Simulated molecular evolution in a full combinatorial library

Katrin Illgen*, Thilo Enderle, Clemens Broger and Lutz Weber*

Background: The Darwinian concept of 'survival of the fittest' has inspired the development of evolutionary optimization methods to find molecules with desired properties in iterative feedback cycles of synthesis and testing. These methods have recently been applied to the computer-guided heuristic selection of molecules that bind with high affinity to a given biological target. We describe the optimization behavior and performance of genetic algorithms (GAs) that select molecules from a combinatorial library of potential thrombin inhibitors in 'artificial molecular evolution' experiments, on the basis of biological screening results.

Results: A full combinatorial library of 15,360 members structurally biased towards the serine protease thrombin was synthesized, and all were tested for their ability to inhibit the protease activity of thrombin. Using the resulting large structure–activity landscape, we simulated the evolutionary selection of potent thrombin inhibitors from this library using GAs. Optimal parameter sets were found (encoding strategy, population size, mutation and cross-over rate) for this artificial molecular evolution.

Conclusions: A GA-based evolutionary selection is a valuable combinatorial optimization strategy to discover compounds with desired properties without needing to synthesize and test all possible combinations (i.e. all molecules). GAs are especially powerful when dealing with very large combinatorial libraries for which synthesis and screening of all members is not possible and/or when only a small number of compounds compared with the library size can be synthesized or tested. The optimization gradient or 'learning' per individual increases when using smaller population sizes and decreases for higher mutation rates.

Address: Preclinical Research, F. Hoffmann-La Roche AG, CH-4070 Basel, Switzerland.

*Present address: morphochem AG, Gmunder Straße 37-37a, D-81379 München, Germany.

Correspondence: Lutz Weber
E-mail: lutz.weber@morphochem.de

Key words: combinatorial chemistry, combinatorial optimization, evolutionary chemistry, genetic algorithm, thrombin inhibitors

Received: **17 December 1999**
Revisions requested: **13 January 2000**
Revisions received: **3 March 2000**
Accepted: **15 March 2000**

Published: **30 May 2000**

Chemistry & Biology 2000, **7**:433–441

1074-5521/00/$ – see front matter

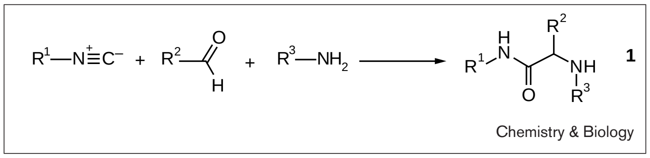
Introduction

The idea of generating a large number of diverse small molecules by combinatorial chemistry was first proposed by Ivar Ugi in 1961 [1] using the Ugi four-component reaction (4CR) and the systematic, combinatorial variation of the corresponding starting materials. Thus, by using 100 different amines, aldehydes, carboxylic acids and isonitriles, it is theoretically possible to generate 100 million different reaction products. This number is close to the number of different, unique antibodies that is encoded in the DNA of the human genome for the primary immune response. In analogy to the library of possible antibodies, one could argue that within that library of 100 million Ugi-type 4CR products one might find binding ligands for almost any target protein; however, it is still far beyond our practical possibilities to synthesize and evaluate individually the biological properties of such a large number of small molecules.

An astonishingly efficient, combinatorial process is used by nature both to store the genomic information for all possible primary antibodies and to find those antibodies that bind tightly to a given target molecule. Thus, the variable DNA sequence of a unique antibody is assembled from a predefined set of 100–1000 different V, J and D gene segments on the chromosomal DNA — a combinatorial principle, similar to the above Ugi-type multi-component condensation reaction. In the primary immune response to the antigen, a whole library of antibodies is generated. Out of this initial population, antibodies are selected that bind with a certain affinity to the given antigen. Starting with this binding sub-population, new antibodies, which are not encoded on the chromosomal DNA, are generated by somatic hypermutations. The entire process is repeated several times until high-affinity antibodies are generated. This evolutionary immune response is therefore an affinity-driven feedback cycle that yields antibodies with high affinity to the target antigen [2].

Similarly, medicinal chemistry, aiming at the discovery of high-affinity ligands for target molecules, is about understanding the relationship between the abstraction 'chemical structure' and the properties of the corresponding real molecules, called structure–activity relationships (SARs). Many methods have been introduced to aid the explicit understanding of SARs, such as molecular modeling or comparative molecular field analysis (CoMFA). An alternative to these methods is the use of heuristic method

Figure 1



An Ugi-type three-component reaction was used to generate the library.

evolutionary algorithms that implicitly discover the relationship between chemical structures and biological properties. Artificial evolutionary selection methods have been developed, for example by selecting peptides (e.g. phage display or antibodies) or oligonucleotides (SELEX procedure) that bind to the target of choice by using nature's amplification and selection machinery. A method that selects useful, 'drug-like' small molecules from very large combinatorial compound libraries by an efficient evolutionary process would, however, be of even higher interest for many applications in drug discovery, crop protection and material sciences.

The concept of darwinian selection and evolution has inspired the development and application of evolutionary programming [3] or genetic algorithms (GAs) [4] to find optimal solutions for combinatorial problems [5] in multidimensional and very large search spaces. Artificial evolutionary optimization has recently been introduced to find molecules with high biological activity from large virtual combinatorial libraries of molecules without the need to synthesize and to determine the biological activity of all members of this library. Previously we reported the selection of thrombin inhibitors from a combinatorial library of Ugi-type reaction products [6] using a GA. A similar approach has also been used to find peptidic trypsin inhibitors [7] and better substrates for stromelysin [8]. The iterative synthesis and biological testing of small sub-libraries from a large virtual library was combined with the measurement of the biological activity of individual members of those sub-libraries. Biological activities as a selection feedback for the GA were used to propose new molecules for synthesis. After several cycles of synthesis and testing, molecules with desired properties were found.

Here we try to address problems that have not been solved by those early examples. What is the optimization efficiency of GAs in finding the best molecules out of such a combinatorial library? What are suitable parameters for a GA-driven darwinian-like evolutionary process for small molecules?

To answer these questions, it is necessary to know *a priori* all solutions, that is the biological affinities of all members of the library against the target protein of interest. The resulting structure–activity landscape can than be used as a model for very large libraries comprising the search space for GA-driven experiments. By running GAs within this search space *a posteriori*, we can also optimize various parameters of the GAs and evaluate their influence on the search efficiency of GAs in simulated molecular evolution experiments.

We have therefore synthesized a complete combinatorial library of 15,360 parallel Ugi-type reaction products having a structural bias towards the serine protease thrombin as a test case. We determined the biological activity of all members of this library with a chromogenic enzyme assay.

Results

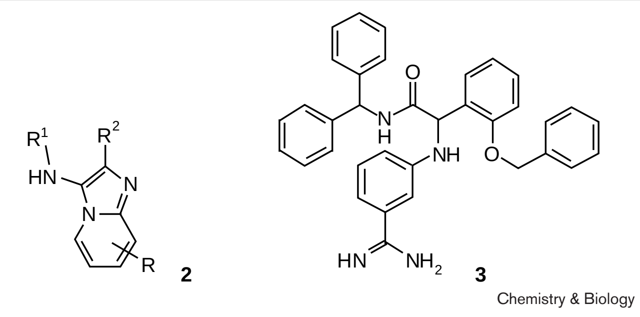
Library synthesis and testing

Thrombin was selected as the target protein for the combinatorial optimization of active-site inhibitor molecules because many structural requirements of this enzyme are known. We previously showed that N,N′-disubstituted amino-acid amides **1** are potentially interesting structures for highly active thrombin inhibitors [6]. These molecules are accessible by an Ugi-type three-component reaction (3CR) using isonitriles, aldehydes and amines as the reaction components (Figure 1).

We chose 80 aldehydes (**A1**–**A80**), 12 amines (**B1**–**B12**) and 16 isonitriles (**C1**–**C16**) for synthesis to give a combinatorial library of 80 × 12 × 16 = 15,360 reaction products. The respective isonitriles and aldehydes were selected to cover a broad range of chemical diversities by using large and small aliphatic, aromatic, heteroaromatic, benzylic and substituents with hydrogen-bond donor and acceptor substituents. The amines **B1**–**B12** were chosen to provide a structural bias towards the thrombin arginine-binding S1 pocket. Most of these arginine-mimetic amines were already known to have an affinity for thrombin in the higher micromolar range. Because of this structural bias, one can expect that the molecules of this library will cover a broad range of affinities towards thrombin.

The 15,360 reactions were performed in parallel using 384-well plates and a final reaction volume of 15 μl. Solutions of the corresponding starting materials in methanol (0.1 M) were dispensed with a robotics system. A complete conversion in all reactions was assumed in order to calculate the inhibitory concentrations (IC_{50}) of the crude reaction products which were determined using a conventional chromogenic assay using the paranitroanilide of D-Phe–Gly–Arg as substrate. The respective molecular weights were calculated for all 15,360 reaction products and verified using ionspray mass spectroscopy. The expected molecular mass ions of **1** were found with high intensity in about 60% of the 15,360 products (full details will be published elsewhere). Three amines (**B0**, **B2** and **B3**) gave the imidazo[1,2-a]pyridine derivatives **2** (Figure 2) instead of **1** in a [1 + 4] cycloaddition reaction as described previously [9].

Figure 2



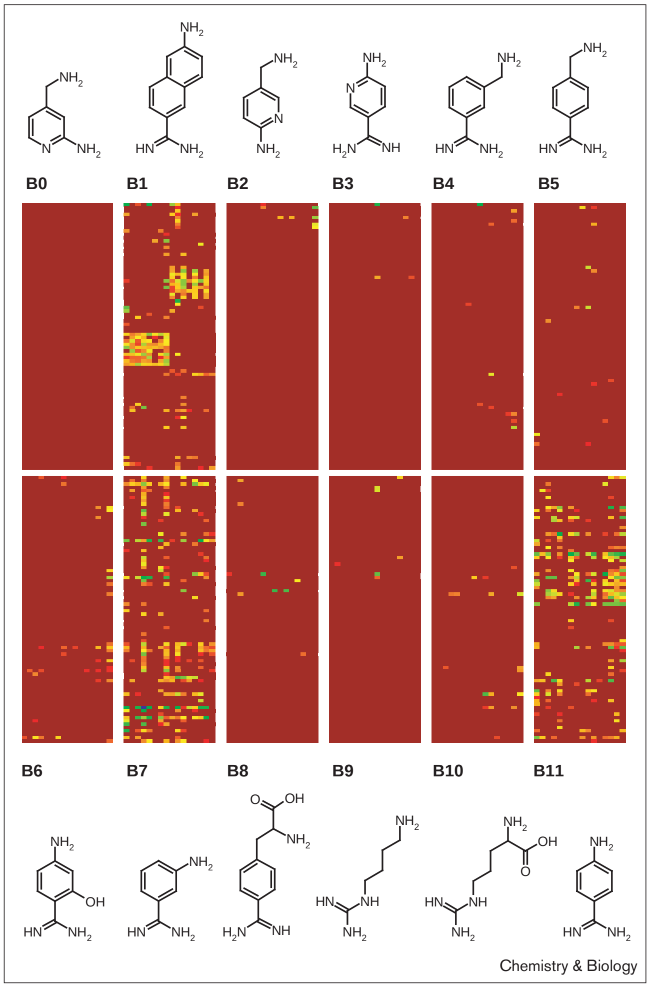
Side-product **2** is formed with the amines **B0**, **B2** and **B3**. Product **3** is a 2 nm inhibitor of thrombin.

Figure 3 shows the IC_{50} values of all reaction products, coded by spectral colors in a range between 100 μM (red) and 100 nM (green). Values outside of this range are coded brown ($IC_{50} > 100$ μM) and blue ($IC_{50} < 100$ nM). Each rectangle corresponds to one amine **B**, whereas the aldehydes and isonitriles are varied in the vertical and horizontal direction, respectively. Thus, Figure 3 represents a two-dimensional projection of the three-dimensional SAR. Re-synthesis of the most active combination (aldehyde **A69**, amine **B7** and isonitrile **C3**) on a larger scale gave **3** with an isolated yield of 40%. Re-testing the racemic compound **3** with a full dose–response curve yielded a k_i value of 2 nM against thrombin (IC_{50} values are dependent on the concentrations of substrate and protein used, whereas K_i values are independent of these concentrations and can be estimated from the IC_{50} value by a constant factor. IC_{50} values were estimated for all compounds by measuring the thrombin inhibition at 100, 10, 1, 0.1 and 0 μM for each of the 15,360 products. The IC_{50} value of inactive compounds was set arbitrarily at 150 μM). In summary, out of all 15,360 products, only 9 (0.059%) had IC_{50} values below 1 μM, 54 (0.352%) were between 1 μM and 10 μM and 675 (4.395%) were between 10 μM and 100 μM. Hence, the fraction of active products in the library is rather low (<5%) despite the biased nature of our choice of amines.

Encoding molecules

Natural evolutionary systems are composed of two layers: encoding (genotypes) and realization (phenotypes). Both layers are connected by an operator that provides the recipe for how to construct the phenotype from the genotype. For example, the triplet UUC in a gene encodes for the molecular building-block phenylalanine in proteins. It is important to note that the genotype does not reflect any physico-chemical property of the encoded phenotype directly. Evolutionary selection in biological systems may therefore be regarded as a combinatorial optimization system that uncovers implicitly the relationship between the sequence of a genome and the fitness of its phenotype.

Figure 3



Color-coded thrombin inhibitory concentrations of the 12 × 16 × 80 library.

Artificial evolution for small molecules requires the implementation of genes for small molecules. By convention, molecules are generally represented on the basis of valence-bond theory [10]. Contrary to such general encoding schemes, combinatorial libraries allow an efficient encoding using arbitrary encoding schemes. Binary strings, alphanumeric values or real-valued strings have been used to encode the starting materials of a combinatorial library [10].

The encoding scheme that is used may have an impact on the outcome of the artificial evolution optimization experiment. Thus, cross-over applied to genomes in which the variable parts of a compound structure are encoded by a binary representation may result in a different outcome to cross-over applied to a real-valued representation. In a real-valued encoding scheme, a cross-over may intersect the genome only in between these real-values (i.e. building blocks or substituents). In the real-valued representation, the building blocks of the respective parents are exchanged among the children but both their number and nature will remain the same. In the binary string representation, however, a cross-over may also take place within the binary string that represents a given building block. As a result, the children might contain different building blocks to their parents. This is also the case for the encoding of proteins in nature, for example, a cross-over in-between the triplet UUC (phenylalanine), such as UU/C, may result in the novel building block UUA (leucine) which was not present in the DNA of the parent protein encoding. This cross-over strategy has the effect that the cross-over operator is effectively a mixture of cross-over and mutation where the mutation probability increases with an increasing length of the encoding string for a given building block. For example, there are four possibilities to cut UUC: at the beginning and the end (/UUC or UUC/) as well as twice in between (U/UC or UU/C). Whereas the first strategy of replacing only whole building blocks appeals more to the intuition of chemists, the latter is more similar to natural cross-over in DNA.

For our artificial evolutionary experiments we used both binary and real-valued representations of the corresponding building blocks. For the binary representation, the educt numbers were first converted into a binary number, for example educt **A5** was encoded by a 'gene' 0000101. Next, the complete bit string for the reaction product was constructed — in our case from the three binary numbers of the aldehyde, amine and isonitrile — resulting in the bit string **0000101 1010 0010** for the reaction of starting materials **A5**, **B10** and **C2**. For the real-valued decimal representation, the genome of the final molecule is simply **5 10 2**.

The genetic algorithm

Genetic algorithms have been applied to a series of problems in molecular diversity and combinatorial chemistry. The building-block hypothesis [3] assumes that when low-order, 'fit' schemes (building blocks, or in our context the starting materials **A**, **B** and **C**) are combined, schemes of higher order and better fitness may result. The ability to produce meaningful solutions by combining building blocks was suggested to be a primary source of the search power of the GAs [5].

Operators for artificial GAs have been introduced on the basis of their natural counterparts, explained below.

Replication

Nature replicates good solutions in each generation in order to remember them; however, computer-based approaches allow a more efficient way to remember good solutions. Each experimental result may be stored and used at a later time without the need for repeated synthesis and biological testing. The 'replication' of **0000101 1010 0010** or **5 10 2** would yield the same bit string.

Mutation

This operator changes a bit in the chromosome with a certain probability; for example, **0000101 1010 0010** or **5 10 2** may be mutated to **0000101 1010** 1010 or **A5B10C9**. In our GA implementation, a 1% mutation rate per bit for the binary encoding corresponds to an 15% mutation rate for the whole, 15-bit chromosome.

Cross-over

This operator takes two strings, cuts each of them into two parts and reassembles them to arrive at two new valid strings. For binary strings, this may result in **0000101 10/10 0010** and 0010101 11/10 1011 to yield **0000101 10**10 1011 and 0010101 11**10 0010** as children. For real-valued strings, **5 10 2** and **21 14 11** may give **5 10 11** and **21 14 2**. Care was taken to ensure the same cutting probability between A and C strings, by using a virtually 'circular' bit string.

Selection and generation of the new population

The GA was set up by randomly choosing N strings encoding for N reaction products in the starting population. The experimental biological results for the corresponding reaction products were looked up in the result database of all 15,360 products and the encoding strings were than sorted according to their biological activities. The N best strings of the list of all parent generations were afterwards used to generate N new strings (child population) with a given cross-over and a given mutation probability. For cross-over, two individuals were chosen randomly from this list. The generation of already evaluated individuals ('replication') was not allowed for the reasons outlined above and was contrary to other procedures [7,8]. As the N best compounds were always remembered and used to generate N new children, this method could be called the 'best half' as opposed to the 'best third' method [10]. This general approach is the same as in our previous experiment [6] that used binary encoding, a population size of 20, a crossover rate of 100% per chromosome and a mutation rate of 1% for each bit.

Simulated evolution experiments

For the first time, the knowledge the biological activities of the complete combinatorial library allows us to study the optimization behavior of GAs on a real structure–activity landscape. The goal was thus to find optimal GA parameters that discover the best reaction products out of the 15,360 with a minimum number of experiments.

Finding combinations of optimal parameters for a GA is a combinatorial optimization problem by itself and depends also on the structure of the search space [5]. In this study, the search space is given by the four-dimensional structure–activity landscape spanned by the starting materials **A**, **B** and **C** and the corresponding IC_{50} values. Variable parameters of the GA were generation size, mutation rate and cross-over strategy. In order to investigate the influence of these parameters on the optimization behavior, each parameter set was used in 100 parallel runs and the averages of the various results were calculated. Population sizes of 5, 10, 20 or 80 and various mutation rates were investigated. Cross-overs that allow cutting between strings that encode for starting materials **A**, **B** and **C** (e.g. 0000101 **/** 1010 0010), which is the only option for real valued encoding, or crossovers that also allow cutting within these bit strings (e.g. 0000101 10 **/** 10 0010) were used.

All results were compared with a 'random' screening method in which all molecules are tested sequentially until suitable molecules are found. This method is the equivalent to 'high-throughput screening' (HTS) of compound libraries, which was simulated by a repeated random selection of new products from the 15,360-product library and building up a list of *N* best compounds as described above.

The influence of the generation size *N* (5, 10, 20 and 80) on the average fitness of the *N* best products at a given generation by applying crossover by cutting between binary educt genes (option c) and a fixed mutation rate of 1% at each bit of the bit string is shown in Figure 4.

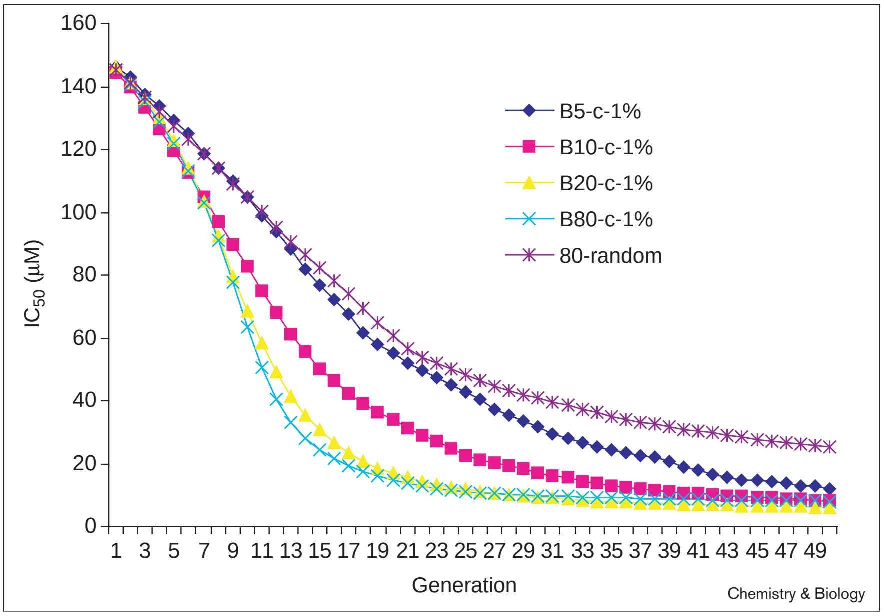
Thus, the GA-driven selection of new members was able to find considerably more active compounds compared with random screening within the first few generations. The slopes of the GA curves are steeper at early generation owing to the 'learning' process and decrease again when many of the limited number of active compounds have been discovered. Finally, all curves arrive at the same point when all compounds of the library have been evaluated, which is the case in our example in generation 768 using a population size of 20. To quantify the benefit of an evolutionary GA strategy as opposed to random screening we may introduce a performance criterion *P*:

$$P = (m_{\mathrm{GA}} / N_{\mathrm{GA}}) / (m_{\mathrm{random}} / N_{\mathrm{random}}) \qquad (1)$$

where m_{GA} and m_{random} are the slopes of the performance curves for the GA and random selection (HTS screening) normalized with the respective generation sizes. *P* represents the average activity increase that was achieved by an individual in one evolutionary cycle using the given GA parameters. The maximum value $P_{\max}$ for generation sizes of 5, 10, 20 and 80 and the parameters shown in Figure 4 was 23.6, 14.1, 10.3 and 3.1, respectively. In other words, small populations 'learn more' per individual, whereas larger populations 'learn faster' as shown by $P_{\max} \times N$, that is, 118, 141, 206 and 248, respectively. This measure therefore provides an easy tool to judge on the efficiency of applying GAs or related evolutionary methods.

Figure 5 shows the influence of the mutation rate and cross-over strategy on the average activity of *N* best products using a generation size of 20. Cutting the strings in a chemically meaningful way only between starting materials (option c) shows a slightly better performance compared with allowing a DNA-like crossover at any bit (option n); however, the latter strategy will result in children with higher diversity as starting materials may be used that were not part of the parent products. Increasing the mutation

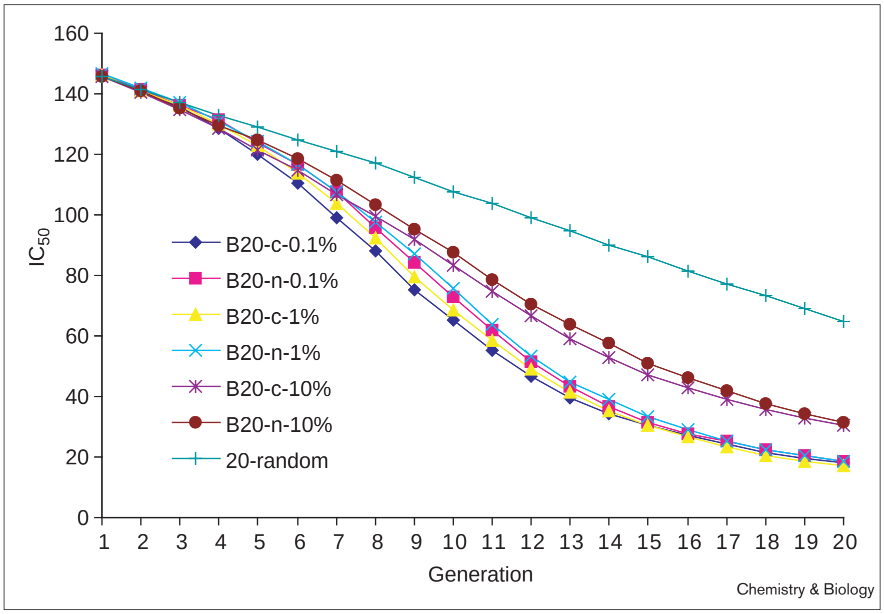
Figure 4



Chemistry & Biology

The influence of the generation size *N* on the average fitness of the *N* best parents at a given generation. *N* was set to 5, 10, 20 or 80 using a cross-over by cutting between starting material genes (option c) and a fixed mutation rate of 1%. The random selection of 80 new products, as opposed to the GA-driven selection, is shown by the curve '80-random'.

Figure 5



Influence of the mutation rate and cross-over strategy on the average fitness of the 20 best parents at a given generation. The mutation rate was set to 0.1, 1 or 10% at each bit of the bit string. Crossover is between starting materials genes only (option c) or also within genes (option n). Comparison with random selection is given by the curve 20-random.

rate from 0.1% to 10% lowers the performance of the GA by 'destroying' the acquired knowledge about fit strings that is implicitly hidden in the pattern of the string. The GA parameter set that performed best used binary encoding, a 1% mutation rate and cutting between starting materials only (strategy B20-c-1%, Figure 5).

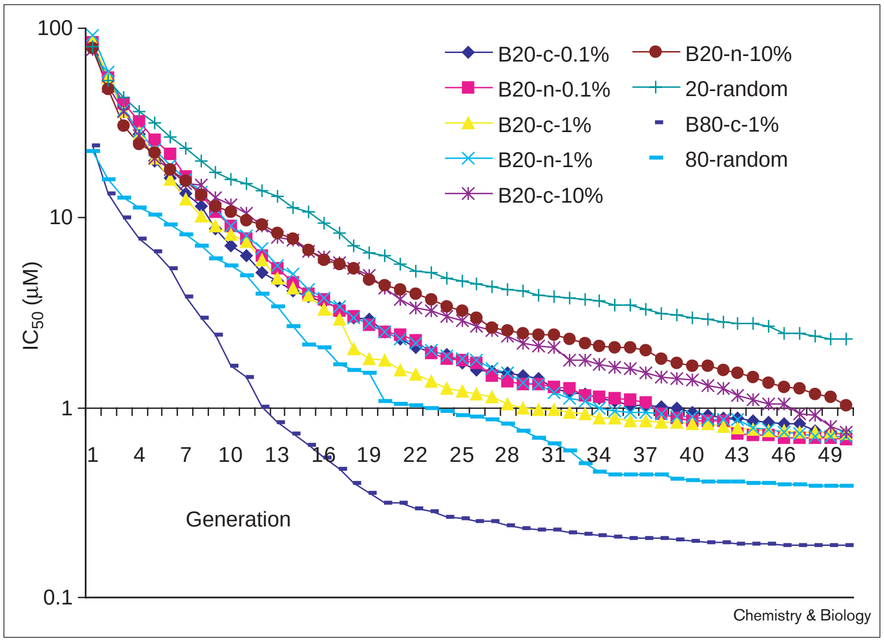
Another parameter of evolutionary optimization is the time taken for the best solutions to be found. Figure 6 shows the activity of the most active product found during the course of a given GA by using various parameter sets and averaging each set over 100 parallel runs. Thus, strategies B20-c-1% and B80-c-1% discover products with an activity below 1 μM with a probability of 90% in generation 30 and 13, respectively. This corresponds to the synthesis of 600 compounds for $N = 20$ and 1040 for $N = 80$. Again, the choice of optimal parameters in real evolutionary experiments will also depend on how difficult it is to synthesize the corresponding products.

The average number of products needed to find one of these nine products by random screening is 15.360/9 = 1.707, therefore, the performance increase provided by the GA in finding the best products is 284% (1707/600) and 164% (1707/1040) for B20-c-1% and B80-c-1%, respectively — higher when compared with random screening. Again, the performance of the GA is better per individual when using smaller generation sizes and worse when using a high mutation rate.

Genetic algorithms learn in an implicit, heuristic manner how to arrive at solutions with desired properties, thereby mimicking evolution in Nature. Figure 7 shows the activity of the N selected new children at each generation for different GA parameters. The activity of randomly selected new compounds (20-random selection) always remains the same on average; however, those for the GA-driven selection increased at the beginning of the evolution process. The respective curves may be regarded as a second and alternative measure for the implicit 'learning' process during a given GA. Whereas the slopes are rather the same for different crossover strategies using a generation size of 20, a marked decrease in learning is seen for the higher mutation rate of 10% in Figure 7. Using a mutation rate of 100% is equivalent to random screening (data not shown).

After reaching a maximum, the learning decreases, owing to the increasing depletion of active products from our search space. The dependence on the generation size gives a similar picture as shown in Figure 7. The shapes of the respective curves are rather similar, but the slope and generation number of the average activity maximum provide a tool to assess the learning process during a GA-driven evolution. In summary, the B20-c-1% strategy was found to be optimal with respect to the number of used individuals, the slope and the achieved maximum average activity increase of the children population at generation 18. This strategy therefore served as a benchmark for comparison with other optimization strategies. Thus, we used real-valued encoding with decimal numbers (option D) and different mutation rates (Figure 8). Essentially, the real-valued encoding algorithm behaves similarly when

Figure 6

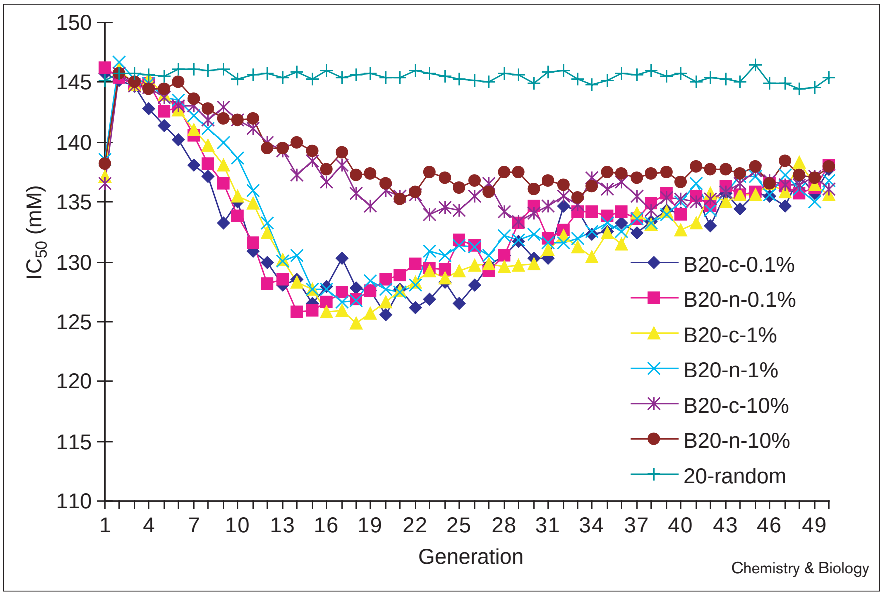


The activity of best reaction product found by the GA during the course of evolution depending on the mutation rate, cross-over and generation size. The results are displayed as averages from 100 parallel runs for each GA parameter set and compared with results from random selection (20- and 80-random).

compared with binary encoding; however, higher mutation rates are needed during the GA run and 20% was found as the best parameter.

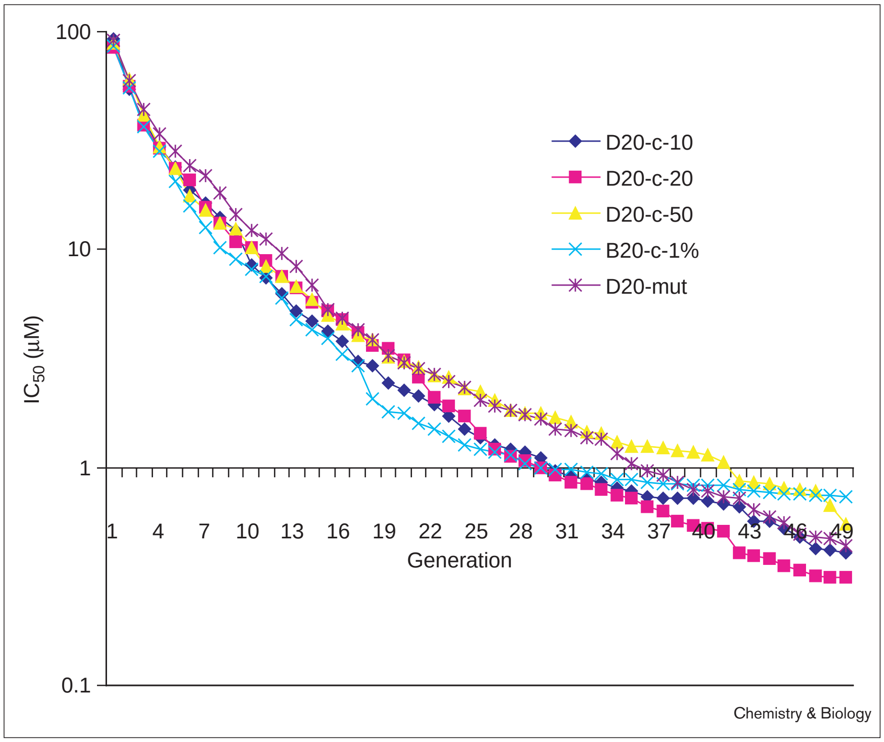
An alternative strategy to our GA-based approach can be implemented by using only mutations and avoiding cross-over. Such algorithms are known as 'hill climbing' or 'simulated annealing' algorithms in which a compound will be selected for further mutations if its activity improves within a certain activity range (in our case the activity range of the population of the *N* best compounds). This algorithm (see Figure 8, strategy D20-mut) is only of advantage if some of the best compounds have been discovered already. Here, the systematic variation around the best compounds yields faster more active compounds than the evolutionary GA approach.

Figure 7



The average activity of the selected 20 new children at each new generation for different mutation rates and cross-over strategies.

Figure 8



The activity of best reaction product found by the GA during the course of evolution depending on decimal (option D) and binary encoding (option B) strategies. Cross-over was used between starting materials (option c), whereas strategy D20-mut uses mutations only. The results are displayed as averages from 100 parallel runs for each GA parameter set.

Discussion

In summary, an evolutionary strategy for screening compound libraries using a GA provides a valuable strategy to discover compounds with desired properties without the need to synthesize and test all possible compounds. The performance of this approach can be optimized by choosing appropriate parameters. A GA is especially powerful when dealing with very large combinatorial libraries, and/or when only a small number of compounds compared with the library size can be synthesized or tested. In our current example, neither the search space nor the three variable parameters **A**, **B** and **C** can be considered as large. We therefore expect that for more relevant problems — using more variables and larger compound libraries — the increase in efficiency will be even better than observed in our case.

We observed that two contradictory factors influence performance. On the one hand, the optimization gradient or 'learning ability' per individual increases by using smaller generation sizes. On the other hand, the best solutions are found faster when using larger generation sizes. The optimum parameters for a given combinatorial optimization will depend on the complexity and 'smoothness' of the search space [5,10] and on the number of compounds that can be made and tested simultaneously. This finding was recently confirmed by Gobbi and Poppinger [11] also for general, noncombinatorial libraries, by using a GA and chromosomes for small molecules that were based on topological descriptors.

Beside GAs there are several other heuristic methods that use feedback-driven algorithms, such as simulated annealing, neural networks and hybrid versions with GAs. Successful applications towards the discovery of biologically active molecules have been published (see [12] for a recent review).

Significance

We believe that a medicinal chemist uses principles that are similar to evolutionary algorithms, although maybe not explicitly or consciously, during the course of a compound optimization. The benefits of a genetic-algorithm-driven small-molecule optimization approach as presented in this current work are twofold: first, it can be part of a fully automated system; and second, it might be especially useful when dealing with very complex, multidimensional search spaces. General parameters have been determined that allow a facile implementation of artificial evolutionary chemistry.

Materials and methods

General library synthesis

12 Amines, 80 aldehydes and 16 isocyanides were prepared as 0.1 M solutions. An aliquot (5 μl) of each component was dispensed into 384-well microtiter plates in a combinatorial fashion to give 15,360 individual reaction products. The plates were sealed and allowed to

react for 24 h, after which the solvent was evaporated at room temperature for 1.5 h. Dry, crude reaction products were then dissolved in dimethylsulfoxide to give 10 mM solutions of the products (assuming a 100% yield of the reaction to expected products). Small aliquots were taken and subjected to mass analysis. The remainder was used in the biological assays for thrombin inhibition.

Biological thrombin assay

The thrombin biological enzyme inhibition assay used 100, 10, 1, 0.1 and 0 μM concentrations for each of the 15.360 reaction products in 384-well plates. HNPT buffer (Hepes 100 mM, NaCl 140 mM, Peg 6000 0.1%, Tween 80 0.02%) at a pH of 7.8 was used. The assay volumes were 15 μl buffer, 5 μl inhibitor, 20 μl thrombin solution (1.25 nM) and 20 μl pyro-Glu–Pro–Arg-pNA (0.5 mM) solution as the chromogenic substrate was used. The incubation time of the enzyme with the inhibitor was 10 min. The conversion of the substrate was measured at 405 nm in a kinetic mode over 10 min. Based on the slopes of the kinetic substrate conversion with four different inhibitor concentrations, 15,360 IC_{50} values were calculated.

*Thrombin inhibitor (**3**)*

A solution of 0.8 mmol of m-aminobenzamidine dihydrochloride and 0.8 mmol triethylamine in 20 ml tetrahydrofurane was added to 1 mmol of o-benzyloxybenzaldehyde and 1.5 ml water to get a clear solution. After 1 h stirring at ambient temperature to form the imine, 1 mmol diphenylmethyl isocyanide was added. The reaction mixture was stirred overnight. After removing the solvent under reduced pressure the crude product was purified by preparative high-performance liquid chromatography (HPLC) using a methanol/water/0.5% acetic acid gradient. Evaporation of the solvent gave **3** as a white powder in 18% yield; ms (ESI/TOF) for $(C_{35}H_{32}N_4O_2+H)^+$ m/z=541.2; 1H-NMR (DMSO-d6): d = 8.85 (d, 1H, NH), 7.44-7.39 (m, 4H, C(NH)(NH2), 7.3-7.18 (m, 16H, Ph), 7.13-7.11 (m, 3H, Ph), 7.03 (d, 1H, Ph), 6.9-6.85 (m, 3H, Ph), 6.58 (d, 1H, NH), 6.10(d, 1H, CH), 5.71 (d, 1H, CH), 5.19 (s, 2H, CH2), 1.72 (s, 3H, acetat).

References

1. Ugi, I. & Steinbrückner, C. (1961). Reactions of isonitriles and carbonyl compounds, amines and azidic acid. *Chem. Ber.* **94**, 734-742.
2. Ballow, M. & Schaffer, F.M. (1994). Molecular genetics of immunoglobulin genes and the generation of antibody diversity. *Clin. Allergy Immunol.* **3**, 3-40.
3. Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung Technischer Systeme nach Prinzipien der Biologischen Evolution [Evolution Strategies: Optimization of Technical Systems with Principles of Biological Evolution].* Frommann-Holzboog, Stuttgart.
4. Holland, J.H. (1975). *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, Ann Arbor, MI.
5. Forrest M. & Mitchell, M. (1996). Relative building-block fitness and the building-block hypothesis. In *Foundations of Genetic Algorithms 2.* (Whitley, D. ed.), pp. 109-126, Morgan Kaufmann, San Mateo, CA.
6. Weber, L., Wallbaum, S., Broger, C. & Gubernator, K. (1995). A genetic algorithm optimizing the biological activity of combinatorial compound libraries. *Angew. Chem. Int. Ed. Engl.* **107**, 2453-2454.
7. Yokobayashi, Y., Ikebukuro, K., McNiven, S. & Karube I. (1996). Directed evolution of trypsin inhibiting peptides using a genetic algorithm. *J. Chem. Soc., Perkin Trans. I* **20**, 2435-2437.
8. Singh, J., *et al.*, & Treasurywala, A.M. (1996). Application of genetic algorithms to combinatorial synthesis: a computational approach to lead identification and lead optimization. *J. Am. Chem. Soc.* **118,** 1669-1676.
9. Gröbke, K., Mehlin, F. & Weber, L. (1998). Synthesis of Imidazo[1,2-a] annulated pyridines, pyrazines and pyrimidines by a novel three-component condensation. *SYNLETT* **6**, 661-663.
10. Weber L. (1998). Applications of genetic algorithms in molecular diversity. *Curr. Opin. Chem. Biol.* **2**, 381-385.
11. Gobbi, A. & Poppinger, D. (1998). Genetic optimization of combinatorial libraries. *Biotechnol. Bioeng.* **61,** 47-54.
12. Dean, P.M. & Lewis, R.A. (1999). *Molecular Diversity in Drug Design*. Kluwer Academic Publishers, Dordrecht.