Journal of Computer-Aided Molecular Design, **16:** 551–567, 2002.
KLUWER/ESCOM

Genetic algorithm for the design of molecules with desired properties

Stefan Kamphausen[2,§], Nils Höltge[2,§], Frank Wirsching[1,§], Corinna Morys-Wortmann[1,2,§], Daniel Riester[1,2], Ruediger Goetz[2], Marcel Thürk[2] & Andreas Schwienhorst[1,*]
[1]*Abteilung fuer Molekulare Genetik und Praeparative Molekularbiologie, Institut für Mikrobiologie und Genetik, Grisebachstr. 8, 37077 Goettingen, Germany;* [2]*Novel Science International GmbH, Obere Karspuele 36, 37073 Goettingen, Germany*

Received 15 July 2002; Accepted 9 October 2002

Key words: Genetic algorithm, thrombin inhibitor screening, computer-assisted drug discovery, peptide library, *de novo* design

Summary

The design of molecules with desired properties is still a challenge because of the largely unpredictable end results. Computational methods can be used to assist and speed up this process. In particular, genetic algorithms have proved to be powerful tools with a wide range of applications, e.g. in the field of drug development. Here, we propose a new genetic algorithm that has been tailored to meet the demands of *de novo* drug design, i.e. efficient optimization based on small training sets that are analyzed in only a small number of design cycles. The efficiency of the design algorithm was demonstrated in the context of several different applications. First, RNA molecules were optimized with respect to folding energy. Second, a spinglass was optimized as a model system for the optimization of multiletter alphabet biopolymers such as peptides. Finally, the feasibility of the computer-assisted molecular design approach was demonstrated for the *de novo* construction of peptidic thrombin inhibitors using an iterative process of 4 design cycles of computer-guided optimization. Synthesis and experimental fitness determination of only 600 different compounds from a virtual library of more than 10^{17} molecules was necessary to achieve this goal.

Introduction

Optimization strategies that mimic Darwinian evolution have long been applied to the automated *de novo* design of molecules, particularly within the field of modern drug discovery. Computer-based evolutionary optimization of molecules with respect to a set of desired properties depends on two integral parts: (i) a suitable representation of the molecules, e.g. a sequence of building blocks and/or a set of descriptors as a measure for 'similarity'; and (ii) a systematic and efficient search strategy that facilitates navigation in the extremely large chemical structure space omitting traps of local optima. Most of the recent advances in the field have been greatly stimulated and enhanced by a range of relatively new algorithms, predominantly imported from the artificial intelligence field. The core of these computing techniques are considered to be artificial genetic algorithms, neural networks, fuzzy logic and some other less well–known techniques (for review see [1]). Genetic algorithms (GA) were developed by Holland in the 1970s [2] and follow most closely the principles of Darwinian evolution. Populations of candidate solutions evolve under a certain selective pressure. The higher the fitness of a member of the population, the more likely is its (error-prone) reproduction. Variation is introduced into the offspring by applying two operators, crossover and mutation. The potential of GAs for solving combinatorial optimization problems has been explored in great detail [3]. In the field of drug design, GAs have been used with considerable success to solve problems like docking of ligands to target pro-

*To whom correspondence should be addressed. E-mail: aschwie1@gwdg.de
§These authors contributed equally to the results presented.

teins, structure alignment, variable selection in QSAR (Quantitative Structure-Activity Relationship) studies and design of combinatorial libraries (for review see [4]). However, many GAs are not particularly well suited to serve in an efficient *de novo* drug discovery process (Figure 1). Many algorithms have only been demonstrated to work with theoretical scoring functions [5]. Since our present state of knowledge concerning structure-function relationships in molecules with complex biological functions does not allow the theoretical prediction of molecular properties alone; experimental data obtained by analysis of candidate molecules should also be used as part of the scoring function. Furthermore, GAs often need large populations of solutions and – even more importantly – a large number of generations (> 50), in order to converge to a satisfactory fitness optimum. In exceptional cases, these problems could be overcome, and only then by combining GA with local optimization methods [6–8]. If computer-assisted molecular design really wants to compete with random search strategies such as high-throughput screening of large libraries, it has to work comparatively much more efficiently, i.e. depend on testing considerably smaller numbers of molecules in less time.

Here, we present a new group of genetic algorithms that was designed to converge rapidly to a high-grade local fitness optimum. The algorithms make use of a new crossover operator (n×m crossover) and a tournament selection procedure [9] to define parent molecules. Herein, computer-assisted optimization has been used to solve three different types of problems: (i) the optimization of RNA sequences with respect to folding energy; (ii) the optimization of a spinglass; and (iii) the optimization of peptidic thrombin inhibitors in an iterative process using only experimental data as fitness parameters since other known applications of computer-aided molecular design also deal with the same topic [10, 11].

Methods

Algorithms

To compare the performance of new algorithms with classical GA approaches, we constructed a standard GA as a reference. Candidate sequences of the same length **L** were used. The basic steps used in this reference GA follow the patterns shown in Figure 2. First, an initial population (size: $\mathbf{S_{pop}}$) of candidate sequences is generated by a random process (1). Then, the fitness of each candidate is evaluated using a 'fitness function', which takes as input a candidate solution and returns a numeric score (2). In the selection step (3), $\mathbf{n_{can}} = 2$ candidates (parents) are selected, each by a tournament procedure with $\mathbf{n_{Tourn}} = 2$ tournament participants drawn randomly from the population. In each tournament selection procedure, the fittest candidate is selected. In the breeding step (4), a crossover is carried out between the two selected parent sequences at a probability of $\mathbf{P_{CO}}$. In the case of the reference algorithm, only a standard 1-point-crossover operator was used. Briefly, both sequences are cut at corresponding sites. The first half of the first sequence is then combined with the second part of the second sequence. The other possible recombination product is discarded. If no crossover was chosen (probability: $\mathbf{1\text{-}P_{CO}}$), then one of the parent sequences was simply copied to become the new individual. In the next breeding step (5), the new individual is mutated with a mutation rate per position of $\mathbf{P_{mut}}$. The new individual is added to a transient pool (6) of new candidates and steps (3) to (6) are iterated until the transient pool reached the population size $\mathbf{S_{pop}}$ (as the initial population). Finally, the transient pool of molecules is added to the initial pool to generate the population for the next cycle (7). The whole procedure is iterated for a given number of cycles.

The first type of the new GA with n×m crossover (n = number of parents, m = number of positions of crossover) was designed to handle candidate sequences of the same length **L**. The basic steps of the algorithm are depicted in Figure 3. First, an initial population (size: $\mathbf{S_{pop}}$) of candidate sequences is generated by a random process (1). Then, the fitness of each candidate is assigned according to a 'fitness function' (2). In the selection step (3), $\mathbf{n_{can}}$ candidates (parents) are selected, each by a tournament procedure with $\mathbf{n_{Tourn}}$ tournament participants drawn randomly from the population. In the breeding step (4), a n×m crossover is carried out between selected parent sequences. In this paper, we only focus on the simplest case where $\mathbf{n_{can}}$ is equal to the number of elements in the sequence (L). First, all $\mathbf{n_{can}}$ candidates (parents) are aligned. The new individual is then generated by combining the first position of the first sequence with the second position of the second sequence and so on. This is equivalent to building a n×m matrix using the sequences as rows and choosing the elements of the main diagonal as the symbols of the new individual. Alternatively, the new individual is generated by

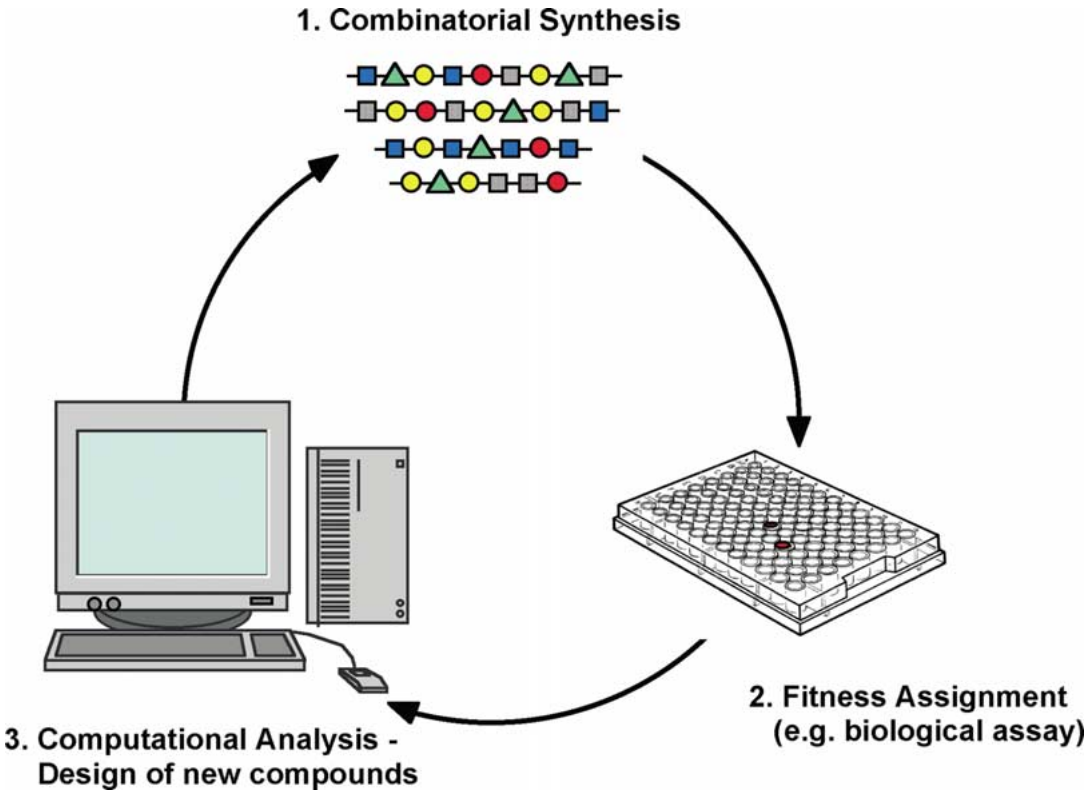


Figure 1. Computer-assisted drug discovery (CADDIS). The optimization process is organized in design cycles. After synthesis, compounds are tested in a suitable bioassay to obtain fitness parameters. Alternatively, fitness parameters are deduced from compound structures. Both algebraic representations of compound structures and fitness parameters are used by an algorithm to calculate a new library of compounds which again are characterized as described. The process is iterated until molecules with desired properties are identified.

combining elements within diagonals at one or two positions at either side of the main diagonal. Throughout this paper we apply GAs with n×m crossover using a normal distribution of possible shifts −2 (2%), −1 (22%), 0 (52%), +1 (22%), +2 (2%) around 0 (main diagonal). In this way, we allow for shifts of subsequences to another position relative to the termini of the sequences, a feature unknown in classical GAs. The new individual is added to a transient pool (5) of new candidates and steps (3) to (5) are iterated until the transient pool reaches the population size $\mathbf{S_{pop}}$ (as the initial population). Finally, the transient pool of molecules is added to the initial pool to generate the population for the next cycle (6). The whole procedure is iterated for a given number of generations.

To overcome the limitation of fixed sequence length, we constructed a second type of GA with n×m crossover. The flowchart is essentially the same as in Figure 3 with the n×m crossover operator changed as follows. First, the upper and lower limits for sequence length are determined. In the case of RNA structure optimization we chose $L_{min} = 20$ and $L_{max} = 32$ and the 4 symbol alphabet G, A, U and C. The next step is to determine the length of the offspring. The length is computed as the average of the lengths of the best individual and the average length of the selected parents:

$$L_{new} = \frac{L_{best} + \frac{1}{L_{max}} \sum_{i=0}^{L_{max}} L_i}{2}$$

where L_{best} denotes the length of the best individual of the current start population and L_i is the length of the i-th parent selected. In the breeding step (4), sequences of all $\mathbf{n_{can}}$ selected candidates (parents) are aligned and a n×m crossover is carried out between parent sequences as described. However, since the matrix of parent elements (between L_{min} and L_{max} per sequence) is now not necessarily filled at all positions, we decided to use repeats of the corresponding sequence

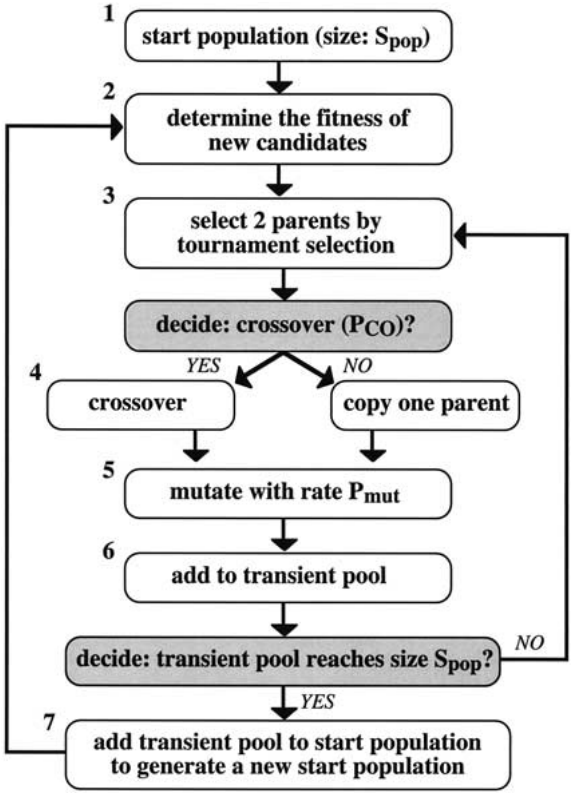


Figure 2. General flowchart for a standard genetic algorithm.

until the maximum length L_{max} is achieved:

ACUCUUCACUGGAGAGUUGG → **ACUCUUC**
ACUGGAGAGUUGGACUCUUCACUGG

The new individual is then generated by combining elements of each parent sequence (rows in the matrix) as described. According to the previously calculated offspring length, we randomly chose which rows to take and which to skip.

Algorithms were encoded in C++, compiled with gcc version 2.95.3 and never used more than 2 MB of memory. They were implemented on a PentiumIII 1 GHz with 1GB RAM.

Fitness functions

The fitness function or scoring function is the part of the (genetic) algorithm where it is adapted to a specific problem. The score could either be derived from empirical evaluation (e.g. conformational energy), assignments according to a theoretical model or experimental biological measurements. The latter, namely the experimental determination of thrombin inhibition is described below.

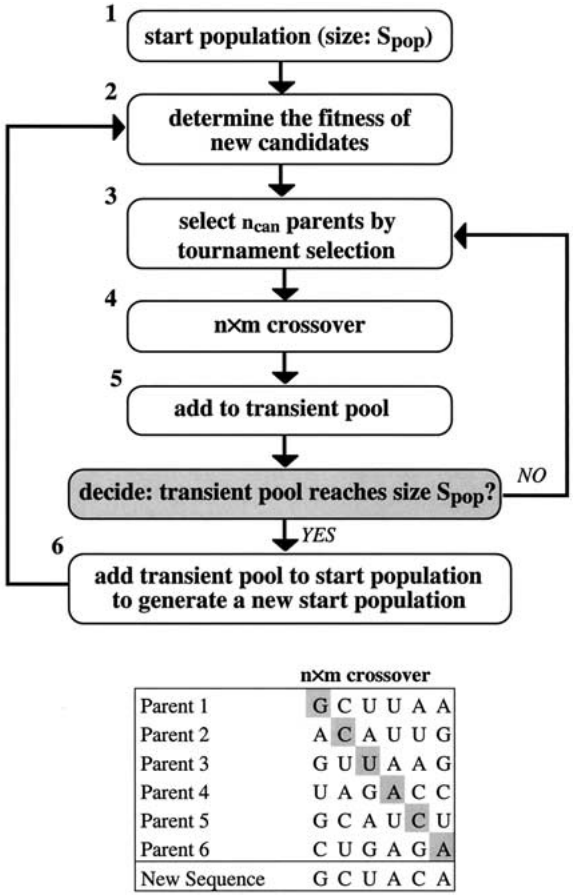


Figure 3. Flowchart for the genetic algorithm with n×m crossover. The effect of the new crossover operator is demonstrated using a case with six parent sequences (n = 6) of sequence length 6 and 5 crossover points (m = 5).

Free energy of RNA folding (ΔG) as a fitness function was calculated with the help of the Vienna RNA Package [12] which predicts the secondary structure of RNA according to Zuker [13]. In the case of variable length RNA structure optimization, the fitness function was calculated from ΔG and sequence length L_i and punishes long sequences:

$$f_i = \frac{\Delta G_i}{L_i^2}$$

A second type of optimization was based on a spin-glass model (for introduction see: www.informatik.uni-koeln.de/ls-juenger/projects/spinglass.html). Spin-glasses are usually collections of basic structural elements or building blocks that interact with each other when there are nearest neighbors on the underlying regular grid. Spinglass systems where first used in the field of condensed matter physics, e.g. to describe the

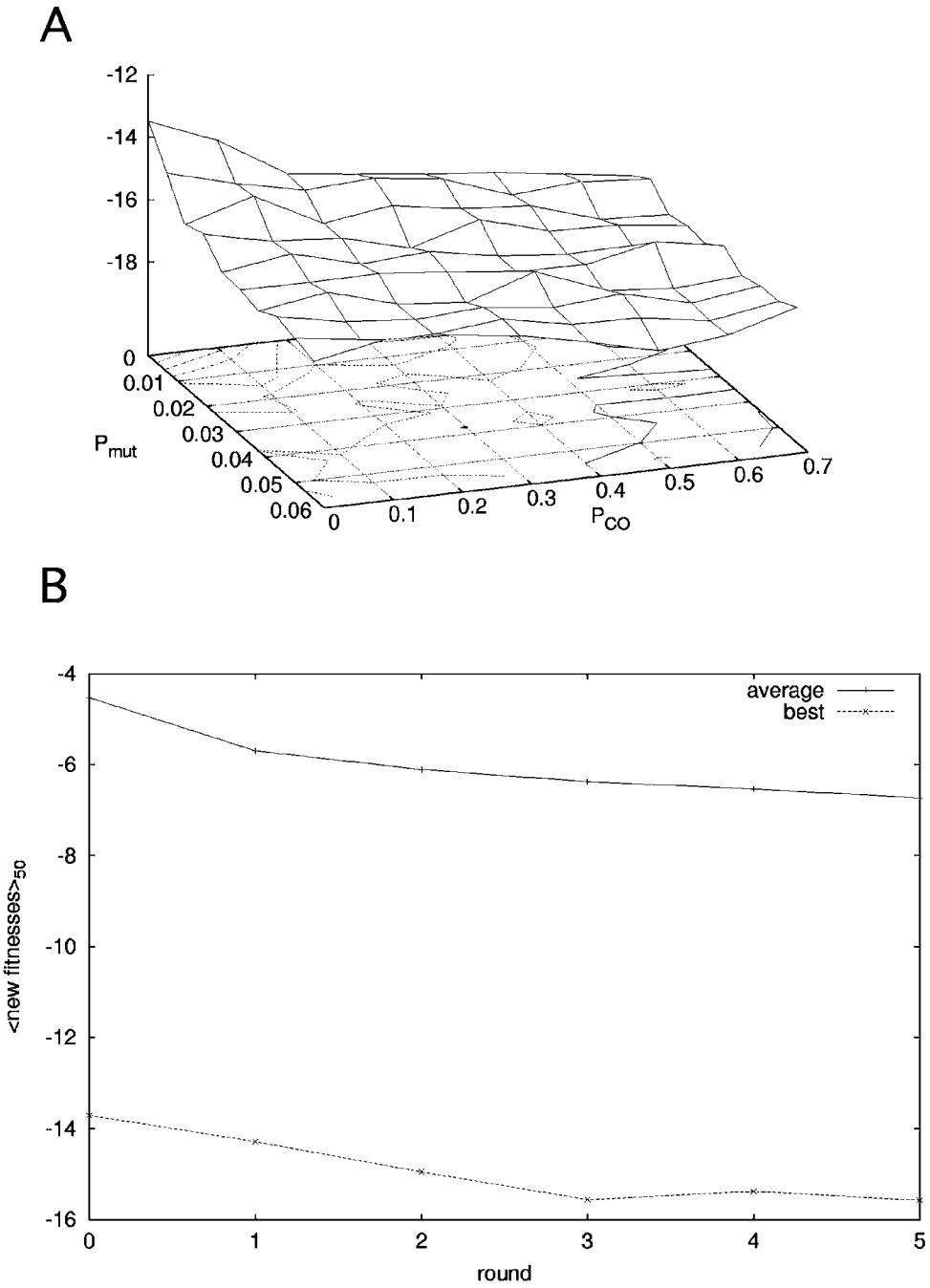


Figure 4. Results of standard GA simulations for RNA structure optimization (invariable sequence length). Since genetic methods are stochastic in nature we carried out 50 runs for every set of parameters and calculated the average fitness. (A) The diagram shows the average best fitness (ΔG) in dependence of mutation rate $\mathbf{P_{mut}}$ and crossover probability $\mathbf{P_{CO}}$. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. Sequence length is 32 (invariable). (B) Average fitness progression in experiments using the optimal set of parameters: $\mathbf{P_{mut}} = 0.05625$ and $\mathbf{P_{CO}} = 0.5$. The best new individuals per cycle and the average fitness of all new individuals per cycle are depicted.

interaction of spins, i.e. magnetic moments and meanwhile have been applied in a number of biological fields such as brain research [14] and protein analysis [15]. In our case we constructed a spinglass based on 30 different building blocks that were interacting within sequences of length 6–12. The fitness function was defined as a query on the spinglass matrix:

$$f_i = \frac{\sum_{j,k}^{L_i} S_{j,k,\sigma_j,\sigma_k}}{L_i^2}$$

with L_i = length of the sequence ($L_{min} = 6$ and $L_{max} = 12$), and $S_{j,k,\sigma_j,\sigma_k}$ = four-dimensional spinglass matrix for a given sequence with $1 \leq j,k \leq 30$ (30 symbol alphabet) and $1 \leq \sigma_j, \sigma_k \leq 12$. The matrix was initially filled with equally distributed random numbers in the range of $[-1,1]$.

Peptide synthesis

Peptide libraries were synthesized using the SPOT synthesis technology [16] on a modified Auto-Spot Robot ASP 222 (Abimed, Germany). Fmoc-protected amino acids were from Calbiochem-Novabiochem (Switzerland) and membranes derivatized with Fmoc-proline were from Hartmann Analytik (Germany). The Fmoc-group was removed with 20% piperidine in DMF and the membranes were washed in DMF. To form the water–cleavable safety-catch linker, Boc-lysine(Fmoc)-OH [17] was coupled to the membrane in a 10 x 20 spot format using 0.3 µl solution per spot. The unreacted sites were capped with acetic anhydride. The next 4–10 amino acids were coupled according to the sequence using 0,2 µl per spot three times. Unreacted sites were capped with acetic anhydride and the Fmoc-protection group was cleaved with 20% piperidine in DMF in each cycle. After a final deprotection and capping step, the side chain protection groups were cleaved with 50% TFA, 45% DCM, 3% Triisopropylsilane and 2% H_2O. Release of soluble peptides occurred through treatment with 200 µl 0.1 M Tris, 0.1 M NaCl, 0.1% PEG 8000 (pH 8.0). During this procedure, the C-terminal proline undergoes rearrangement to yield the diketopiperazine moiety. Since HPLC–analysis of random samples of the synthesized peptides revealed a purity of >95%, peptides were used without further purification. The mean yield of peptide per spot was determined by amino acid analysis to be 18 nmol ± 1 nmol.

For preparative peptide synthesis, a NovaSyn Crystal Peptide Synthesizer (Calbiochem-Novabiochem, Switzerland) was used. Fmoc-Pro-PEG-PS or PAL-PEG-PS resin (PerSeptive Biosystems, USA) was used to yield either the C-terminal diketopiperazine or the peptide-amide. Peptides were N-terminally acetylated, and finally deprotected and cleaved from the resin with 95% TFA, 3% triisopropylsilane and 2% H_2O. After precipitation with diethylether and lyophilization, the peptides were analyzed and purified by HPLC (Äkta 10, Pharmacia, Sweden) using a 218TP54 reversed phase column (Vydac, USA). A gradient of A = 0.1% TFA and B = 70% acetonitrile with 0.1% TFA was applied and the chromatogram was monitored at 214 nm. The identity of the product was confirmed by mass spectrometry. The peptide content after purification was determined by amino acid analysis.

Screening for thrombin inhibition

Thrombin was purchased from Sigma-Aldrich (Deisenhofen, Germany). Tos-Gly-Pro-Arg-MCA was from Chromogenix (Antwerp, Belgium). Peptides were tested for their thrombin inhibitory activity using a fluorogenic assay as described [18]. Briefly, 30 µl of peptide solution were used in a total assay volume of 200 µl. Residual thrombin activities were determined in comparison to a standard reaction without inhibitor which was set to 100%. Assays were carried out with the help of a robotic workstation (CyBi™-Screen-Machine, CyBio AG, Germany) including a Polarstar fluorescence reader (BMG, Germany). Pipet calibrations were largely based on results obtained by Michael Berg [19]. All assay data are mean values obtained from at least three independent measurements.

Determination of inhibitory activity (K_i)

The inhibition of thrombin was quantified using the fluorogenic assay as described. Briefly, data on the inhibitory constants Ki were obtained by monitoring the cleavage of fluorogenic substrate Tos-Gly-Pro-Arg-4-methyl-coumaryl-7-amide by thrombin in the presence of 0.125–4 µM inhibitor. The assay was carried out in 40 µl assay buffer (0.05 M Tris, 0.1 M NaCl, 0.1% PEG 8000, pH 7.6) with 10 µl of human thrombin solution (10^{-5} U/µl in assay buffer) and 140 µl of a solution of the fluorogenic substrate in assay buffer at a concentration of 30 µM. Aliquots of the test compound (10 µl) at various concentrations were added. Rates of hydrolysis of the substrate were measured by monitoring the reactions at 455 nm for the release of 7-amino-4-methyl-coumarin (AMC).

Fluorescence intensity was calibrated using AMC. The reaction reached a steady state within 3 min after mixing thrombin with the substrate and an inhibitor. The steady state velocity was then measured for 30 min. The kinetic data of the competitive inhibition (K_m, V_{max} and K_i) were analyzed using a Hanes plot of the kinetic data (A/V against A at several values of (i). Ki values were mean values obtained from four independent measurements.

Results

RNA structure optimization

We have designed a new group of genetic algorithms that was tailored to converge rapidly to a high-grade local fitness optimum. The key element of the algorithms is a new crossover operator (n×m crossover) that recombines n parent sequences at m crossover sites. Parents were selected by tournament selection. For each parent, a number of $\mathbf{n_{Tourn}}$ tournament participants were drawn randomly from the population and the fittest was taken as parent for subsequent steps of crossover and mutation.

To compare the performance of the new algorithms with classical GA constructs, we chose the optimization of RNA sequences with respect to folding energy as a first test system. In this case, sequence length was kept invariable. Starting with the standard GA (Figure 2), a number of runs were carried out to optimize and to evaluate the influence of several parameters. In Fig. 4 the results of the standard GA are depicted. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. Sequence length is $\mathbf{L} = 32$ (invariable). 50 runs for every set of parameters were carried out to calculate the average fitness from the best sequence of each individual run (in the following termed 'average best fitness'). Altogether these runs took 4.5 min of computer time. Crossover alone ($\mathbf{P_{mut}} = 0$) was able to achieve optimization of ΔG from -13.5 ± 1.65 (initial value) to -17.0 ± 1.86. By adding mutation (optimal parameters: $\mathbf{P_{mut}} = 0.05625 = 1.8/\mathbf{L}$ and $\mathbf{P_{CO}} = 0.5$), the performance could be slightly improved to reach average ΔG values of -17.5 ± 1.95. The five best sequences of these optimizations are depicted in Table 1. Alternatively, 50 cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 45$. Using optimal parameters ($\mathbf{P_{mut}} = 0.0375 = 1.2/\mathbf{L}$ and $\mathbf{P_{CO}} = 0.5$ in this case), the performance could be significantly improved to -20.8 ± 2.21. The five best sequences of these optimizations are also depicted in Table 1.

From the group of new algorithms, we first examined the GA with n×m crossover and no sequence length variation (Figure 3). In Figure 5 the results of the GA with n×m crossover are depicted. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. Sequence length is $\mathbf{L} = 32$ (invariable). Again, 50 runs for every set of parameters were carried out to calculate the average best fitness. Altogether these runs took 4.5 min of computer time. In particular, the dependence of the average fitness after five cycles of optimization on the number of parents $\mathbf{n_{can}}$ and the number of tournament competitors $\mathbf{n_{Tourn}}$ was analyzed. Using optimal parameters ($\mathbf{n_{can}} = 12$ and $\mathbf{n_{Tourn}} = 17$), an average fitness (ΔG) of -24.7 ± 3.0 could be achieved. The five best sequences of these optimizations are depicted in Table 1. Also depicted in Figure 5 (negative values for $\mathbf{n_{Tourn}}$) are the results for experiments with linearly increasing numbers of tournament competitors $\mathbf{n_{Tourn}}$. A value of $\mathbf{n_{Tourn}} = -3$ symbolizes a linear progression with three tournament competitors in cycle 1, six tournament competitors in cycle 2, nine tournament competitors in cycle 3 and so on.

Molecular design is usually not necessarily limited to sequences of the same length. In particular, in drug design, the goal often is to obtain small molecule drug candidates [20]. We therefore introduced a version of the GA with n×m crossover and sequence length variation from 20 to 32 nucleotides. To primarily obtain smaller molecules, the fitness function includes a penalty for large sequences. The results are depicted in Figure 6. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. In particular, the dependence of the average fitness after five cycles of optimization on the number of tournament competitors $\mathbf{n_{Tourn}}$ was analyzed. A hundred runs for every set of parameters were carried out to calculate the average best fitness. Altogether these runs took 13 min of computer time. Again, the dynamic mode, i.e. the experiment with linearly increasing numbers of tournament competitors $\mathbf{n_{Tourn}}$, showed a rather favorable performance. Using optimal parameters ($\mathbf{n_{Tourn}} = -10$), an average fitness ($\Delta G/L^2$) of -0.04 ± 0.005 could be achieved. The average sequence length after optimization was 23 with an average ΔG of -21.2. In contrast, a comparable experiment using the GA with n×m crossover and an invariable sequence length of 23 yielded an

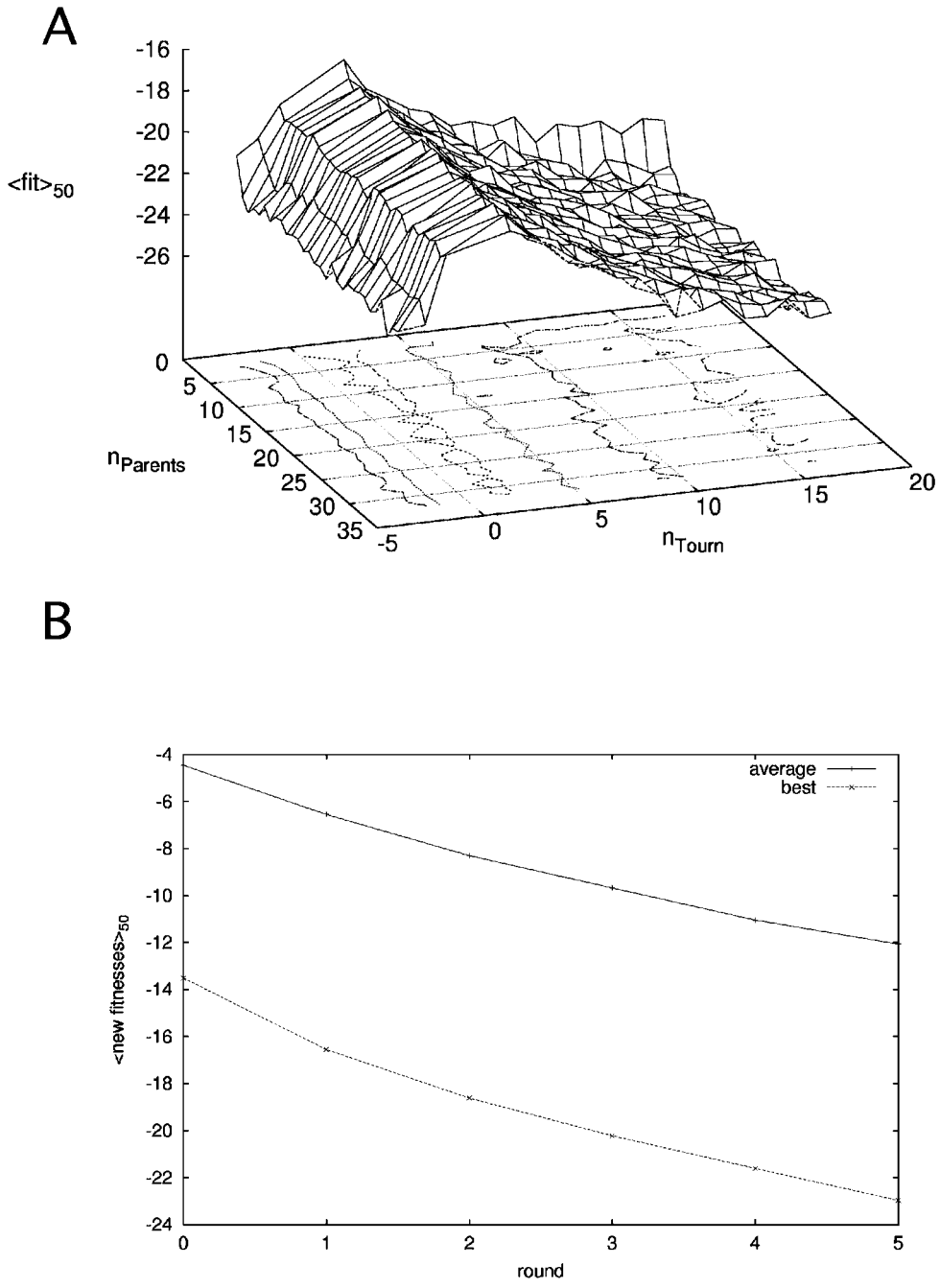
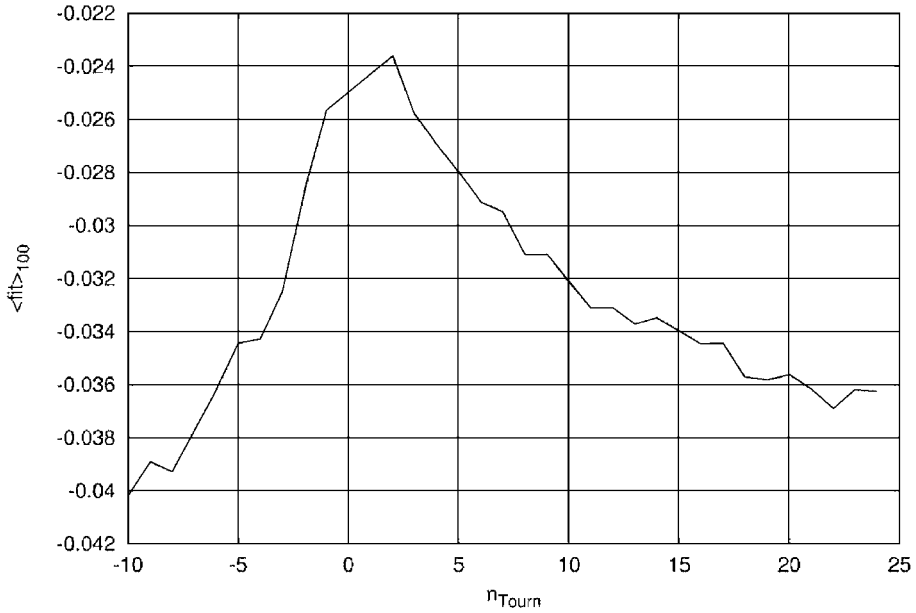


Figure 5. Results of GA with n×m crossover simulations for RNA structure optimization (invariable length). Again, 50 runs for every set of parameters were carried out and the average fitness was calculated. (A) The diagram shows the average best fitness (ΔG) in dependence of the number of parents $\mathbf{n_{can}}$ and the number of tournament competitors $\mathbf{n_{Tourn}}$. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. Sequence length is 32 (invariable). B) Average fitness progression in experiments using the optimal set of parameters: $\mathbf{n_{can}} = 12$ and $\mathbf{n_{Tourn}} = 17$. The best new individuals per cycle and the average fitness of all new individuals per cycle are depicted. Negative values of $\mathbf{n_{Tourn}}$ give the factor for a dynamically increasing selection pressure (see also Figure 6 for explanation).

A



B

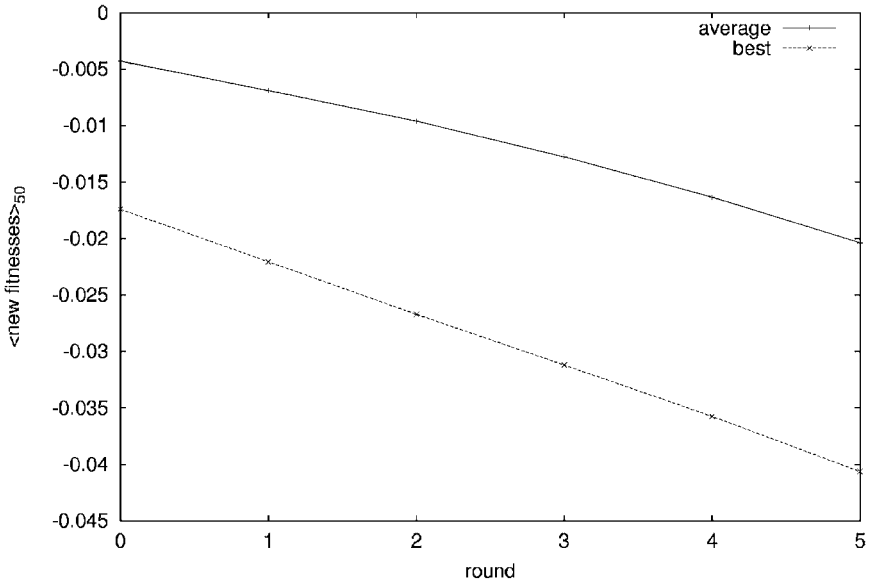


Figure 6. Results of GA with n×m crossover simulations for RNA structure optimization (variable length). (A) The dependence of the average fitness after five cycles of optimization on number of tournament competitors $\mathbf{n_{Tourn}}$ was analyzed. A hundred runs for every set of parameters were carried out and the average best fitness was calculated. (B) Average fitness progression in experiments using the optimal value of $\mathbf{n_{Tourn}} = -10$ (linear progression with ten tournament competitors in cycle 1, twenty tournament competitors in cycle 2, and so on). The best new individuals per cycle and the average fitness of all new individuals per cycle are depicted.

Table 1. Sequences and concomittant ΔG values of the best individuals after 50 runs of optimization for the indicated number of cycles using the fixed sequence length versions of the standard GA and the new GA with n×m crossover.

Standard GA					
Sequence	$\mathbf{P_{CO}}$	$\mathbf{P_{mut}}$	$\mathbf{S_{pop}}$	No. of cycles	Fitness (ΔG)
UAGGUGCGGAGCGCCGUUCUGGCGCUCCGCAC	0.5	0.05625	128	5	−25.9
CCCUGGCAGGGGCCCUCACCCCUGCUAGGGUA	0.5	0.05625	128	5	−25.6
CUGGCGGCCCCAUGCACACGUGGGGCCGUCGA	0.5	0.05625	128	5	−25.2
GAGAGGGCGCGCCAUUAUGUGGCGCGUCCUCC	0.5	0.05625	128	5	−25.0
GUCGCCGCGCACGUAGUCGAGCGCGGCGACAG	0.5	0.05625	128	5	−24.9
GCGCGGGGGCUCCGUUGUGGGCCCCCGCGCGU	0.5	0.0375	45	50	−28.9
CCGGUCUCCCGGGUAACCCGGGAGGCCGGGAU	0.5	0.0375	45	50	−28.7
AGGGGGGCCCGGCAGUGGGUGGGGCCCCCCUA	0.5	0.0375	45	50	−28.1
AGGCGGGCCUCGUGCUUGGGCACGGGGCCCGU	0.5	0.0375	45	50	−28.0
CCGGAGCUCCGGUAGAGACCGGAGCUCCGGCG	0.5	0.0375	45	50	−28.0

GA with n × m crossover					
Sequence	$\mathbf{n_{can}}$	$\mathbf{n_{Tourn}}$	$\mathbf{S_{pop}}$	No. of cycles	Fitness (ΔG)
AGGGGCCGGGGAGGGAAACCUCCCCGGCCCCC	12	17	128	5	−37.5
AGGCCCCGGCGUGGGAAACCACGCCGGGGCCC	12	17	128	5	−36.5
CGGCGGGGGGGGGGGGGACCUCCCCUCCGCCG	12	17	128	5	−35.0
GAUGGGCGCAGCCCCGCAAGGGGCUGCGCCCG	12	17	128	5	−34.5
GGGGGAGCCGGGCCCCGGGGCCCGGCUCCCUC	12	17	128	5	−34.4

$\mathbf{P_{CO}}$ = crossover probability; $\mathbf{P}_{mut}$ = mutation rate per position; $\mathbf{S_{pop}}$ = initial population size; $\mathbf{n_{can}}$ = on the number of parents; $\mathbf{n}_{Tourn}$ = tournament competitors.

average ΔG of -18.62 ± 2.01 (data not shown). The five best sequences of the optimizations using the GA with n×m crossover and variable sequence length are depicted in Table 2.

Optimization of polymers with large alphabets of building blocks

To test the feasibility of our computer-assisted design approach for the optimization of (bio)polymers that consist of large numbers of different building blocks, we first chose a simple *in silico* experiment. In this case the fitness function was deduced from a spinglass matrix as described.

In Figure 7 the results of the GA with n×m crossover are depicted. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. Sequence length is $\mathbf{L} = 32$ (invariable). A hundred runs for every set of parameters were carried out to calculate the average best fitness. Altogether these runs took 1 min of computer time. In particular, the dependence of the average fitness after five cycles of optimization on the number of parents $\mathbf{n_{can}}$ and the number of tournament competitors $\mathbf{n_{Tourn}}$ was analyzed. Using optimal parameters ($\mathbf{n_{can}} = 8$ and $\mathbf{n_{Tourn}} = -10$), an average fitness of -0.036 ± 0.003 could be achieved. Also depicted in Figure 7 is the fitness progression in experiments using the optimal set of parameters.

The results of the GA with n×m crossover and variable sequence length (6–12) are depicted in Figure 8. Again, five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. Sequence length is $\mathbf{L} = 32$ (invariable). Two-hundred runs for every set of parameters were carried out to calculate the average best fitness. In particular, the dependence of the average fitness after five cycles of optimization on the number of tournament competitors $\mathbf{n_{Tourn}}$ was analyzed. Using optimal parameters ($\mathbf{n_{Tourn}} = -7$), an average fitness of -0.052 ± 0.005 could be achieved. To compare the variable and invariable sequence length versions of the algorithm, we also analyzed the fitness progression at $\mathbf{n_{Tourn}} = -10$ (Figure 8B). Again, the variable length version of the algorithm reached significantly higher fitness values as compared to the fixed length algorithm. The set

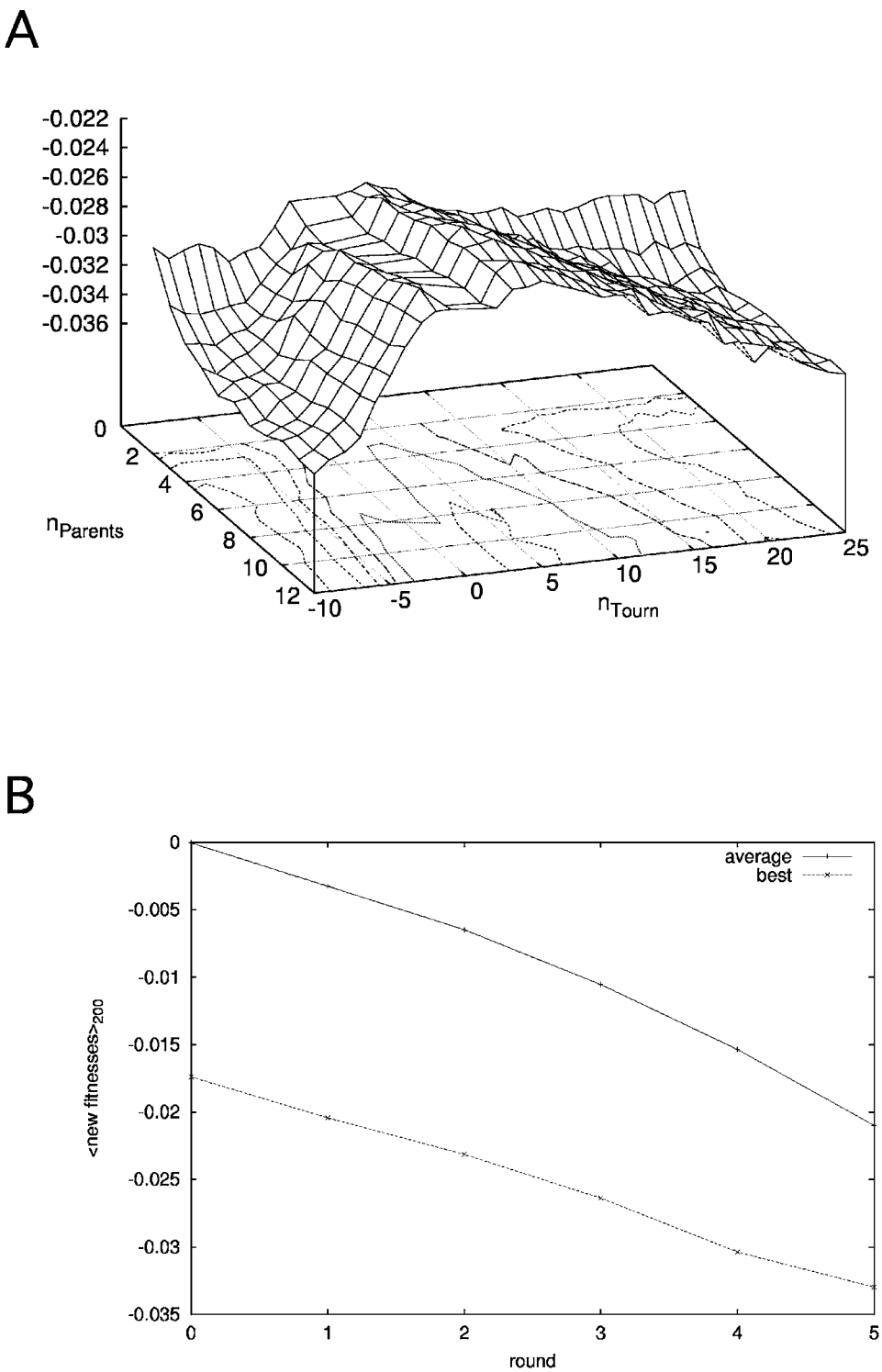


Figure 7. Results of the GA with n×m crossover simulations for spinglass-based sequence optimization (fixed length). (A) The diagram shows the average best fitness in dependence of the number of parents $\mathbf{n_{can}}$ and the number of tournament competitors $\mathbf{n_{Tourn}}$. Negative values of $\mathbf{n_{Tourn}}$ give the factor for a dynamically increasing selection pressure (see also Figure 6 for explanation). For each set of parameters, 100 separate runs were averaged. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. Sequence length was 32 (invariable). (B) Average fitness progression in experiments using the optimal set of parameters: $\mathbf{n_{can}} = 12$ and $\mathbf{n_{Tourn}} = -10$. The best new individuals per cycle and the average fitness of all new individuals per cycle are depicted.

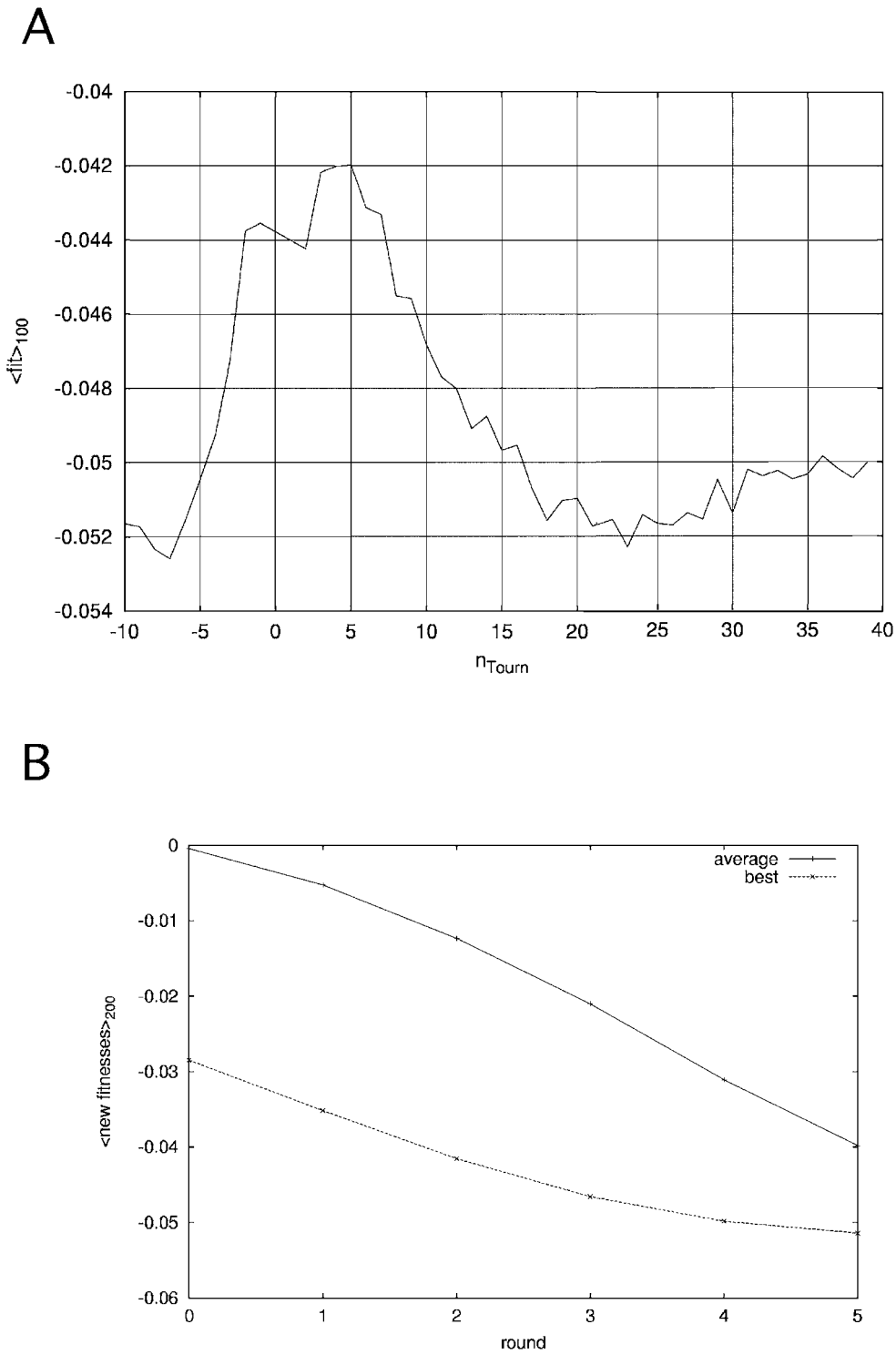


Figure 8. Results of the GA with n×m crossover simulations for spinglass-based sequence optimization (variable length: 6–12). (A) The diagram shows the average best fitness in dependence of the number of tournament competitors $\mathbf{n_{Tourn}}$. For each set of parameters, 200 separate runs were averaged. Five cycles of optimization were carried out with an initial population size of $\mathbf{S_{pop}} = 128$. (B) Average fitness progression in experiments using $\mathbf{n_{Tourn}} = -10$. The best new individuals per cycle and the average fitness of all new individuals per cycle are depicted.

Table 2. Sequences and concomittant fitness values of the best individuals after 50 runs of optimization for five cycles using the variable sequence length version of the new GA with n×m crossover.

GA with n × m crossover					
Sequence	$\mathbf{n_{can}}$	$\mathbf{n_{Tourn}}$	$\mathbf{S_{pop}}$	Length	Fitness
GCCCCCCCGCAAGGGGGGGC	12	17	128	20	−0.0570
GGGGGCGGGCAACCGCCCCC	12	17	128	20	−0.0540
GGCCCCCCGUGAGGGGGGCCC	12	17	128	21	−0.0535
GCCCCCCCGAAGGGGGGGGCA	12	17	128	21	−0.0531
GCCCCCCCCUUGGGGGGGGC	12	17	128	20	−0.0513

$\mathbf{S_{pop}}$ = initial population size; $\mathbf{n_{can}}$ = on the number of parents; $\mathbf{n_{Tourn}}$ = tournament competitors

of optimal parameters was kept for all subsequent applications of the variable length GA with n×m crossover.

Optimization of peptidic thrombin inhibitors

To evaluate the feasibility of computer-assisted molecular design as an experimental approach, we chose the design of thrombin inhibitors as a first test case. The study was conducted by screening a series of peptides, since this class of compounds has already proved to be promising, i.e. several good examples of highly active peptidic thrombin inhibitors are known [21, 22]. In addition, peptides are easily accessible by parallel solid phase synthesis. In this study, peptides were synthesized by a modified version of the SPOT-synthesis method [16] and covered sequence lengths from 6 to 12 amino acids. A total of 30 different building blocks were chosen, containing several non-natural moieties (see legend to Figure 10). All peptides were acetylated at the N-terminus and contained the diketopiperazine moiety at the C-terminus. For simplicity, we omitted extensive variation of the N- and C-termini, although this certainly restricts the attainable maximum inhibitory activity considerably [21, 22].

As in the case of previous optimizations, the overall process was again organized in design cycles (Figure 1). After synthesis, molecules were recovered in soluble form using the diketopiperazine forming reaction [17]. To start with, a library of 123 randomly chosen peptides (cycle 0) was generated. Approximately 2.7 nmol of each of the peptides was used without further purification to determine the thrombin inhibitory activity using a standard amidolytic assay [18]. In four subsequent cycles, 123 peptides were synthesized and characterized as described. As controls, active peptides of previous cycles were resynthesized and assayed in every subsequent cycle. Thrombin inhibition by the same control peptide synthesized in different cycles, i.e. residual thrombin activities determined in a standard fluorometric assay, differed by 5% (standard deviation).

Figure 9 displays residual thrombin activities after inhibition with peptides from each design cycle. Each subsequent cycle revealed molecules with increased thrombin inhibitory activity. Starting with residual thrombin activities of ≥81% in cycle 0, the most active peptide of cycle 4 (peptide NSCI 521, RccRRRWRK) revealed a residual thrombin activity of 26%. The fraction of molecules with significant activity (≤95% residual thrombin activity) increased from 3% in cycle 0 to 14–17% in cycles 2–4.

Peptides from cycles 0 and 1 showed the same sequences distribution, i.e. the same mean pair distance as calculated for a random sample of the concomittant sequence space. In particular, no preferential sequence motif could be traced. The picture changed significantly in subsequent cycles. In cycle 2, a considerable large fraction of peptides contained the N-terminal motif cXc– with X being any amino acid. This fraction increased from 8% in cycle 2 to 49% in cycle 4, and contained the most active molecules of cycles 2 and 3. Also in cycle 2, the first member of a sequence family appeared, containing the N-terminal motif RcXcXR–. The fraction of this family increased to 10% in cycle 4 and yielded the most active thrombin inhibitor of all four cycles of optimization, i.e. NSCI 521. The remaining sequences not belonging to either class of molecules, however, still approximated a random sequence distribution.

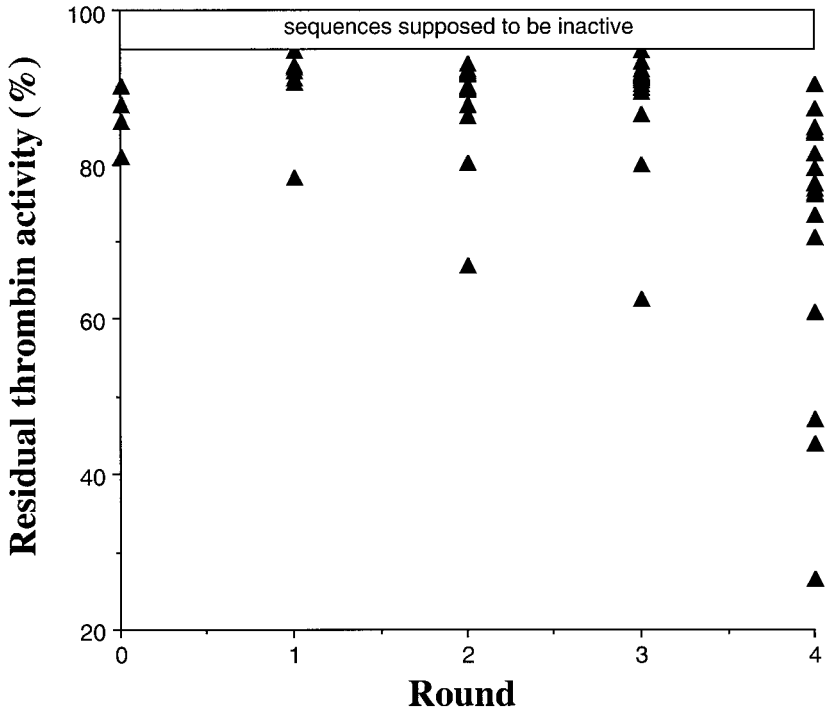


Figure 9. Optimization of peptidic thrombin inhibitors. Depicted are residual thrombin activities after inhibition with peptides from each cycle of optimization. Thrombin activities were determined in a standard amidolytic assay using a fluorogenic substrate. Compounds decreasing residual thrombin activity to only >95% were assumed to be inactive.

Experimental analysis of the fitness space

To explore sequence space in the direct vicinity of NSCI 521, we synthesized all 1-error mutants of the same length (Figure 10A). Interestingly, the vast majority of mutations in positions 1–4 dramatically diminished activity. Here, only two mutants, c3→N3 and c3→D3 showed a slightly higher thrombin inhibitory activity as compared to NSCI 521. Both mutations would not have been predicted in the context of other approaches which use physico-chemical descriptors to describe amino acid side chains [23]. In contrast to positions 1–4, mutations in positions 5–10 hardly impaired thrombin inhibitory activity. The only exception is mutation R5→c5 which leads to a complete loss of activity. On the other hand, only very few substitutions, most of them found in position 5, slightly increased thrombin inhibitory activity as compared to NSCI 521. Despite the limited set of molecules analyzed so far, it may be justified to assume that NSCI 521 came very close to the local optimum which in fact turned out to be a rather narrow peak in the fitness landscape.

As substitution experiments have already indicated, the important part for the thrombin inhibitory activity of NSCI 521 resides exclusively in the N-terminal tetrapeptide. To substantiate this assumption, we investigated all possible truncated forms of the peptide (Figure 10B). Only molecules containing the N-terminal tetrapeptide turned out to exhibit considerable thrombin inhibitory activity. However, all truncated variants were significantly less active as compared to the full length peptide.

Kinetic analysis of selected inhibitory peptides

Both NSCI 521 and its N-terminal tetrapeptide were purified and subjected to further kinetic analysis using a standard thrombin inhibitory assay with fluorogenic substrate Tos-Gly-Pro-Arg-4-methyl-coumaryl-7-amide. The full length peptide NSCI 521 binds to thrombin with a K_i of 297 nM and the N-terminal tetrapeptide exhibits a K_i of 10.24 μM. In comparison to other peptidic inhibitors of thrombin, the newly discovered inhibitors exhibit considerable activity. Peptidic inhibitors that are known to be significantly more active than NSCI 521 are either substantially larger such as hirudin, which covers a considerable surface area of thrombin, or carry (terminal) modifications such as DuP 714 [22]. All these inhibitors, how-

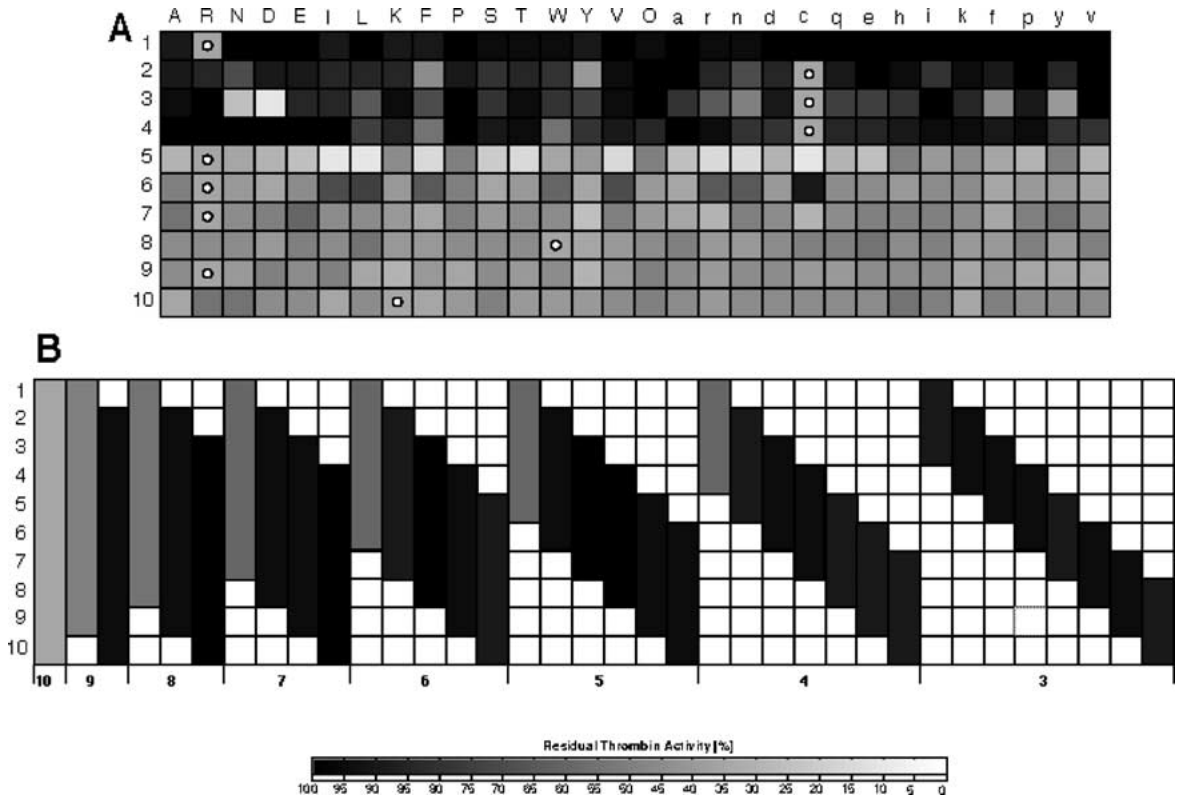


Figure 10. Structure-activity study of peptide NSCI 521 (RcccRRRWRK), the most active inhibitor from optimization cycle 4. Residual thrombin activities of 1-error mutants (A) and all possible truncated forms of peptide NSCI 521 were determined in a standard amidolytic thrombin assay. Abbreviations of amino acid moieties are: A = L-alanine, R = L-arginine, N = L-asparagine, D = L-aspartic acid, E = L-glutamic acid, I = L-isoleucine, L = L-leucine, K = L-lysine, F = L-phenylalanine, P = L-proline, S = L-serine, T = L-threonine, W = L-tryptophane, Y = L-tyrosine, V = L-valine, O = L-ornithine, a = D-alanine, r = L-phenylglycine, n = L-norleucine, d = D-aspartic acid, c = cyclohexylalanine, q = D-glutamine, e = D-glutamic acid, h = D-histidine, i = citrulline, k = D-lysine, f = D-phenylalanine, p = D-proline, y = D-tyrosine, v = D-valine.

ever, were not part of the restricted compound space accessible in the experiment described above.

Discussion

This article describes an implementation of a new genetic algorithm that has been tailored to meet the demands of *de novo* drug design, i.e. efficient optimization based on small training sets and few design cycles. A parameterization study shows that GAs are rather robust against small changes in population size [data not shown]. Noticeable differences have been observed with changes in mutation rate and crossover probability. As an appropriate error-threshold for mutation rate, we used the reciprocal of the sequence length [24, 25]. For standard GAs, optimal mutation rates have been shown to depend on recombination [26, 27]. For high mutation probabilities, recombination can push the population over the error threshold [27]. In the context of GA-based drug design, high crossover probability was indeed observed to hinder fitness convergence or to lead to premature convergence [28]. However, in this case, relatively high mutation probability turned out to restore a good performance.

A major influence on the performance of the algorithm has been seen for differences in the number $\mathbf{n_{Tourn}}$ of tournament participants that were recruited to select parent sequences for further variation. Within the optimization procedure, the entity $\mathbf{n_{Tourn}}$ is equivalent to a kind of selection pressure. The larger $\mathbf{n_{Tourn}}$, the higher is the probability to include the most fittest molecules into the tournament, i.e. into the subsequent variation steps. With $\mathbf{n_{Tourn}}$ approaching the population size ($\mathbf{S_{pop}}$) however, each tournament leads to the same fittest sequence. At this point, crossover is

not effective any more, i.e. does not lead to new sequence species. Variation at this point is thus reduced to mutation. A progression, e.g. linear increase of $\mathbf{n_{Tourn}}$ with every subsequent design cycle could be advantageous. In early cycles, a wide search space can be explored and a rapid convergence towards high fitness values including a preferentially local search is achievable in later cycles. In the interplay with $\mathbf{n_{Tourn}}$, the n×m crossover operator turned out to be a powerful tool in combining sequence information of many parent molecules that previously have been selected according to their fitness. The search is therefore quickly restricted to 'promising' areas in fitness space. In the first implementation of the novel algorithm we chose the simplest case where the number of parent sequences n_{can} is equal to the number of elements in the sequence (L), i.e. each two neighboring building blocks in a polymer sequence are separated by a crossover point. Thus, the n×m crossover resembles more a particularly intelligent kind of point mutation, that favors building blocks from successful parent molecules at each position. In this way it works similar to certain doping strategies [8] that have been introduced into the field of evolutionary molecular design.

Using the new GA optimization based on small training sets and a small number of design cycles was indeed successful. Optimizations based on theoretical fitness functions suggested, that about five design cycles with approximately 128 compounds per cycle are generally sufficient to obtain relatively good molecular solutions. Both RNA molecules as well as multiletter alphabet biopolymers such as peptides could be optimized. Computer-assisted drug discovery as an experimental approach also proved to be efficient. Within five cycles of optimization, a novel nanomolar inhibitor of thrombin was generated *de novo*. Mutational analysis showed that the inhibitor resides close to the local optimum in the fitness landscape. On the other hand, efficient convergence in the direction of one class of molecules does not prevent other classes of molecular solutions to appear; as has been observed in cycle 4 of the thrombin inhibitor search. It is possible, that even better solutions could be found in subsequent cycles. However, as with any sampling strategy that is not testing all compounds of a given library, there is certainly no guarantee to find the absolute optimum. In addition, it must be stressed that overinterpretation of poor–quality data can lead to false positives and false negatives, i.e. false QSPRs and thus slows down or completely prevents optimization. In particular, with lower quality experimental data, initial analysis of a compound may suggest an activity only to be contradicted by a subsequent analysis of the same or related compounds. The main limitation of the current version of the computer-assisted drug discovery approach, however, is the inability to perform optimizations on nonpolymeric small molecules. This restriction could be overcome, e.g. by using a fragment-based type of encoding [29, 30].

In contrast to present day drug discovery technology, our novel approach is neither time-consuming nor costly, since only small numbers of compounds have to be tested. It thus seems realizable to increase the biological information gain even early in the drug discovery process by including a multitude of additional assay parameters such as ADME/T characteristics to optimize many relevant properties in parallel. Here, however, appropriate weighting of individual contributions of different parameters to the fitness function has to be fixed.

Acknowledgements

This work was in part supported by grant 203.32329-4/1-99 (12) of the Ministry of Research of Lower Saxony and grant BioFuture 0311852 from the Bundesministerium für Forschung und Technologie, Germany. Technical support from Cybio and BMG is greatly acknowledged. The authors are grateful to Prof. H.-J. Fritz, Prof. H. Kern and all members of the Institute for Plant Pathology at the University of Göttingen for providing laboratory space.

References

1. Maddalena, D. J., Exp. Opin. Ther. Patents, 8 (1998) 249–258.
2. Holland, J. H., Adaptation in natural and artificial systems, MIT Press, Cambridge, MA, 1992.
3. Goldberg, D. E., Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Publishing Company, Inc., Reading, Massachusetts, 1989.
4. Terfloth, L. and Gasteiger, J., Drug Discovery Today, 6 (2001) S102–S108.
5. Glen, R. C. and Payne, A. W. R., J. Comp.-Aided Mol. Design, 9 (1995) 181–202.
6. Devillers, J. J., Chem. Inf. Comput. Sci., 36 (1996) 1061–1066.
7. So, S.-S. and Karplus, M. J., Med. Chem., 39 (1996) 1521–1530.
8. Tomandl, D., Schober, A. and Schwienhorst, A., J. Comp.-Aided Mol. Design, 11 (1997) 29–38.
9. Pohlheim, H. 1998, GEATbx: Genetic and Evolutionary Algorithm Toolbox for use with MATLAB Version 1.92. [http://www.geatbx.com/docu/algindex.html]

10. Illgen, K., Enderle, T., Broger, C. and Weber, L., Chem. Biol., 7 (2000) 433–441.
11. Weber, L., Wallbaum, S., Broger, C. and Gubernator, K., Angew. Chem., 107 (1995) 2452–2454.
12. Hofacker, I. L., Fontana, W. and Stadler, P. F., Monatshefte Chem., 125 (1994) 167–188.
13. Zuker, M., Nucl. Ac. Res., 9 (1981) 133–148.
14. Matthysse, S., Neurochem. Res., 16 (1991) 397–408.
15. Lin, C. Y., Hu, C. K. and Hansmann, U. H., Phys. Rev. E, 64 (2001) 052903.
16. Frank, R., Tetrahedron, 48 (1992) 9217–9232.
17. Bray, A. M., Maeji, N. J. and Geysen, H. M., Tetrahedron Lett., 31 (1990) 5811–5814.
18. Wirsching, F., Opitz, T., Dietrich, R. and Schwienhorst, A., Gene, 204 (1997) 177–184.
19. Berg, M. *Methodenentwicklung und Etablierung von Fluoreszenzassays zur Wirkstoffsuche im 1536-Well Format*, Universität Karlsruhe, Karlsruhe, 1998.
20. Lipinski, C. A., Lombardo, F., Dominy, B. W. and Feeney, P. J., Adv. Drug Delivery Rev., 23 (1997) 3–25.
21. Claeson, G., Blood Coag. Fibrinol., 5 (1994) 411–436.
22. Hauptmann, J. and Stürzebecher, J., Thrombosis Res., 93 (1999) 203–241.
23. Schneider, G. and Wrede, P., Biophys. J., 66 (1994) 335–344.
24. Eigen, M., Die Naturwissenschaften, 58 (1971) 465–523.
25. Eigen, M. and Schuster, P., Die Naturwissenschaften, 64 (1977) 541–565.
26. Barnett, L. J., Theor. Biol., subm. (2000).
27. Ochoa, G. and Harvey, I., *In* Banzhaf, W. and Reeves, C. (eds), Foundations of Genetic Algorithms, vol. 5. Morgan Kaufmann, San Francisco, 1999, pp. 245–264.
28. Douget, D., Thoreau, E. and Grassy, G. J., Comp.-Aid. Mol. Design, 14 (2000) 449–466.
29. Westhead, D. R., Clark, D. E., Frenkel, D., Li, J., Murray, C. W., Robson, B. and Waszkowycz, B., J. Comp.-Aid. Mol. Design, 9 (1995) 139–148.
30. Schneider, G., Lee, M.-L., Stahl, M., Schneider, P. J., Comp.-Aid. Mol. Design, 14 (2000) 487–494.