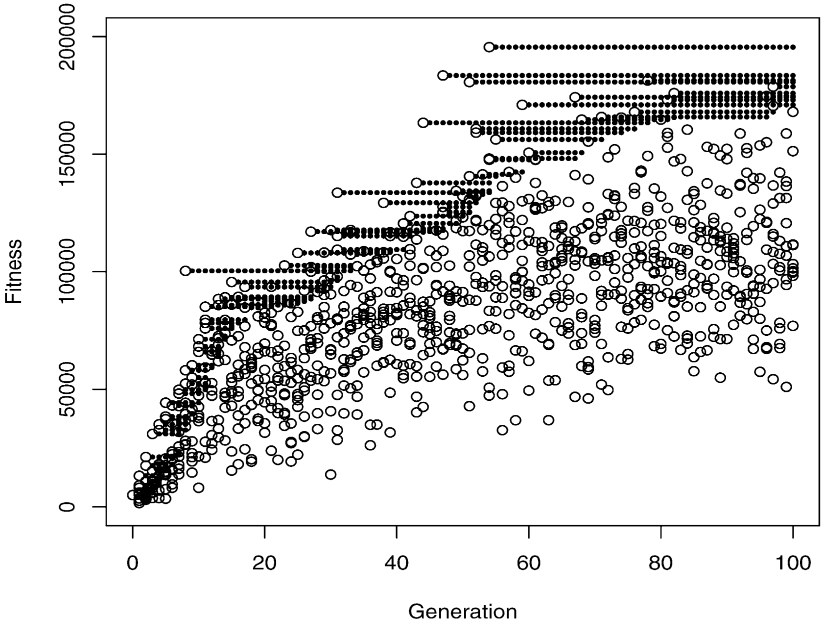

Fig. 4.9. Measured fitness of each parent system (small closed circles) used by a typical EA designing template-directed ligation systems, and of child systems (open circles) produced by the EA.

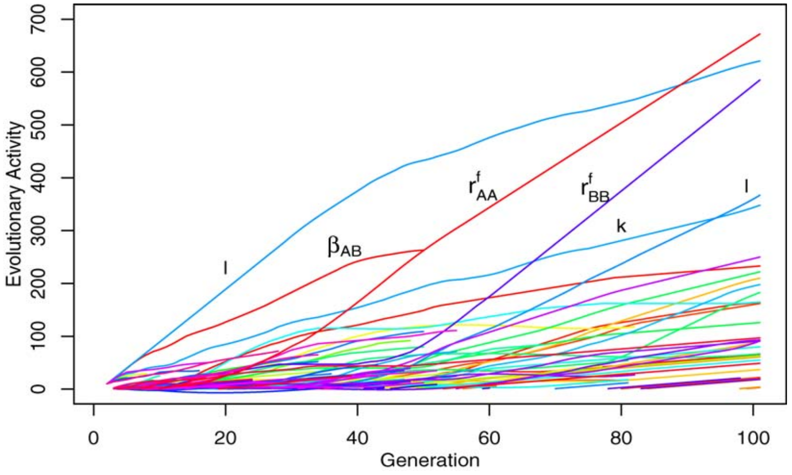

Fig. 4.10. Evolutionary activity waves of types of alleles at all parental loci in the EA shown in Fig. 4.9. One β_{AB} activity wave ends around generation 50 when β_{AB} waves for alleles with higher values originate; these new waves are mixed with others in the bottom of the diagram. For color, see Color Plate Section.

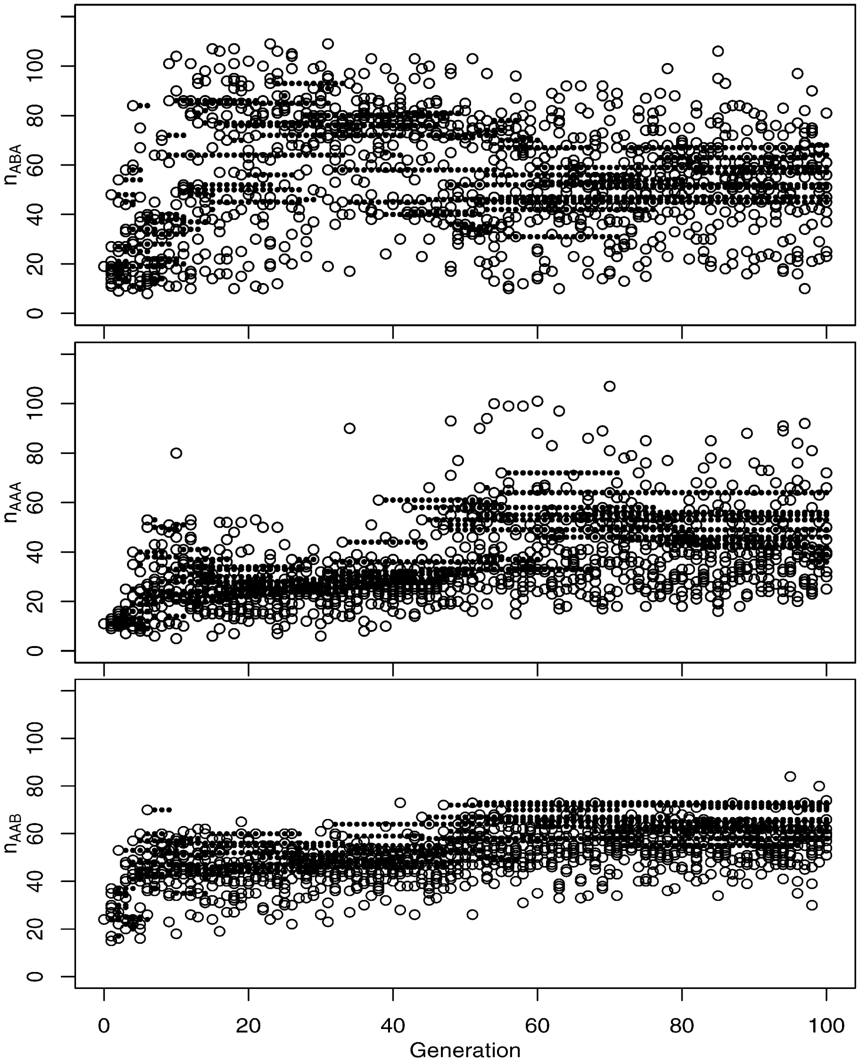

Fig. 4.11. Time series of scatter plots of the three components of fitness—n_{ABA} above, n_{AAA} in the middle, and n_{AAB} below—in conditions started with the three corresponding kinds of templates in the EA shown in Figs. 4.9 and 4.10. As in Fig. 4.9, the component fitness of the parents of each generation are shown with small closed circles and those of the children with open circles. Note the initial rise in n_{ABA}, followed by rises in n_{AAA} and finally n_{AAB}.

each component of fitness of the genomes in the EA in Fig. 4.9. The EA first increases n_{ABA}; this is the simplest fitness component to optimize because it involves only the AB bond. This is followed by increasing n_{AAA}; this is somewhat more complex because it requires forming both AA and BB bonds for the template and target respectively, but always in isolation from each other. The fitness component that increases the slowest is n_{AAB}. This involves correctly forming each possible type of bond. Forming both like-type and complement-type bonds within the same oligomer proves very difficult in dbDPD. Note also that the parents shift somewhat in their rank in n_{AAA} and n_{ABA}, but are always near the top of n_{AAB}—more evidence of the latter's key importance.

5. Discussion

The behavior of probabilistic models like dbDPD is stochastic. This makes measurement of their expected behavior noisy, presenting an extra hurdle for our EA. One simple solution is to do enough replicates to average away the noise [3], but this is expensive in terms of fitness measurements. EAs can compensate for this by rewarding high measured fitness and punishing low measured fitness more gently. But this lessens the EA's ability to quickly abandon dead ends and capitalize on new breakthroughs.

The stochasticity of the dbDPD simulations also explains in part why different EAs sometimes find different solutions to the same optimization problem, the remaining variation due to stochasticity within the EA. We ran many EAs identical in all but the initial seed for the random number generator and saw significant differences in the time it takes to reach optimal solutions, as well the variance in the fitness of the child genomes produced. The differences are analogous to the differences between the EAs shown in Figs. 4.2 and 4.3, though those EAs dealt with slightly different problems. The first finds an optimal solution faster than the second, but the greater variance of children fitness reveals that the solution is less robust.

The heart of a EA is its fitness function. Our non-uniform ligation fitness function multiplies three components. The fitness component graphs (Fig. 4.11) reveal a significant gap between desired and observed dbDPD behavior. The EA achieves a high values for one fitness component (ligating ABA trimers, n_{ABA}), but the resulting systems are poor at ligating other kinds of trimers. A better fitness function might improve results. For example, our fitness functions for ligation merely counts the oligomers of the right types, but it ignored the actual process that produced the oligomers. One could use a fitness function that explicitly assays to what extent the process of template-directed ligation is actually happening. This would appropriately devalue systems that produce the right oligomers by some other process than ligation. This fitness function should produce general, non-uniform template-directed ligation, providing it is possible to realize this process in the dbDPD framework.

But evolutionary design methods can succeed only if an adequate solution exists somewhere in the search space. We concluded from the present study that the dbDPD framework can produce only limited forms of templated ligation. Certain simple oligomers, such as the ABA case, were readily produced, but the mechanism for this was not solely template-directed ligation. Furthermore, the yield of ligation with more complex non-uniform oligomers, such as AAB, was poor.

This limitation appears to stem from the radial symmetry and specific bond rules in the dbDPD framework. In real templated ligation, a free monomer is held in a specific spatial orientation by weak bonds with another monomer in the template. If this happens at two adjacent locations in the template, then the template can hold two free monomers in the correct spatial orientation long enough that a strong bond forms between them. Without the template, free monomers are unlikely to bond because of the low probability of them coming into contact in the correct spatial orientation and remaining there long enough for a bond to form. In dbDPD, on the other hand, weak associative forces attract any free monomers within r_0 so a *cluster* of free monomers can be attracted to a given location in the template. But strong bonds may form between any two monomers which are within r^{f} for their respective types. So in dbDPD, while strong bonds will form between two free monomers attracted to *adjacent* locations in the template (as in real templated ligation), strong bonds are just as likely to form between two free monomers attracted to the *same* location in the template. Thus, the sequence of strong bonds that form is not controlled specifically by the sequence of monomers in the template.

For these reasons, it appears that the present dbDPD framework is insufficient to achieve templated ligation of non-uniform oligomers. Appropriate changes to dbDPD may resolve this problem, but the consequent increase in computational costs is presently unknown.

6. Conclusion

We here achieve evolutionary design of a model chemistry involving self-assembling amphiphiles in water. This demonstrates how evolutionary algorithms and similar indirect design methods can be used to “program” or optimize self-assembling chemical structures. This method works in a number of contexts, including those illustrated by micelle size and templated ligation of uniform oligomers.

The majority of dbDPD simulation parameters optimized here can be effectively varied in actual chemical systems. However, in some cases they cannot be varied independently—*i.e.*, there may not exist an amphiphile which has a particular repulsion with other head groups as well as a particular length tail—but otherwise they are analogous to real chemical variables. Thus, it is possible that optimal dbDPD systems correspond to real chemical systems with the same desired characteristics.

The evolutionary algorithm used here was intentionally kept very simple. EA performance can typically be boosted by a variety of methods [16]. So we expect that the performance shown here can be significantly improved.

The evolutionary design method shown here can be used to optimize other chemical functionalities in other kinds of chemical systems, both real and simulations. It has already been applied with high-throughput screening to optimize real amphiphile systems in the wet lab [39]. The resulting marriage of machine learning with chemical screening yields an automated, intelligent screening method for finding solutions in very sparse samples of huge and high-dimensional search spaces.

Acknowledgements

Thanks to the European Center for Living Technology (ECLT) for hospitality during some of the work reported here. Thanks to Rüdi Fuchslin and Anders Eriksson for helpful discussion of DPD, and to Thomas Maeke and John McCaskill for the DPD code from which the present model was developed. Thanks for helpful discussion to Norman Packard, Martin Hanczyc, and Morgan Theis. Thanks also to audiences at Dublin City University, the University of Nottingham, and EA'05 where some of this work was reported. This work was funded in part by PACE (Programmable Artificial Cell Evolution), a European Integrated Project in the EU FP6 IST FET Complex Systems Initiative.

References

[1] O.L. Acevedo, L.E. Orgel, Nonenzymatic transcription of an oligodeoxynucleotide 14 residues long, *Journal of Molecular Biology* **197** (1987) 187–193.

[2] M.A. Bedau, C.T. Brown, Visualizing evolutionary activity of genotypes, *Artificial Life* **5** (1999) 17–35.

[3] M.A. Bedau, A.J. Buchanan, G. Gazzola, M. Hanczyc, T. Maeke, J.S. McCaskill, I. Poli, N.H. Packard, Evolutionary design of a DDPD model of ligation, in: *Lecture Notes in Computer Science*, vol. 3871, 2005, pp. 201–212.

[4] M.A. Bedau, N.H. Packard, Measurement of evolutionary activity, teleology, and life, in: C. Langton, C. Taylor, D. Farmer, S. Rasmussen (Eds.), *Artificial Life II*, Addison–Wesley, Redwood City, 1991, pp. 431–461.

[5] M.A. Bedau, E. Snyder, N.H. Packard, A classification of long-term evolutionary dynamics, in: C. Adami, R. Belew, H. Kitano, C. Taylor (Eds.), *Artificial Life VI*, MIT Press, Cambridge, 1998, pp. 228–237.

[6] G. Besold, I. Vattulainen, M. Karttunen, J.M. Polson, Towards better integrators for dissipative particle dynamics simulations, *Phys. Rev. E Rapid Comm.* **62** (2000) 7611–7614.

[7] C. Bohler, P.E. Nielsen, L.E. Orgel, Template switching between PNA and RNA oligonucleotides, *Nature* **376** (1995) 578–581.

[8] S. Brenner, R.A. Lerner, Encoded combinatorial chemistry, *Proceedings of the National Academy of Science USA* **89** (1992) 5381–5383.

[9] K.B. Chapman, J.W. Szostak, In vitro selection of catalytic RNAs, *Current Opinions in Structural Biology* **4** (1994) 618–622.

[10] D. Cliff, I. Harvey, P. Husbands, Explorations in evolutionary robotics, *Adaptive Behavior* **2** (1993) 71–104.

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

[12] J. Ellman, B. Stoddard, J. Wells, Combinatorial thinking in chemistry and biology, *Proceedings of the National Academy of Science USA* **94** (1997) 2779–2782.

[13] S. Nolfi, D. Floreano, *Evolutionary Robotics: The Biology, Intelligence, and Technology of Self-Organizing Machines*, MIT Press, Cambridge, 2004.

[14] S. Forrest, Genetic algorithms: Principles of natural selection applied to computation, *Science* **261** (1993) 872–878.

[15] G. Gazzola, A.J. Buchanan, N.H. Packard, M.A. Bedau, Catalysis by self-assembled structures in emergent reaction networks (2006), submitted for publication.

[16] D.E. Goldberg, *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*, Kluwer Academic Publishers, Boston, MA, 2002.

[17] D.E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison–Wesley, Reading, MA, 1989.

[18] R. Groot, P. Warren, Dissipative particle dynamics: Bridging the gap between atomistic and mesoscopic simulations, *Journal of Chemical Physics* **107** (1997) 4423–4435.

[19] A.R. Hill Jr, L.E. Orgel, T. Wu, The limits of template-directed synthesis with nucleoside-5'-phosphoro(2-methyl)imidazolides, *Origins of Life and Evolution of the Biosphere* **23** (1993) 285–290.

[20] J.H. Holland, *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, 1975. Reprinted by MIT Press, Cambridge, MA, 1992.

[21] P. Hoogerbrugge, J. Koelman, Simulating microscopic hydrodynamic phenomena with dissipative particle dynamics, *Europhysics Letters* **19** (1992) 155–160.

[22] D. Irvine, C. Tuerk, L. Gold, SELEXION. Systematic evolution of ligands by exponential enrichment with integrated optimization by non-linear analysis, *Journal of Molecular Biology* **222** (1991) 739–761.

[23] G. Joyce, Directed evolution of nucleic acid enzymes, *Annual Review of Biochemistry* **73** (2004) 791–836.

[24] G.F. Joyce, T. Inoue, L.E. Orgel, Non-enzymatic template-directed synthesis on RNA random copolymers. Poly(C, U) templates, *Journal of Molecular Biology* **176** (1984) 279–306.

[25] G.F. Joyce, L.E. Orgel, Non-enzymatic template-directed synthesis on RNA random copolymers. Poly(C, A) templates, *Journal of Molecular Biology* **202** (1988) 677–681.

[26] S. Jury, P. Bladon, M. Cates, S. Krishna, M. Hagen, N. Ruddock, P. Warren, Simulation of amphiphilic mesophases using dissipative particle dynamics, *Physical Chemistry and Chemical Physics* **1** (1999) 2051–2056.

[27] M. Kranenburg, M. Venturoli, B. Smit, Phase behavior and induced interdigitation in bilayers studied with dissipative particle dynamics, *Journal of Physical Chemistry* **107** (2003) 11491–11501.

[28] H. Lipson, J.B. Pollack, Automatic design and manufacture of robotic lifeforms, *Nature* **406** (2000) 974–978.

[29] R. Liu, L.E. Orgel, Enzymatic synthesis of polymers containing nicotinamide mononucleotide, *Nucleic Acids Research* **23** (1995) 3742–3749.

[30] C. Marsh, Theoretical aspects of dissipative particle dynamics, PhD Thesis, University of Oxford, 1998.

[31] J.B. Pollack, H. Lipson, G. Hornby, P. Funes, Three generations of automatically designed robots, *Artificial Life* **7** (2001) 215–223.

[32] S. Rasmussen, L. Chen, D. Deamer, D. Krakauer, N. Packard, P. Stadler, M. Bedau, Transitions from nonliving to living matter, *Science* **303** (2004) 963–965.

[33] M.J. Raven, M.A. Bedau, General framework for evolutionary activity, in: *Lecture Notes in Artificial Intelligence*, vol. 2801, 2003, pp. 676–685.

[34] E. Reddington, A. Sapienza, B. Gurau, R. Viswanathan, S. Sarangapani, E.S. Smotkin, T.E. Mallouk, Combinatorial electrochemistry: A highly parallel, optical screening method for discovery of better electrocatalysts, *Science* **280** (1998) 1735–1737.

[35] R. Rohatgi, D.P. Bartel, J.W. Szostak, Nonenzymatic, template-directed ligation of oligoribonucleotides is highly regioselective for the formation of 3'-5' phosphodiester bonds, *Journal of the American Chemical Society* **118** (1996) 3340–3344.

[36] J. Shillcock, R. Lipowsky, Equilibrium structure and lateral stress distribution from dissipative particle dynamics simulations, *Journal of Chemical Physics* **117** (2002) 5048–5061.

[37] J. Singh, M.A. Ator, E.P. Jaeger, M.P. Allen, D.A. Whipple, J.E. Soloweij, S. Chowdhary, A.M. Treasurywala, Application of genetic algorithms to combinatorial synthesis: A computational approach to lead compound identification and lead optimization, *Journal of the American Chemical Society* **118** (1996) 1669–1676.

[38] J.W. Szostak, D.P. Bartel, P.L. Luisi, Synthesizing life, *Nature* **409** (2001) 387–390.

[39] M. Theis, G. Gazzola, M. Forlin, I. Poli, M.M. Hanczyc, M.A. Bedau, Optimal formulation of complex chemical systems with a genetic algorithm (2006), submitted for publication.

[40] T. Tjivikua, P. Ballester, J. Rebek Jr, A self-replicating system, *Journal of the American Chemical Society* **112** (1990) 1249–1250.

[41] S. Trofimov, E. Nies, M. Michels, Thermodynamic consistency in dissipative particle dynamics simulations of strongly nonideal liquids and liquid mixtures, *Journal of Chemical Physics* **117** (2002) 9383–9394.

[42] C. Tuerk, L. Gold, Systematic evolution of ligands by exponential enrichment: RNA ligands to bacteriophage T4 DNA polymerase, *Science* **249** (1990) 505–510.

[43] I. Vattulainen, M. Karttunen, G. Besold, J. Polson, Integration schemes for dissipative particle dynamics simulations: From softly interacting systems towards hybrid models, *Journal of Chemical Physics* **116** (2002) 3967–3979.

[44] G. von Kiedrowski, A self-replicating hexadeoxynucleotide, *Angewandte Chemie International Edition in English* **25** (1986) 932–935.

[45] R.A. Watson, S.G. Ficici, J.B. Pollack, Embodied evolution: Distributing an evolutionary algorithm in a population of robots, *Robotics and Autonomous Systems* **39** (2002) 1–18.

[46] L. Weber, S. Wallbaum, C. Broger, K. Gubernator, Optimization of the biological activity of combinatorial compound libraries by a genetic algorithm, *Angewandte Chemie International Edition in English* **34** (20) (1995) 2280–2282.

[47] M. Wright, G. Joyce, Continuous in vitro evolution of catalytic function, *Science* **276** (1997) 614–617.

[48] S. Yamamoto, S. Hyodo, Budding and fission dynamics of two-component vesicles, *Journal of Chemical Physics* **118** (2003) 7937–7943.

[49] S. Yamamoto, Y. Maruyama, S. Hyodo, Dissipative particle dynamics study of spontaneous vesicle formation of amphiphilic molecules, *Journal of Chemical Physics* **116** (2002) 5842–5849.

[50] W.S. Zielinski, L.E. Orgel, Oligoaminonucleoside phosphoramidates. Oligomerization of dimers of 3'-amino-3'-deoxy-nucleotides (GC and CG) in aqueous solution, *Nucleic Acids Research* **15** (1987) 1699–1715.