Identification of Hits and Lead Structure Candidates with Limited Resources by Adaptive Optimization

Andreas Schüller and Gisbert Schneider*

Institute of Organic Chemistry and Chemical Biology, Johann Wolfgang Goethe-University, Siesmayerstrasse 70, 60323 Frankfurt, Germany

Received April 8, 2008

Three stochastic optimization algorithms (Simulated Annealing (SA), Evolution Strategy (ES), and Particle Swarm Optimization (PSO)) and a Random Search were assessed for their ability to generate small activity-enriched subsets of molecular compound libraries. The optimization algorithms were employed to perform an "intelligent" iterative sampling of library molecules avoiding the biological testing of the full library. This study was performed to find a suitable optimization algorithm along with suitable parametrization. Particularly, the optimal number of iterations and population size were of interest. Optimizations were performed with limited resources as the maximal number of compound evaluations was restricted to 300. Results show that all three optimization algorithms are able to produce comparably good results, clearly outperforming a Random Search. While ES was able to come up with good solutions after a few optimization cycles, SA favored high numbers of iterations and was therefore less suited for library design. We introduce PSOs as an alternative approach to focused library design. PSO was able to produce high quality solutions while exhibiting marked autoadaptivity. Its implicit step size control makes it a straightforward out-of-the-box optimization algorithm. We further demonstrate that a nearest neighbor algorithm can successfully be applied to map from continuous search space to discrete chemical space.

INTRODUCTION

The identification of new lead candidates is a crucial task in the early phase of drug discovery. The general goal is to select a small number of compounds with desired properties (*e.g.* bioactivity against a drug target) from the plethora of physically or hypothetically available screening compounds.[1] Chemical space is vast—the number of synthetically accessible organic molecules has been estimated to be in the range of $10^{60}-10^{100}$.[2–4] It is evident that exhaustive screening of such a large number of substances is by no means possible. Advances in high-throughput screening (HTS) and parallel synthesis since the early 1990s have provided a valuable tool for standard pharmaceutical research,[5] and large compound libraries can be synthesized in a combinatorial fashion and screened with help of robotics.[6]

HTS campaigns demand a considerable financial effort and do not always yield many validated hits.[7–9] Alternative approaches employ low-throughput screening methods and concentrate on "cherry-picking" of selected compounds—often only tens to hundreds—with predicted desired activity.[1,10] Among those are computational techniques for virtual screening and automated molecular docking,[11] pharmacophore searching,[12] QSAR methods,[13] and *de novo* design.[14] A complementary approach is adaptive focused library design.[8,15] Here, iterative search strategies are employed to perform "smart" sampling of screening compounds by help of stochastic optimization algorithms.[16] The result of adaptive focused library design is a small molecular library enriched with desired compounds that are focused toward a specific biological target.[8] In an iterative fashion candidates are selected from the pool of available screening compounds, and their quality is evaluated. The quality values are fed back into the algorithm, and a new generation of compounds is suggested by variation of successful candidates. Quality values are determined by a "fitness function" and range from similarity indices to experimentally measured binding constants.[14] Adaptive focused library optimization thus operates with repetitive cycles of variation and evaluation. These correspond to iterative synthesize-and-test rounds in the laboratory. Summarizing, the three integral ingredients of adaptive library optimization are i) a pool (repository) of screening compounds, ii) an optimization algorithm, and iii) a fitness function.

Evolutionary Algorithms (EA) and among those especially the Genetic Algorithms (GA)[17] have been commonly employed in molecular library design.[15,18–27] EAs are based on simplistic concepts of Darwinian evolution. Populations of offspring are generated by the genetic operators "mutation" and "cross-over", and the principle of "natural selection" is applied to choose parents for reproduction. Optimization of combinatorial libraries is the most common example in literature,[16] although vendor screening libraries may serve equally well as compound repository.[28]

Most of these studies used some computational way to evaluate the fitness of selected compounds.[14] The need for optimization of multiple targets was taken up and realized with weighted sums,[19,29] desirability scores,[24] and Pareto ranking.[20,21] Few studies operated directly on measured biological activity data as fitness function. Such reports

* Corresponding author fax: (+) 49 69 798 24880; e-mail: g.schneider@chemie.uni-frankfurt.de. Corresponding author address: Chair for Chem- and Bioinformatics, Institute of Organic Chemistry and Chemical Biology, Goethe-University, Siesmayerstrasse 70, D-60323 Frankfurt am Main, Germany.

10.1021/ci8001205 CCC: $40.75 © 2008 American Chemical Society
Published on Web 07/01/2008

Table 1. Number of Active Compounds in the Ugi Data Set (First Row) and Their IC_{50} Ranges for Each Serine Protease (Second Row)[a]

	chymotrypsin	factor Xa	trypsin	tryptase	uPA
no. of actives	34 (0.2%)	1703 (10.8%)	305 (1.9%)	4723 (29.8%)	1390 (8.8%)
median IC_{50}	37.2 (4.1−72.0)	47.9 (33.9−67.6)	67.4 (50.0−84.1)	38.0 (27.5−52.5)	67.1 (52.4−85.4)
mean top 20 IC_{50}	13.7	2.3	17.7	0.6	10.4

[a] The number ranges in brackets give the lower and the upper quartile of the IC_{50}, respectively. The fourth row provides the mean IC_{50} of the top 20 most active compounds for each enzyme. These numbers represent the global optimum of a 20-member focused library.

include the design of thrombin inhibitors,[22,26] peptidic stromelysin substrates,[25] peptidic trypsin inhibitors,[27] compounds against various biological targets (antibacterial, antifungal, anti-Alzheimer's disease, and GPCR antagonists),[28] antibacterial compounds,[24] and, closely related, the *de novo* design of cannabinoid receptor ligands.[30]

Another optimization algorithm that has been successfully applied to library design is Simulated Annealing (SA).[23,31,32] SA is a global optimization algorithm grounded on the simulated cooling of a physical system.[33–35] Other proprietary iterative search strategies that are not based on a common optimization algorithm have been reported as well.[28,36,37]

Here, we also use the Evolution Strategy (ES)[38–40] for iterative focused library design. Previously, ESs were successfully applied to ligand-based *de novo* design.[14,41–43] ESs differ from GAs in several ways: Whereas GAs are more focused on working with the genotype (that is, for instance, the binary-encoded representation of the starting materials of a combinatorial reaction), ESs pay more attention to the direct effects on the phenotype. A further difference is that GAs preferably use cross-over operators, while ESs primarily (or even exclusively) work with mutation operators. In GAs, solutions are generally represented in the form of discrete-valued "chromosomes" which makes them intuitively suited for combinatorial library design. In contrast, ESs operate on real-valued data vectors as genotypes. This form of representation is especially suited for precompiled compound collections which are encoded by molecular descriptors. The present study consequently employs two such precompiled data sets, of which one is a fully enumerated combinatorial library (see "Materials and Methods" for details).

Our aim was to find a suitable optimization strategy with proper parametrization for adaptive focused library optimization in preparation for a prospective low-throughput screening experiment. We compared the different behavior of three well-established optimization algorithms: SA, ES, and Particle Swarm Optimization (PSO). PSO is a biologically inspired population-based optimization technique grounded on an abstraction of biological swarms.[44,45] They have been successfully applied to a number of problems in cheminformatics.[46–51] To our knowledge, this is the first time Particle Swarm Optimization is used for molecular library design. The design of the experiments in this study was guided by practical considerations with iterative low-throughput screening in mind. We simulated limited laboratory resources by restricting the number of compounds that may be tested to 300. We also assumed to have very little information about the structure−activity relationship of the biological target and thus did not perform any prefiltering or feature selection of the screening data sets. With respect to the applicability in a prospective study, fitness values were not computed. Rather, the fitness function returned biological activity data that were determined beforehand in experimental assays.[22,52] The goal of all experiments was the design of activity-enriched focused subsets of the full screening libraries.

MATERIALS AND METHODS

Data Sets and Molecular Descriptors. Two different molecular data sets were employed for the present study. The first data set, referred to as the *Ugi data set*, is a combinatorial library of 15,840 products of a three-component Ugi-type reaction synthesized from 15 amines, 44 aldehydes, and 24 isonitriles.[22,52–54] All Ugi products had been tested for inhibition of five serine proteases (Table 1):[52] chymotrypsin, factor Xa, trypsin, tryptase, and urokinase-type plasminogen activator (uPA). The inhibitory concentration (IC_{50}), giving a decrease in the substrate cleavage rate of 50% compared to the uninhibited reaction, was estimated by using an *in vitro* assay for each individual enzyme. Molecules with $IC_{50} \geq 100$ μM were considered inactive. The *Ugi data set* is available for download from our Web site at http://www.modlab.de/.

As a second data set, we used the Collection of Bioactive Reference Analogues (COBRA, version 6.1).[55] COBRA contains 8311 biologically active reference compounds that were manually compiled from selected scientific journals and various textbooks. The collection is annotated with target receptor information and activity data. The version of COBRA used in this work contained binary activity data for each target receptor. 228 (2.7%) reference molecules represented active inhibitors against factor Xa, the relevant target receptor in COBRA for this study.

For all compounds, hydrogen atoms were added and salts were removed with the software tool CLIFF.[56] Charges were neutralized, and a single 3D conformation per molecule was calculated with the structure generator CORINA.[57,58] Three types of molecular descriptor sets were calculated:

1) *MOE 2D*: 146 2D descriptors were computed with the Molecular Operating Environment (MOE, version 2004.03).[59] The descriptors, which were calculated from the connection table of the compounds only, can be grouped into 'physical properties' (charge, logP, weight, . . .), 'subdivided surface areas' (logP and molar refractivity surface properties), 'atom counts and bond counts', 'Kier & Hall connectivity and Kappa shape indices', 'adjacency and distance matrix descriptors', 'pharmacophore feature descriptors', and 'partial charge descriptors'. Partial charges were calculated beforehand in MOE using the MMFF94x force field.[59] All descriptors were autoscaled (mean centered and scaled to unit variance) to yield the final set of descriptors (abbreviated *MOE 2D* from here on).

2) *MOE 2D PCA*: In order to reduce dimensionality and redundancy, principle component analysis (PCA) was per-

formed on the *MOE 2D* set within MOE. All principle components with an eigenvalue greater than or equal to unity were taken as the resulting second set of descriptors (abbr.: *MOE 2D PCA*).[60] 16 principle components explaining 93.4% of the total variance were selected as descriptors for the *Ugi data set*, and 22 principle components explaining 90.1% of the total variance were chosen for COBRA.

3) *CATS*: The third set of descriptors consisted of the 150-dimensional topological CATS descriptor.[61] CATS is an alignment-free pharmacophore representation of a molecule considering the pharmacophoric types of hydrogen-bond donor (D), hydrogen-bond acceptor (A), lipophilic (L), positively charged or ionizable (P), and negatively charged or ionizable (N). The computational protocol of CATS assigns potential pharmacophoric points (PPPs) to atoms and functional groups of a molecule and calculates the mutual topological shortest path quantified by the number of bonds between all PPPs. A mapping function combines the information of the topological distance and the PPP assignments to obtain a pharmacophoric correlation vector (CV). The mapping function counts the frequencies of all possible PPP pair distances and sums them in ten bins covering the distances of 0 to 9 bonds. The 15 pairing combinations of PPPs (DD, DA, DP, DN, DL, AA, AP, AN, AL, PP, PN, PL, NN, NL, LL) and 10 different distances (bins) result in a 150-dimensional CV. The final step of the CATS descriptor calculation is the scaling of the vector.[62] Like the *MOE 2D* and the *MOE 2D PCA* descriptor sets, CATS descriptors were autoscaled to ensure comparable value ranges.

Optimization Algorithms. Optimization algorithms are problem solvers for mathematically defined minimization or maximization problems.[63] They seek to find a set of input variables $\mathbf{x}_{\mathbf{opt}} = (x_1, x_2,..., x_{n\text{-}1}, x_n)$ for which a function f($\mathbf{x}$) is optimal. An instance of $\mathbf{x}$ can be seen as the position vector of a solution candidate to f($\mathbf{x}$) in an *n*-dimensional search space, where f($\mathbf{x}$) is called the objective function, sometimes also referred to as "cost function", "scoring function" or—especially in the context of evolutionary algorithms—"fitness function". The algorithms employed in this study belong to the group of stochastic optimization algorithms whose common characteristic is the incorporation of some random element (e.g., a pseudorandom number) during optimization.[64] Their proposed solutions are generally a heuristic, that is, a good approximation of the global optimum. The four search algorithms compared in this study are i) Random Search, ii) Simulated Annealing,[33–35] iii) Evolution Strategy,[38–40] and iv) Particle Swarm Optimization.[44,45] Pseudocode of our implementations of the optimization algorithms is available as Supporting Information.

Random Search. Random search is used throughout this work as a reference for comparison to elucidate the question as to how good an expected result would be if it was determined by chance only. During Random Search a specified amount of solutions are drawn randomly from a discrete set of solutions (here: molecules of a screening library), their objective function value is evaluated, and the solutions are sorted accordingly. Any other search strategy should in any case perform better than a Random Search.

Simulated Annealing. Simulated Annealing (SA) is a general purpose global optimization algorithm which operates by simulating the cooling of a physical system whose possible energies correspond to the values of the objective function being minimized.[33–35] As physical systems assume only low energy states when the temperature is lowered to absolute zero, so should an optimization strategy following this analogy. Central to SA is the Metropolis acceptance criterion

$$R < e^{-\Delta E/T} \tag{1}$$

where R is a pseudorandom number drawn from an uniform distribution in [0,1], ΔE is the difference in energy or, more generally speaking, the difference in quality of two neighboring solutions, and T is a system-specific temperature factor. If eq 1 holds true, a new solution candidate is accepted even if its quality is worse than the original solution. Accounting for the upper detection limit of activity assays (here: $\mathrm{IC}_{50} =$ 100 μM), we extended the Metropolis acceptance criterion to reject inactive compounds even if they passed the Metropolis equation. This modification prohibits a jump from an "active" solution candidate to an "inactive" one. In our implementation an exponential cooling scheme

$$T(t) = T_0\alpha^t \tag{2}$$

was applied, where $T(t)$ refers to the temperature at time step t, T_0 denotes the initial temperature, and α is a cooling factor. The value of T_0 was chosen in a way that a half-maximal change in a solution's quality resulted in a Metropolis acceptance probability of approximately 50% in the initial state. The cooling rate α was chosen to reach a final temperature of 1 in the last iteration of optimization. Neighboring solutions were created from the current solution by applying an Evolution Strategy-like mutation operator with fixed step size (see below for details). As a variation of this scheme, we implemented a multistart algorithm that performs several SA runs in parallel and archives the best solutions of all runs.

Evolution Strategy. Evolution Strategies (ES) are heuristic, population-based optimization algorithms developed in the late 1960s by Rechenberg.[38–40] They belong to the class of evolutionary algorithms, and their mode of operation is inspired by concepts of biological evolution. Neighboring solutions, called offspring individuals, are created by mutation and cross-over operators from parent individuals which themselves evolve by help of simulated natural selection from a population of individuals. A straightforward (μ, λ) ES selects μ parents which breed λ offspring in each generation. Our implementation uses only a mutation operator to breed offspring. Individuals are created by adding (0,σ)-normally distributed pseudorandom numbers to each component of the parent's position vector, where σ denotes the standard deviation of a Gaussian distribution. As a consequence offspring individuals are approximately ($\sigma\sqrt{n}$, $\sigma/\sqrt{2}$)-normally distributed around their parents. Or in other words, new offspring are placed in the shell of a hypersphere with radius $r = \sigma \cdot \sqrt{n}$ and width $w = \sigma/\sqrt{2}$ around their parent, where n denotes the dimension of the search space.[39]

Essentially, σ represents a step size parameter of ESs which controls increment and spreading of new offspring. We implemented this parameter as an autoadaptive value in an attempt to escape local minima. Offspring inherit their parent's step size multiplied by 1.3 or 1/1.3 as decided by a 50% chance.[39] ESs operate with high selective pressure as only the μ fittest individuals are selected as parents. Accounting for the upper detection limit of activity assays, our implementation comes with a slight modification in the parent selection step. The actual value of μ is adaptive to the number of actives in a population (eq 3):

$$\mu' = \left\{ \begin{array}{ccc} \mu & \textit{if} & \#actives = 0 \\ \#actives & \textit{if} & 0 < \#actives\# < \mu \\ \mu & \textit{if} & \#actives \geq \mu \end{array} \right\} \qquad (3)$$

where μ' denotes the resulting number of parents, μ denotes the maximal number of parents, and *#actives* gives the number of active individuals (compounds) in a population. This modification results in an increased selection pressure and prevents selection of inactive individuals as parents.

ESs operate on continuous search spaces, but descriptor-encoded molecular libraries represent sparsely populated, discrete search spaces. Chemical space is discrete as only those positions are populated that actually represent "real" molecules. The variation step of optimization algorithms, however, yields position vectors pointing at arbitrary positions of chemical space, and it is almost certain not to encounter a "real" molecule at those arbitrary positions. We therefore decided to perform a nearest neighbor search for each offspring individual. As a consequence, the resulting position vectors of the mutation process are mapped to nearby located positions in discrete space, representing actual molecules. The nearest neighbor was determined by calculating the Manhattan distance of the position vector to the descriptor vectors of each database molecule, sorting the resulting list by ascending order and returning the first list entry (with minimal distance) as nearest neighbor. Note, that although molecular screening libraries represent a discrete search space, solution candidates (here: molecules) are still represented by real-valued descriptor vectors.

Particle Swarm Optimization. Particle Swarm Optimization (PSO) is a biologically inspired population-based optimization technique introduced 1995 by Kennedy and Eberhart.[44,45] PSO is grounded on an abstraction of biological swarms, such as bird flocks or fish swarms, and imitates collective movements and social interactions. As particles (that is, a swarm's objective function solution candidates) move through search space they retain two memories: i) the individual, cognitive memory of their personal best position visited so far, and ii) the global, social memory of the overall best position the swarm has encountered so far. Neighboring solutions are created by letting particles "fly" through search space. In addition to a position vector each particle also holds a velocity vector which determines its direction and increment of movement. In our implementation, a constriction-type Particle Swarm was used.[65] A movement operation includes the update of the velocity vector according to eq 4 and the modification of the position vector according to eq 5

$$v_i(t+1) = K \cdot \left(v_i(t) + c \cdot R_1 \cdot \left(m_i^c - x_i(t)\right) + s \cdot R_2 \cdot \left(m_i^s - x_i(t)\right)\right) \qquad (4)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \qquad (5)$$

where $v_i(t)$ denotes the ith component of the velocity vector at time step t, x denotes the position vector, R_1 and R_2 are uniformly distributed random numbers on [0, 1], m^c is the collective memory and m^s is the social memory, and c and s are positive weighting parameters tuning the influence of the cognitive and social memory, respectively. K is the constriction factor assuring adaptive convergence of the swarm defined as

$$K = \left| \frac{2}{2 - \phi - \sqrt{\phi^2 - 4\phi}} \right|, \text{ with } \phi = c + s \qquad (6)$$

Velocity vectors were randomly initialized within a bounding box defined by the minimal and maximal values of each dimension of the search space. To enable particle swarms to efficiently operate with binary objective function values (e.g., binary fitness values "active" and "inactive") we slightly modified the algorithm's memory update condition: Usually a particle's memory is updated when a better solution is found. In our implementation, the memory is updated when a better or equal solution is encountered. An equal solution is rejected when its fitness (IC_{50}) is above some threshold (that is, for inactive molecules). As for ES, we added an additional nearest neighbor search step to map from continuous space to discrete search space.

Each optimization algorithm requires parameters to adapt to the specific problem at hand ("strategy parameters"). The present study was conducted to find a robust and generalizing method for focused library design. Thus, no comprehensive optimization of the search parameters for the various combinations of search spaces, optimization algorithms, and optimization targets was performed. Rather, a general set of parameters was found on basis of the *Ugi data set* (Table 2). The nearest neighbor step in our implementations of SA, ES, and PSO was adjusted so that a candidate solution could never be selected twice. This is equivalent to a global *tabu* rule. The positioning of neighboring solutions was restricted by a bounding box defined by the minimal and maximal values of each dimension of the search space. Due to the presence of inactive molecules which share the same fitness values effective optimization was only possible once an active compound was found by the search algorithm. The random initial population selection was therefore repeated until either at least one active compound was selected or the termination condition held true. Common properties of our implementations of all four search algorithms are as follows: i) one candidate solution represents exactly one molecule, ii) molecules are encoded by descriptor vectors, and iii) the optimization target was to minimize the objective function.

Table 2. Parameter Setting of the Optimization Algorithms

parameter	value
Simulated Annealing	
initial temp T_0	75.0
initial temp (with initial actives)	20.0
final temp T_{end}	1.0
cooling rate α	$(T_{\text{end}}/T_0)^{1/\#\text{iterations}}$
Evolution Strategy	
strategy	comma (with elitism of 1)
number of parents μ	population size/6
maximal step size σ	5.0
Evolution Strategy mutation operator (used by ES and SA)	
step size σ (CATS)	2.0
step size σ (MOE 2D)	1.5
step size σ (MOE 2D PCA)	0.75
Particle Swarm optimization	
cognitive factor c	2.05
social factor s	2.05
maximal velocity	unconstrained
constriction factor	cf. eq 6

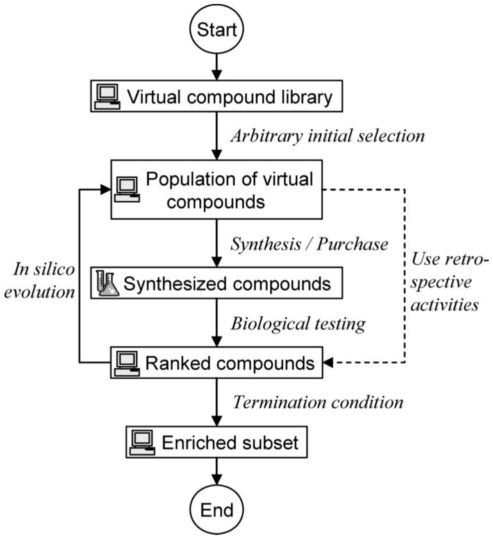

Figure 1. Work flow of adaptive focused library design in computer-assisted low-throughput screening. Steps performed in the computer or the laboratory are marked with a computer and a chemistry icon, respectively. In retrospective experiments conducted for this study activity values for all library molecules were determined beforehand. Hence, the steps of "Synthesis/Purchase" and "Biological testing" were omitted. Instead, activity values were looked up from the list of available retrospective activities (dashed line).

Adaptive Focused Library Optimization. The goal of adaptive library design is to avoid biological testing of a complete screening compound library. Instead, the idea is to navigate in an "intelligent" way through chemical space, "cherry-picking" only molecules with desired properties.[1,10] The setup of a general library optimization study follows this concept (Figure 1): An initial population of molecules is chosen from a molecular library. The selected compounds are synthesized or purchased, and activity is determined experimentally in a laboratory. The measured activity data are fed back into the optimization algorithm which creates a new and hopefully activity-enriched, generation of molecules based on *in silico* evolution. This synthesize-and-test loop continues until a termination criterion is met. The approach is a hybrid, employing the computer as an idea generator for new molecules and making use of actually measured activity data in the step of evaluating the fitness function. New active compounds can potentially be found in each iteration of the procedure by experimental determination of the synthesized molecules in a laboratory.

The present study was performed to find suitable parameters in preparation for a prospective iterative optimization. As a consequence, retrospective experiments were conducted, and all laboratory steps were omitted. Both data sets employed in this study were already annotated by experimentally determined activity data. The steps of "Synthesis/Purchase" and "Biological testing" (Figure 1) were therefore replaced by a single step in which activity values were looked up from the list of available retrospective activities (dashed line in Figure 1). It is important to note the activity data remained hidden to the algorithm and were only made available on request, as in a *pseudo*prospective manner. The main questions to elucidate in this study were as follows:

- Which of the stochastic optimization algorithms is most suitable for focused library design?
- Which population size works best and how many cycles should be performed?

Since we had low-throughput screening with limited financial and personnel resources in mind, as typically seen e.g. in an academic environment, two important constraints further applied to the optimization:

- The maximum number of compound evaluations was restricted.
- A small number of iterations was preferred.

We restricted the maximum number of compound evaluations to 300 (1.9% of the total compounds) in the case of the *Ugi data set* and to 150 (1.8%) in the case of COBRA. The number of 300, respectively 150, refers to the maximal budget of compounds that might be synthesized or purchased and biologically tested in a prospective study. The second constraint has an implication on the temporal aspect of such a campaign. Due to prolonged waiting times in compound synthesis (or purchase/delivery) and biological testing the number of iterations should be kept low in order to retrieve and test new compounds in bulk rather than in minor quantities.

The constrained total number of compound evaluations can be partitioned into varying numbers of population size and generations. The product of population size by number of generations must not be greater than the maximum number of evaluations. To find the best possible combination of the two parameters we performed retrospective optimization runs and compared the results. The 14 different combinations evaluated for the *Ugi data set* were 1 × 300, 2 × 150, 3 × 100, 5 × 60, 10 × 30, 15 × 20, 20 × 15, 30 × 10, 37 × 8, 50 × 6, 60 × 5, 75 × 4, 100 × 3, and 150 × 2, where the first number gives the population size and the second number gives the number of generations. The combinations evaluated for COBRA were 1 × 150, 2 × 75, 3 × 50, 4 × 37, 5 × 30, 7 × 21, 10 × 15, 15 × 10, 21 × 7, 25 × 6, 30 × 5, 37 × 4, 50 × 3, and 75 × 2.

Self-Organizing Maps. Self-Organizing Maps (SOM) belong to the class of unsupervised neural networks and were pioneered by Kohonen in the early 1980s.[66] They can be used to generate low-dimensional, topology preserving projections of high-dimensional data, e.g. representations of chemical compound libraries. SOMs are made up of a single layer of neurons of the same dimension as the input patterns (e.g., descriptor vectors). During the SOM training process the original high-dimensional space is tessellated, resulting in a certain number of data clusters. There are formed as many clusters as are neurons in the SOM. To map similar input patterns onto adjacent regions in output space a topology is introduced to the SOM neuron layer. For SOM training an input pattern is compared to all SOM neurons, and the one most similar neuron is found. Then an updating procedure adapts the vector elements of the winner neuron and its topological neighbors toward the input pattern. As a result, topologically adjacent neurons correspond to adjacent input patterns. Here, we employed the software tool MOLMAP to create two-dimensional maps of our high-dimensional descriptor spaces.[67] 20 × 20 SOMs with a toroidal topology were trained for 1,000,000 iterations. The torus ("doughnut" shape) represents an infinite two-dimensional space. The upper and lower side as well as the left and right side of the maps are connected with each other. MOLMAP uses a Gaussian neighborhood function to define neighborhood behavior of neurons, and Manhattan metric was employed for SOM training. All molecules of the search space, that is either the entire *Ugi data set* (15,840 compounds) or the entire COBRA (8311 compounds), were used for SOM training.

RESULTS AND DISCUSSION

Comparative Parameter Screening. The present study was conducted in order to determine the most suitable adaptive optimization algorithm along with the most fitting parameters for population size and number of generations under the constraint of a limited number of available tests. Three optimization algorithms i) Simulated Annealing (SA), ii) Evolution Strategy (ES), and iii) Particle Swarm Optimization (PSO) were compared against Random Search as described in "Materials and Methods". Three different descriptor representations of the *Ugi data set* (15,840 Ugi-type reaction products), namely 146 MOE 2D descriptors, 16 MOE 2D principal components, and the 150-dimensional CATS descriptor were compared. The budget of compound evaluations was limited to 300, and 14 different combinations of population size and number of generations were screened. The optimization target was to minimize the IC_{50} of the selected library molecules. The quality of an optimization run was assessed as the average IC_{50} value of the top 20 most active molecules found in all generations of one run. The optimization target was thus to generate a 20-member focused library with minimal average IC_{50} value.

The results for the overall best combination of population size and number of generations are given in Tables 3 (CATS), 4 (MOE 2D), and 5 (MOE 2D PCA). The tables show results for optimization runs with two different kinds of initial population selection: i) random selection and ii) inclusion of four active compounds within the initial selection. We will discuss results for the former selection type first.

With the exception of chymotrypsin, all three optimization algorithms clearly outperform a Random Search in finding serine protease inhibitors. The poor result for finding chymotrypsin inhibitors is readily explained with their low abundance in the *Ugi data set*. Only 34 of 15,840 total molecules (0.2%) exhibit activity against chymotrypsin. The underlying fitness landscape might be imagined as a golf course: There are vast plateaus devoid of any information (inactive molecules) with only a very few golf holes (active molecules).[35] The fundamental principle of all optimization algorithms is that of strong causality: small changes in the search space should result in only small changes of a solution's quality.[39,68–70] If the local neighborhood around an optimum does not exhibit any guidance toward this optimum, e.g. in the form of a gradient, any optimization algorithm will at best work as good as a Random Search. As optimizations of trypsin inhibitors (1.9% actives in the *Ugi data set*) succeeded, the lower frequency limit of active compounds in a data set might be somewhere between 0.2% and 1.9%.

There is no clear "winner" among the search algorithms tested. All optimization algorithms (excluding Random Search) were able to produce good results. With appropriate parameters all three optimization algorithms performed similarly. The average difference in quality (ΔIC_{50}) of the optimized 20-member focused libraries (Tables 3, 4, and 5) was 3.3 ± 3.9 μM and never exceeded 20 μM. ES and PSO produced solutions with similar quality, and SA performed slightly worse. With one exception (tryptase inhibitors) optimizations in the CATS descriptor space yielded best results. Interestingly, the 146 MOE 2D descriptors and the 16 MOE 2D principle components performed comparably well. Only 16 principle components (~11% of 146) were sufficient to produce results as good as the complete descriptor set. The fact of reduced dimensionality did considerably speed up the optimization runs (data not shown).

In preliminary optimization experiments, we quickly realized that the abundance of actives in the data sets did not only influence the minimal IC_{50} values of the optimized subsets but also the robustness of the solutions. In a prospective study, one would perform only one (or a few) optimization run but certainly not 100 repetitions as we did to obtain the statistical data. Large error spans (expressed as the difference of upper and lower quartiles, Tables 3, 4, and 5) are therefore an indicator of weak robustness. Optimizations for trypsin inhibitors exhibited the largest error spans (approximately 30 μM) in comparison with the notably smaller error spans of factor Xa, tryptase, and uPA optimizations. Trypsin inhibitors do also present the class of lowest abundance of active molecules (except for chymotrypsin, where optimizations completely failed). Starting from preliminary experiments (not shown) we were able to stabilize optimization runs for trypsin inhibitors by repeating the random selection of the initial population until at least one active molecule was found. Still, chances were high to perform badly in a prospective study due to poor robustness.

The probability P to find at least one active inhibitor in an initial population of, for instance, 30 molecules is

$$P = 1 - \prod_{i=0}^{29} \frac{total - i - \#actives}{total - i} \tag{7}$$

where *total* denotes the total number of library compounds and *#actives* denotes the number of actives compounds. According to eq 7 the probability to find at least one active factor Xa inhibitor in an initial population of 30 molecules is 0.97. The probability for trypsin is 0.44 and 0.06 for chymotrypsin. Analogously, optimization results were good for factor Xa, satisfactory but with large errors for trypsin, and as good as random for chymotrypsin. We therefore decided to include four active compounds within the initial population to further stabilize the optimization runs. The four actives were randomly chosen from the 50 most active factor Xa inhibitors, the 20 most active trypsin inhibitors, and the 10 most active chymotrypsin inhibitors. The enzymes factor Xa, trypsin, and chymotrypsin were chosen because of their increasing difficulty for optimization due to their decreasing numbers of actives in the data set. The probabilities of randomly selecting at least four actives in the initial population are much lower: 0.41 (factor Xa), 0.0025 (trypsin), and 4.7×10^{-7} (chymotrypsin; probabilities calculated with help of a hypergeometric probability density function in Matlab, The MathWorks, version 7.04.365 R14 SP2, with the statistics toolbox). The inclusion of initial actives should therefore provide a better starting point for optimization than pure random selection, and the algorithms can spend more time on actually optimizing the IC_{50} than on finding an active compound to start with.

Table 3. Results of a Comparative Study with a Budget of 300 Compounds on the Ugi Data Set Encoded by the CATS Descriptor[a]

	Random Search	Simulated Annealing	Evolution Strategy	Particle Swarm
Chymotrypsin				
min. IC_{50} [μM]	109.4	108.6	108.3	109.0
quartiles	105.0−110.0	104.7−110.0	104.7−110.0	104.6−110.0
# actives	0.0	1.0	1.0	1.0
quartiles	0.0−1.0	0.0−1.0	0.0−2.0	0.0−1.3
pop. size × gen.		10 × 30	2 × 150	5 × 60
Chymotrypsin with Four Initial Actives				
min. IC_{50} [μM]	109.5	96.0	101.0	94.2
quartiles	106.3−110.0	92.2−104.7	95.9−105.0	87.5−105.0
# actives	0.0	3.0	2.0	3.0
quartiles	0.0−1.0	1.0−4.0	1.0−3.0	1.0−5.0
pop. size × gen.		3 × 100	100 × 3	2 × 150
Factor Xa				
min. IC_{50} [μM]	36.1	8.4	8.3	7.5
quartiles	32.6−40.0	6.9−11.0	6.6−10.2	6.6−11.2
# actives	32.0	158.0	164.0	101.5
quartiles	29.0−36.0	129.8−174.3	133.0−177.3	86.8−137.0
pop. size × gen.		1 × 300	20 × 15	50 × 6
Factor Xa with Four Initial Actives				
min. IC_{50} [μM]	36.4	6.6	6.4	6.0
quartiles	32.7−40.5	5.6−8.3	5.2−7.4	5.2−6.6
# actives	32.0	151.5	123.0	127.0
quartiles	28.0−35.0	127.8−168.0	113.8−140.5	118.8−169.3
pop. size × gen.		3 × 100	37 × 8	5 × 60
Trypsin				
min. IC_{50} [μM]	90.4	35.1	30.8	31.5
quartiles	86.8−93.6	27.7−48.5	26.8−41.3	26.2−47.6
# actives	5.0	46.0	52.5	56.0
quartiles	4.0−7.0	30.0−58.3	33.0−66.0	31.0−64.0
pop. size × gen.		1 × 300	15 × 20	2 × 150
Trypsin with Four Initial Actives				
min. IC_{50} [μM]	90.8	26.7	26.8	24.0
quartiles	87.3−93.8	24.1−30.3	24.9−30.0	22.7−27.2
# actives	6.0	67.0	65.5	70.0
quartiles	4.0−7.0	57.8−74.0	62.0−70.3	65.0−75.3
pop. size × gen.		3 × 100	50 × 6	5 × 60
Tryptase				
min. IC_{50} [μM]	18.6	3.9	2.7	2.8
quartiles	17.1−20.2	2.9−5.3	2.2−4.3	2.4−5.1
# actives	89.0	244.0	235.0	262.0
quartiles	84.0−94.0	234.0−250.0	224.8−243.0	254.8−269.0
pop. size × gen.		1 × 300	30 × 10	3 × 100
uPA				
min. IC_{50} [μM]	59.4	21.3	17.4	15.5
quartiles	54.9−64.4	16.7−31.5	16.0−22.5	14.2−19.7
# actives	26.0	118.0	125.0	144.0
quartiles	23.0−30.0	97.8−135.0	115.8−137.0	131.8−160.3
pop. size × gen.		1 × 300	37 × 8	20 × 15

[a] pop. size × gen. denotes the best combination of population size and number of generations as determined by the mean IC_{50} of the top 20 most active compounds of an optimization (min. IC_{50} [μM]). Minimal IC_{50} values are given as the medians of 100 repetitions along with the lower and upper quartiles. The median number of found actives is given with # actives.

In fact, the inclusion of four initial actives led to a noticeable improvement of the minimal IC_{50} values for factor Xa and trypsin. The best strategy for factor Xa yielded a minimal IC_{50} of 5.8 μM (Table 4, ES) which is close to the global optimum of 2.3 μM (Table 1). The error spans for optimized trypsin inhibitor subsets reduced from 30 μM to approximately 8 μM. Only a modest improvement resulted for chymotrypsin inhibitors, the most difficult optimization target in the present study. It should be noted here that the initial temperature for SA experiments had to be reduced from 75 to 20. Otherwise SA would “jump away” from the initial actives due to a high Metropolis acceptance probability (eq 1).

All three compared optimization algorithms preferred different combinations of population size and number of generations (Figures 2 and 3). SA clearly preferred high numbers of iterations and consequently a small population size. Here, population size refers to the number of concurrent initializations of our multistart SA implementation. SA exhibited an approximate linear dependence on the number of iterations which is inherent to its mode of operation. SA is grounded on the thermic equilibration of a physical system being gradually cooled down. The slower the temperature of the system is decreased, the more time there is for thermic equilibration. In terms of optimization parameters, this refers to a high number of iterations resulting in a prolonged

Table 4. Results of a Comparative Study with a Budget of 300 Compounds on the Ugi Data Set Encoded by the MOE 2D Descriptors[a]

	Random Search	Simulated Annealing	Evolution Strategy	Particle Swarm
		Chymotrypsin		
min. IC_{50} [μM]	109.5	108.5	108.2	108.0
quartiles	105.0−110.0	104.6−110.0	104.4−110.0	104.0−110.0
# actives	0.0	1.0	1.0	1.0
quartiles	0.0−1.0	0.0−1.0	0.0−2.0	0.0−3.0
pop. size × gen.		30 × 10	3 × 100	20 × 15
		Chymotrypsin with Four Initial actives		
min. IC_{50} [μM]	109.5	93.9	93.7	90.3
quartiles	106.3−110.0	89.1−99.1	88.7−99.1	83.6−95.4
# actives	0.0	3.0	3.0	4.0
quartiles	0.0−1.0	2.0−4.0	2.0−4.0	3.0−5.0
pop. size × gen.		3 × 100	100 × 3	3 × 100
		Factor Xa		
min. IC_{50} [μM]	36.1	16.7	13.5	7.9
quartiles	32.2−40.0	11.6−21.8	6.4−21.4	6.2−13.9
# actives	32.0	103.5	112.0	112.5
quartiles	29.0−36.0	90.8−113.0	99.0−120.0	99.0−127.3
pop. size × gen.		3 × 100	37 × 8	20 × 15
		Factor Xa with Four Initial Actives		
min. IC_{50} [μM]	36.3	6.3	5.8	6.7
quartiles	33.1−40.2	5.8−8.0	5.5−6.5	6.2−12.1
# actives	32.0	113.0	118.0	98.5
quartiles	28.0−35.0	95.5−122.5	108.0−124.0	69.0−108.0
pop. size × gen.		2 × 150	30 × 10	2 × 150
		Trypsin		
min. IC_{50} [μM]	90.4	53.3	45.7	33.8
quartiles	86.7−93.4	39.8−74.7	27.8−69.1	27.9−58.7
# actives	6.0	24.0	30.5	40.0
quartiles	4.0−7.0	16.0−33.3	19.0−55.3	23.8−56.0
pop. size × gen.		2 × 150	20 × 15	20 × 15
		Trypsin with Four Initial Actives		
min. IC_{50} [μM]	90.6	30.7	29.4	30.9
quartiles	87.5−93.8	28.8−37.1	27.9−32.2	27.4−40.0
# actives	5.5	51.0	58.0	49.5
quartiles	4.0−7.0	37.8−57.0	53.5−62.0	32.8−65.0
pop. size × gen.		2 × 150	30 × 10	2 × 150
		Tryptase		
min. IC_{50} [μM]	18.7	5.4	2.4	2.2
quartiles	16.9−20.2	3.2−9.0	1.7−8.0	1.7−3.2
# actives	90.0	202.0	206.5	195.0
quartiles	84.0−95.0	182.8−211.3	196.5−216.0	180.0−206.0
pop. size × gen.		3 × 100	20 × 15	20 × 15
		uPA		
min. IC_{50} [μM]	59.5	32.8	31.2	27.7
quartiles	54.9−64.5	27.9−38.6	24.6−36.3	20.4−31.3
# actives	26.0	85.0	92.5	107.0
quartiles	23.0−29.0	77.0−94.0	84.8−100.0	93.0−115.0
pop. size × gen.		2 × 150	30 × 10	3 × 100

[a] pop. size × gen. denotes the best combination of population size and number of generations as determined by the mean IC_{50} of the top 20 most active compounds of an optimization (min. IC_{50} [μM]). Minimal IC_{50} values are given as the medians of 100 repetitions along with the lower and upper quartiles. The median number of found actives is given with # actives.

cooling phase. Jamois et al. observed a similar behavior in their comparison of SA versus GA.[23] This characteristic of SA is opposed to the optimization constraint of a preferably low number of iterations. If one synthesize-and-test cycle was assumed to take 2 weeks in a low-throughput experiment, a study with 60 generations would take 2.3 years to complete, which is certainly not desirable.

PSO produced best results with intermediate to high numbers of iterations, corresponding to intermediate to small population sizes. Thus, PSO appears to be generally suited as optimization algorithm for focused library design. PSO and ES performed comparably well, although larger error spans in optimizations with included actives in the initial population were observed for PSO. However, in case of the MOE 2D descriptor space, PSO produced excellent results even in optimizations without the inclusion of initial actives.

ES, on the other hand, preferred low to intermediate numbers of iterations and consequently higher population sizes. This renders them especially attractive for an application in low-throughput screening campaigns. Generally, optimization runs with as little as four generations (population size: 75) did still perform competitively (in optimizations with included initial actives). Speaking in terms of the above example of a realistic low-throughput experiment, four synthesize-and-test cycles could be completed within a

Table 5. Results of a Comparative Study with a Budget of 300 Compounds on the Ugi Data Set Encoded by the MOE 2D Principle Components[a]

	Random Search	Simulated Annealing	Evolution Strategy	Particle Swarm
Chymotrypsin				
min. IC_{50} [μM]	109.5	108.3	108.2	108.5
quartiles	105.0−110.0	104.7−110.0	104.5−110.0	103.4−110.0
# actives	0.0	1.0	1.0	1.0
quartiles	0.0−1.0	0.0−1.0	0.0−2.0	0.0−2.0
pop. size × gen.		15 × 20	3 × 100	75 × 4
Chymotrypsin with Four Initial actives				
min. IC_{50} [μM]	109.5	89.2	94.4	93.5
quartiles	106.4−110.0	84.1−94.7	89.3−99.7	88.6−105.8
# actives	0.0	4.0	3.0	3.0
quartiles	0.0−1.0	3.0−5.0	2.0−4.0	1.0−4.0
pop. size × gen.		3 × 100	100 × 3	1 × 300
Factor Xa				
min. IC_{50} [μM]	35.8	13.6	11.8	12.3
quartiles	32.3−39.8	8.9−19.3	6.2−19.0	8.7−19.7
# actives	32.0	106.0	123.0	95.0
quartiles	29.0−36.0	96.0−116.0	108.8−134.0	81.3−110.0
pop. size × gen.		2 × 150	20 × 15	20 ×15
Factor Xa with Four Initial Actives				
min. IC_{50} [μM]	36.5	6.6	6.2	8.9
quartiles	33.2− 40.3	5.9−8.0	5.7−6.6	6.9−10.5
# actives	32.0	105.5	113.5	78.0
quartiles	28.0−35.3	96.8−116.0	105.8−119.3	69.0−92.3
pop. size × gen.		3 × 100	30 × 10	5 ×60
Trypsin				
Min IC_{50} [μM]	90.3	48.0	37.8	42.7
quartiles	86.8−93.3	29.5−71.1	27.6−59.9	29.8−62.2
# actives	6.0	29.0	38.0	31.0
quartiles	4.0−7.0	17.8−49.0	21.8−51.0	20.0−47.0
pop. size × gen.		1 × 300	60 × 5	10 × 30
Trypsin with Four Initial Actives				
min. IC_{50} [μM]	90.6	28.3	27.0	29.6
quartiles	87.0−93.6	26.0−29.9	25.2−28.6	28.2−48.2
# actives	6.0	54.0	62.0	51.5
quartiles	4.0−7.0	49.8−59.0	58.0−65.0	25.8−55.0
pop. size × gen.		3 × 100	30 × 10	1 ×300
Tryptase				
min. IC_{50} [μM]	18.7	5.2	2.3	2.5
quartiles	17.2−20.2	3.0−9.7	1.7−6.9	1.9−7.3
# actives	89.0	190.0	197.5	177.0
quartiles	84.0−95.0	182.8−198.0	189.8−205.3	162.5−188.3
pop. size × gen.		2 × 150	30 × 10	5 ×60
uPA				
min. IC_{50} [μM]	59.4	28.4	26.4	19.5
quartiles	54.6−64.6	25.2−33.1	23.9−30.2	16.2−28.5
# actives	26.0	97.0	103.0	110.0
quartiles	23.0−30.0	85.0−106.0	87.0−115.3	97.8−120.3
pop. size × gen.		2 × 150	10 × 30	2 × 150

[a] pop. size×gen. denotes the best combination of population size and number of generations as determined by the mean IC_{50} of the top 20 most active compounds of an optimization (min. IC_{50} [μM]). Minimal IC_{50} values are given as the medians of 100 repetitions along with the lower and upper quartiles. The median number of found actives is given with # actives.

reasonable 2-month period. With an activity-enriched initial population preferred combinations of "population size" × "number of generations" ranged from 20 × 15 to 75 × 4, thus providing sufficient flexibility in the experimental setup of a low-throughput study.

Our modifications of SA, ES, and PSO turned out favorably. The algorithms were modified to reject solution candidates which exceed a certain fitness threshold (see "Materials and Methods"). This makes sense in scenarios where "inactive" solution candidates (here: compounds with $IC_{50} \geq 100\ \mu$M) exist. It seems intuitive from a practical point of view to exclude inactives from further consideration. Also, the relative invariance of factor Xa optimizations (especially with the CATS descriptor, Figure 2) to different combinations of population size by number of generations was a result of consequent modifications of the employed optimization algorithms toward adaptive molecular library optimization (see "Materials and Methods"). For instance, PSO benefits from the global *tabu* rule and optimization by ES with small population sizes was only made possible with the introduction of an elitism of one individual. Generally speaking, the harder the optimization target, the more crucial is an optimal choice of population size and number of generations.

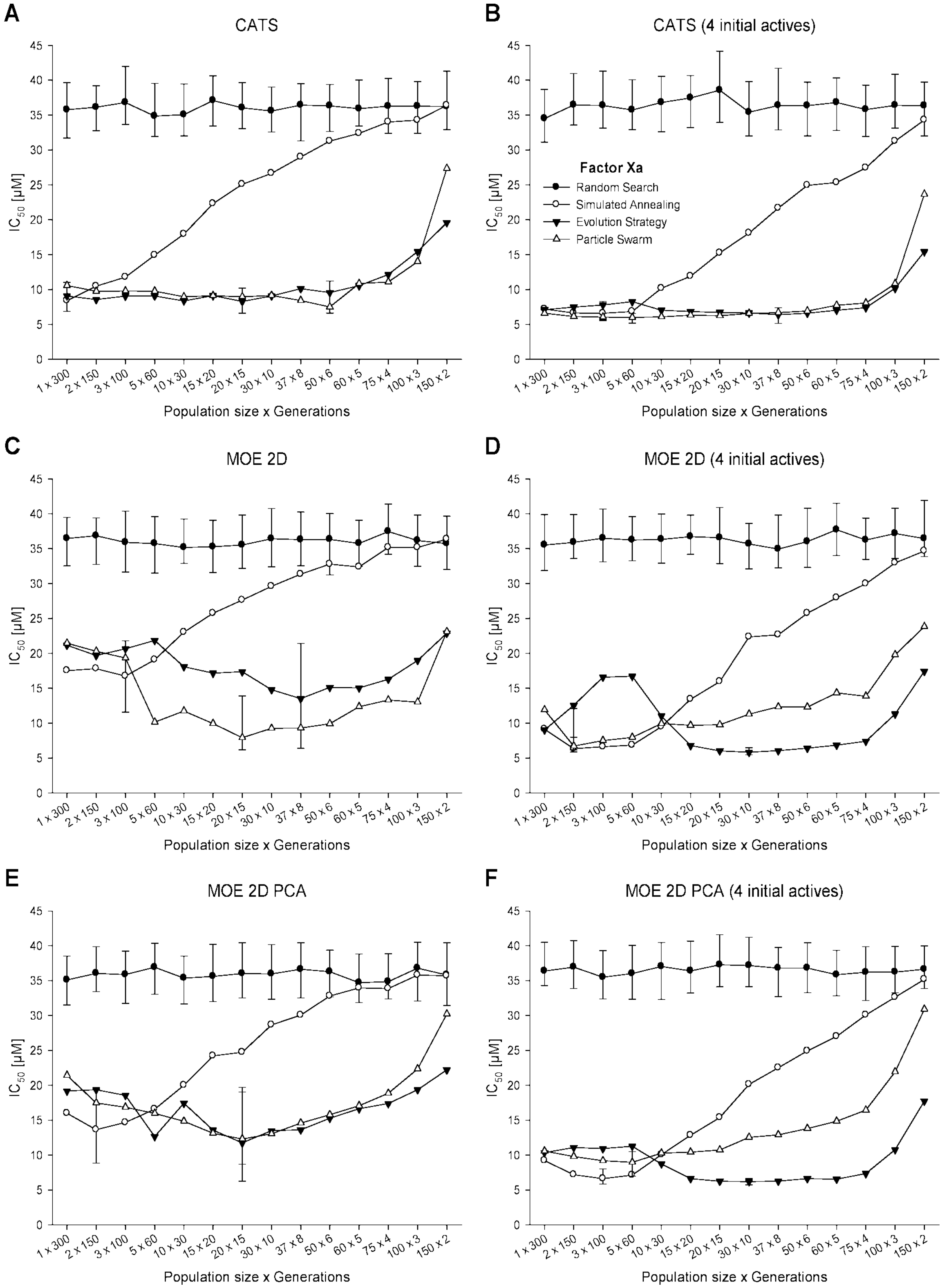

Figure 2. Comparison of different combinations of population size by number of generations for optimizations of factor Xa inhibitors on the *Ugi data set*. The mean IC_{50} of the top 20 most active compounds are plotted as medians of 100 repetitions. Error bars indicate lower and upper quartiles and are given for Random Search and the optimal combinations of the other optimization algorithms.

An unusual decrease in ES optimization results for population sizes of 1 to 5 is observable in Figures 2 and 3. This behavior was a consequence of our rule to calculate the number of parents as *population size/6*. Hence for population sizes 1 to 5 there was a single parent: of four initial actives, three were neglected. Simultaneously, the number of generations decreased from 300 to 60, making optimization even more challenging to ES.

In-Depth Understanding of Adaptive Library Design. It is remarkable that a straightforward ES with as little as four generations still yielded competitive optimization results. We therefore decided to analyze the

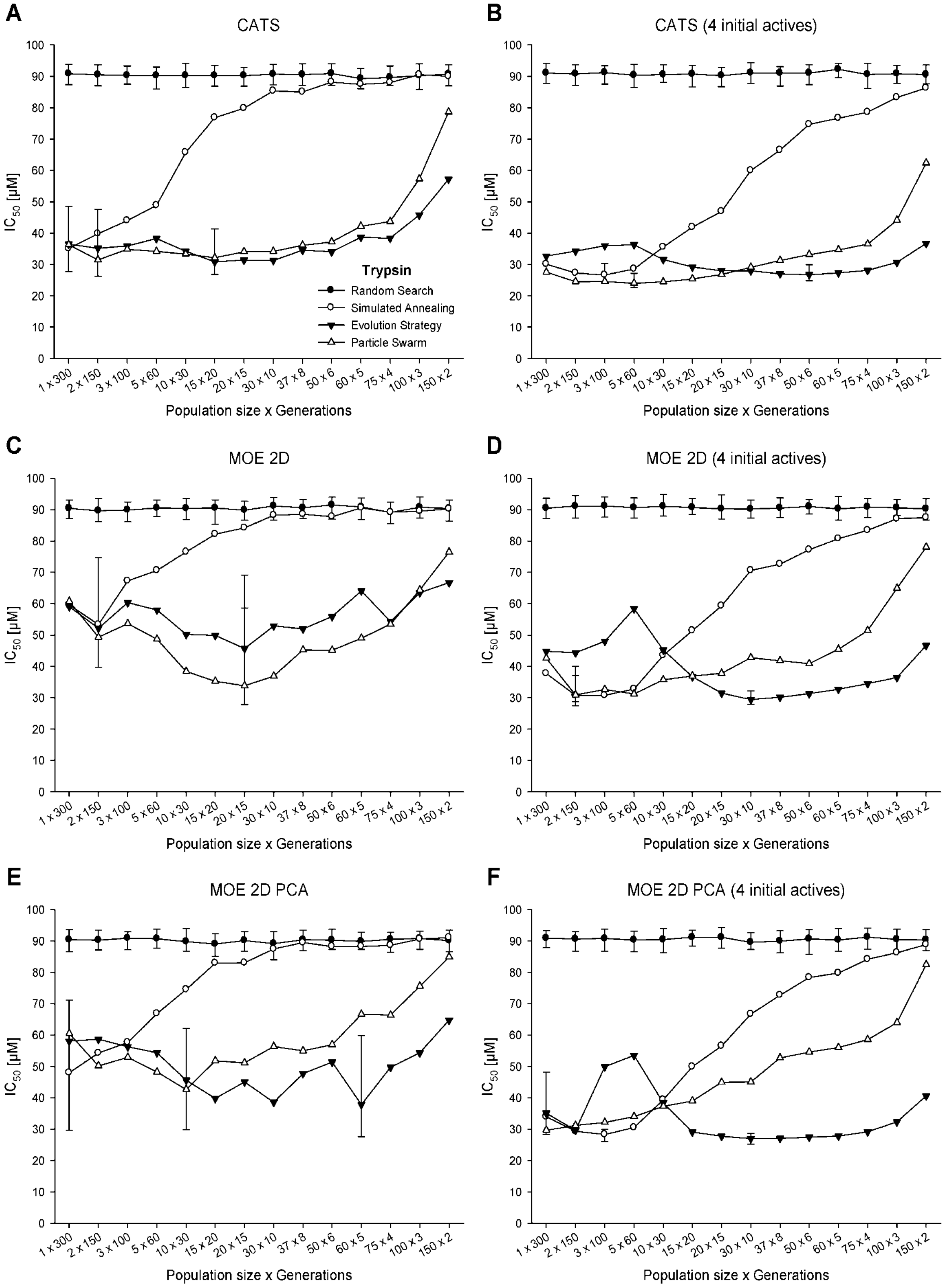

Figure 3. Comparison of different combinations of population size by number of generations for optimizations of trypsin inhibitors on the *Ugi data set*. The mean IC_{50} of the top 20 most active compounds are plotted as medians of 100 repetitions. Error bars indicate lower and upper quartiles and are given for Random Search and the optimal combinations of the other optimization algorithms.

different descriptor spaces by Self-Organizing Maps (SOM). SOMs belong to the class of unsupervised neural networks and were pioneered by Kohonen in the early 1980s.[66] We employed the software tool MOLMAP to create two-dimensional maps of our high-dimensional descriptor spaces.[67] SOMs seek to conserve local neighborhood relations during projection, thus adjacent areas on a SOM correspond to adjacent regions in high-dimensional space. 20 × 20 SOMs with a toroidal topology were trained for 1,000,000 iterations with all 15,840 *Ugi data set* molecules. The torus ("doughnut" shape) represents an infinite two-dimensional space. The

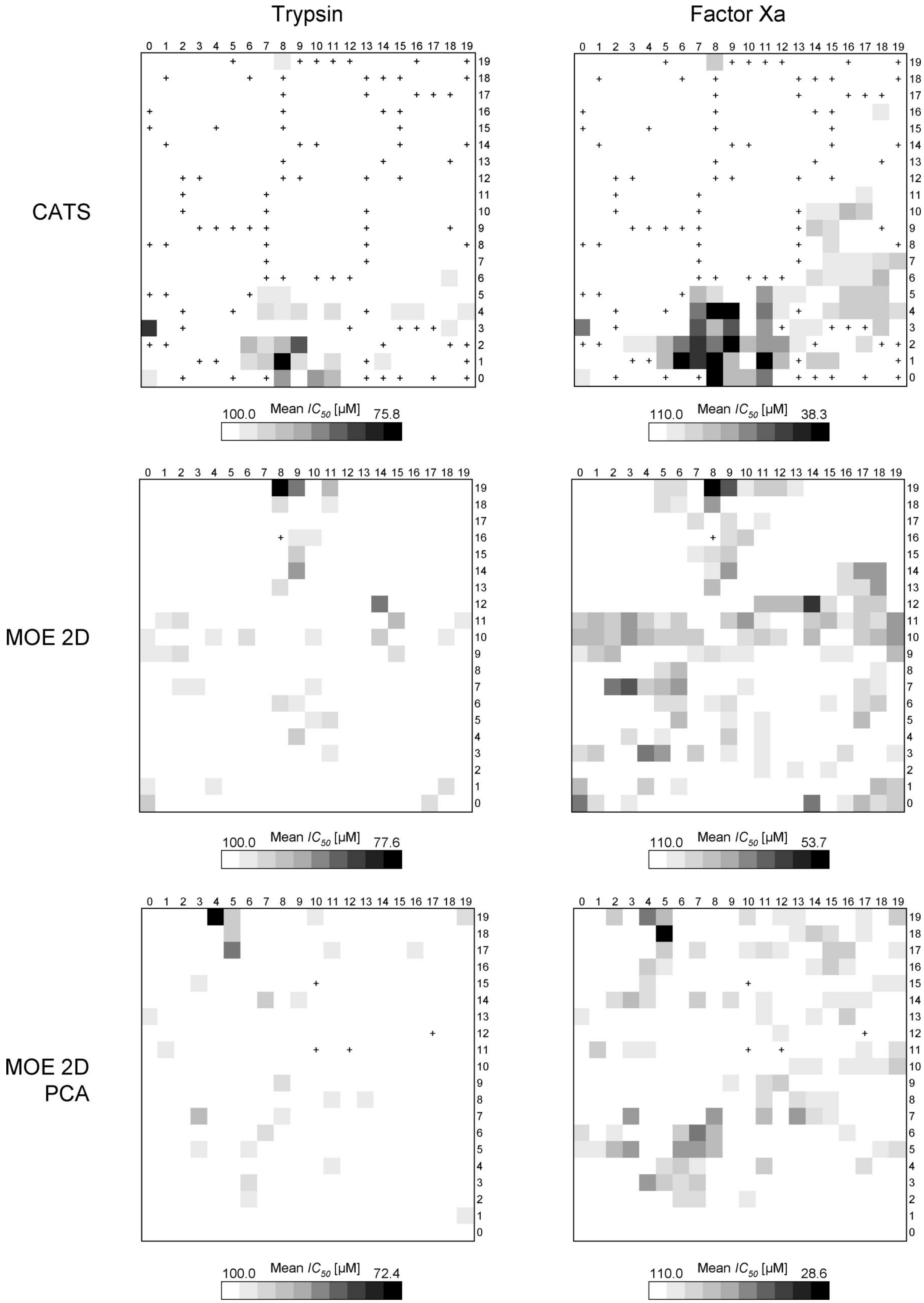

Figure 4. Toroidal self-organizing maps (SOM) of the high dimensional descriptor spaces of the *Ugi data set*. Mean IC_{50} values against trypsin (left side) and factor Xa (right side) are projected on the map. Mean values were calculated by averaging the IC_{50} values of the compounds assigned to a specific neuron. Light areas indicate prevalence of inactive compounds, and dark areas indicate prevalence of active compounds. Plus signs denote empty neurons.

upper and lower side as well as the left and right side of the maps are connected with each other.

Figure 4 depicts the mean IC_{50} values projected onto the maps. Mean values were calculated by averaging the IC_{50} values of the compounds assigned to a specific neuron. Light areas indicate prevalence of inactive compounds, dark areas indicate prevalence of active compounds, and plus signs denote empty neurons. We chose to project mean IC_{50} values onto the maps because this form of presentation resembles the search space for optimization. Shaded areas on the maps not only indicate the presence of active compounds but also indicate their relative frequency compared to inactive compounds as well as their average potency. A clear separation of actives and inactives was not visible in the SOM projections. Inactive compounds generally occupied the whole SOM area. However, active compounds formed clusters. Trypsin and factor Xa inhibitors occupy similar areas in the map. This is also reflected by their activity values: 92% of all trypsin inhibitors also exhibit activity against factor Xa. Trypsin and factor Xa are known to possess very similar binding pockets.[71] Both, active trypsin and factor Xa inhibitors, cluster nicely in the search space spanned by the CATS descriptor. This is an indication as to why optimization runs on the CATS descriptor generally performed better than on the MOE 2D or MOE 2D PCA descriptors. Active molecules on the MOE 2D and MOE 2D PCA self-organizing maps form clusters as well, although to a lesser extent. Their distribution of activities looks similar to each other, reflecting their common source of descriptors. Correspondingly, their performance in the optimization runs was similar as well.

The different degree of clustering might also explain the different effects of the inclusion of known actives within the initial population. In the CATS descriptor space optimization results did only improve a little with the inclusion of known actives. Due to the dense cluster of actives the optimization algorithm could spend more time on *exploiting* the close neighborhood, rather than on *exploring* the search space to follow a fitness gradient. The inclusion of known actives had a greater impact on optimizations in the MOE 2D and MOE 2D PCA descriptor spaces. Due to poorer clustering the optimization algorithms needed to explore a greater hypervolume in the search space to find potent inhibitors. The inclusion of known actives in the initial population short-cut the exploration phase by directly placing the optimization algorithm into areas of high activity (“activity islands”).[72] More time was then available for exploiting those activity islands yielding better optimization results.

To better understand the internals of ES and PSO-based molecular library optimization we collected additional data from the best ES (50 × 6) and PSO (5 × 60) optimization of trypsin inhibitors, using the CATS descriptor and four actives in the initial population. We did not include SA in this analysis as its optimization behavior suggested less suitability for a prospective low-throughput screening campaign. In optimizations performed by ES, compounds selected as offspring were on average the 33rd neighbor of their parents. An interesting peculiarity of discrete molecular descriptor spaces is noteworthy here: While ES operates in continuous space, molecular descriptor spaces represent sparsely populated, discrete spaces. The position vector of a selected parent is varied by a mutation operator to yield a virtual offspring in continuous space. It is almost certain not to encounter a “real” molecule at those arbitrary positions in chemical space. We therefore introduced a nearest neighbor step to map from the virtual offspring in continuous space to a nearby located molecule in discrete chemical space. The median distance of a parent to its virtual offspring was 94 (measured in arbitrary units with Manhattan metric). However, the actual distance of a parent to its real offspring was only 28 units. This surprising effect might be explained by the phenomenon of “silent mutations”.[73] Some mutations to the position vector (“genotype”) did not result in a variation of the phenotype (molecule) as those areas of chemical space were not populated by molecules.

Curious about the impact of the additional nearest neighbor step we then varied the step size parameter of the ES. Step sizes from 0 to 3 were acceptable, corresponding to parent↔virtual offspring distances of 0 to 126 units. Those distances in turn corresponded to parent↔real offspring distances of 19 to 38 units. This indicates that the ES effectively operated in a locally restricted nearest neighbor space. Furthermore, the nearest neighbor step had a considerable influence on the search behavior as it dampened the effect of too small or too large step sizes. Note, that a step size of zero was only possible due to the implemented *tabu* of ever selecting a library molecule twice. This global *tabu* rule and the nearest neighbor step efficiently prevented premature convergence.

Statistical data obtained from PSO runs, in contrast, suggest that PSOs operate more widespread. In the analyzed case, the median parent↔real offspring distance was 47 units compared to the median parent↔virtual offspring distance of 92 units. A moved particle was on average the 356th neighbor of its original position. These data could, however, not capture the interesting dynamics of PSO. We therefore analyzed individual PSO runs separately.

Starting with very large jumps in search space in the very beginning of a PSO run, the swarms rapidly converged within only a few iterations on one of the three reference points of PSO: a particle’s current position, a particle’s overall best position, and the swarm’s overall best position. In this initial phase parent↔real offspring distances went down from 143 units (spanning the whole search space) to 8.6 units (a direct neighbor). This first exploration phase was needed to calibrate the velocity vectors which were randomly initialized. The further behavior is best described as “pulsating”. The swarm periodically expanded and constricted, thereby exploring positions extrapolated from the particles’ current, local best, and global best positions. Step sizes of the “flying” particles varied from jumps spanning half the search space to only marginal movement. The overall trajectory was guided by a swarm’s globally best position. Once a new globally best position was found, the swarm gradually migrated toward it. The effect of the additional nearest neighbor calculation is evident in PSO as well: parent↔virtual offspring distances were about twice as large as parent↔real offspring distances in the CATS descriptor space. PSOs benefited from the *tabu* of ever selecting a library molecule twice. Particularly in optimization setups with a high number of iterations the *tabu* rule efficiently prevented premature convergence.

Summarizing, Evolution Strategies prefer small steps from parent to offspring and thus base their success on parent-offspring similarity and simulated Darwinian “natural selection”. Success in Particle Swarm Optimizations is based on moving particles to extrapolated positions of their current position, their overall best position, and the swarm’s best position. As a consequence, there is less molecular similarity between a particle’s new and old position. This behavior is exemplified in Figure 5. Structures on the left side (**2**−**5**)

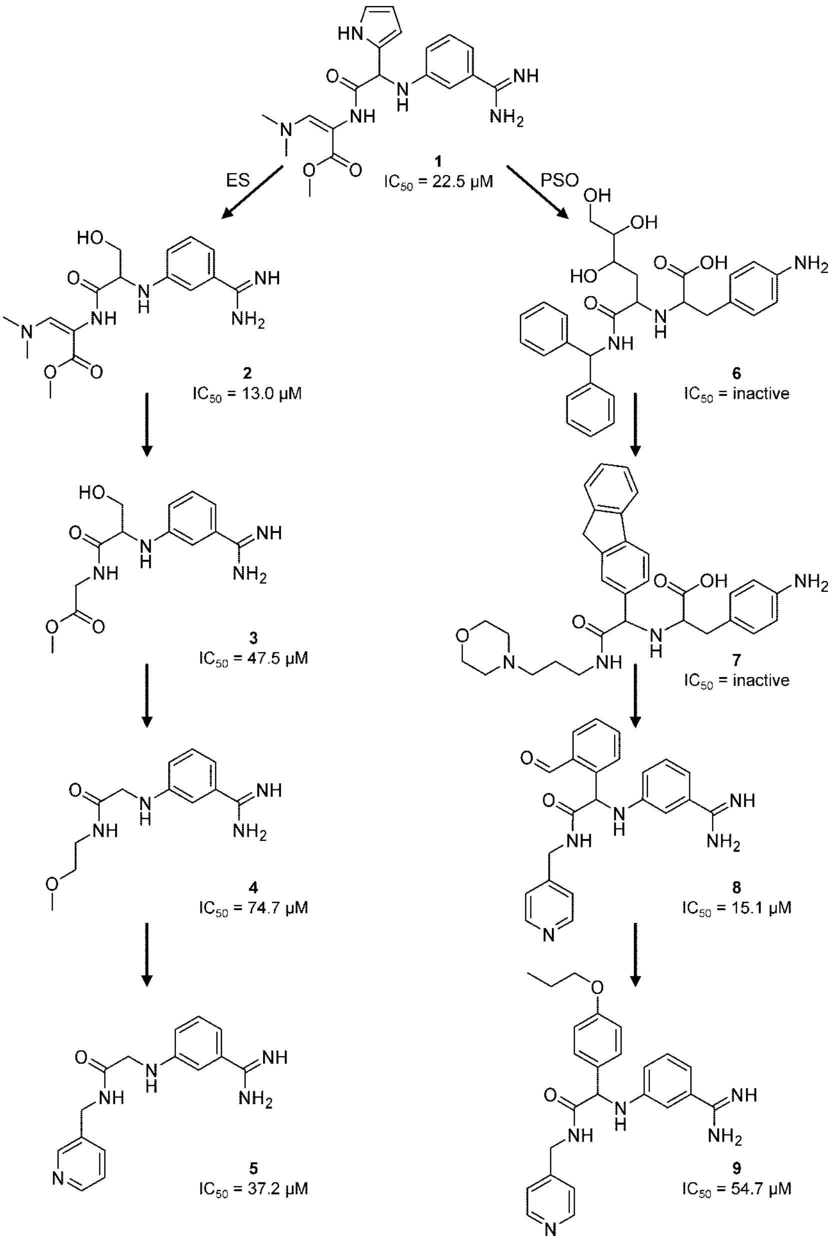

Figure 5. Comparison of the five first generations of a typical ES and PSO run. IC_{50} values indicate trypsin inhibition. Both optimization algorithms started with reference structure **1**. Successful parent molecules are depicted for ES, and the stations of the movement of one particle are shown for PSO.

represent successful parents of an ES with reference compound **1** included in the initial population. The right side represents the stations (compounds **6**–**9**) of a particle guided through chemical space by PSO. The ES operated by repetitive small structural variations of the reference compound. Greater structural variations were observable for the movement of a particle. In this particular example, the first two candidate compounds following the reference structure were found to be inactive. However, guided by its cognitive and social memory the particle returned to an activity island, presumably, close to where it started from. ES does encounter inactive candidate compounds as well. Yet, those individuals "die off" and are not considered as parents for spawning a new generation of compounds. It is noteworthy that the example of trypsin inhibitor design is characterized by a peculiarity: The benzamidine moiety found in compounds **1**–**5**, **8**, and **9** is a preferred fragment binding to the S_1 substrate recognition pocket of the enzyme. 97% of all Ugi-type trypsin inhibitors of our data set contained this fragment as a building block. Therefore, all successful designs in our example contained this preferred substructure. A similar observation had been made for Ugi-type products that were evolved to block thrombin activity.[16,26]

Optimization with Binary Activity. Comparative molecular library optimization results were then collected for the COBRA collection of drugs and lead structures (v. 6.1, 8311 reference compounds). Each compound was annotated with target receptor information and activity labels. The only serine protease sufficiently abundantly represented in COBRA was factor Xa, with 228 (2.7%) actives. Unlike for the Ugi-type products, this time only binary activity values (active vs inactive) were used. Consequently, the performance of an optimization algorithm was measured by its ability to retrieve as many actives as possible. As COBRA contained about half the number of compounds as the *Ugi data set*, we reduced the budget of compound evaluations

Table 6. Results of a Comparative Study with a Budget of 150 Compounds on COBRA[a]

	Random Search	Simulated Annealing	Evolution Strategy	Particle Swarm
Factor Xa CATS				
max. # actives	4.0	18.0	47.5	52.5
quartiles	3.0−5.0	10.8−28.3	28.8−63.0	29.8−70.3
max. % actives	2.7%	12.0%	32.3%	35.5%
quartiles	2.0%−3.3%	7.2%−18.8%	19.6%−42.9%	20.1%−47.5%
pop. size × gen.		3 × 50	7 × 21	4 × 37
Factor Xa CATS with Four Initial Actives				
max. # actives	4.0	57.5	61.5	71.5
quartiles	3.0−5.0	44.0−63.0	45.0−69.0	49.3−80.0
max. % actives	2.7%	38.3%	41.0%	47.7%
quartiles	2.0%−3.3%	29.3%−42.0%	30.0%−46.0%	32.8%−53.3%
pop. size × gen.		3 × 50	15 × 10	5 × 30
Factor Xa MOE 2D				
max. # actives	4.0	29.5	53.5	72.0
quartiles	3.0−5.0	7.8−46.3	39.3−77.3	53.5−89.0
max. % actives	2.7%	19.7%	35.7%	48.0%
quartiles	2.0%−3.3%	5.2%−30.8%	26.2%−51.5%	35.7%−59.3%
pop. size × gen.		1 × 150	2 × 75	1 × 150
Factor Xa MOE 2D with Four Initial Actives				
max. # actives	4.0	60.0	66.5	84.5
quartiles	2.0−5.0	52.8−66.0	46.0−81.0	69.0−94.0
max. % actives	2.7%	40.0%	44.3%	56.3%
quartiles	1.3%−3.3%	35.2%−44.0%	30.7%−54.0%	46.0%−62.7%
pop. size × gen.		2 × 75	2 × 75	1 × 150
Factor Xa MOE 2D PCA				
max. # actives	4.0	24.5	43.0	50.0
quartiles	3.0−5.0	8.0−37.3	27.8−56.3	35.8−64.3
max. % actives	2.7%	16.3%	28.7%	33.3%
quartiles	2.0%−3.3%	5.3% −24.8%	18.5%−37.5%	23.8%−42.8%
pop. size × gen.		1 × 150	3 × 50	2 × 75
Factor Xa MOE 2D PCA with Four Initial Actives				
max. # actives	4.0	47.0	54.0	64.5
quartiles	3.0−5.0	38.0−54.0	50.0−58.0	52.0−79.0
max. % actives	2.7%	31.3%	36.0%	43.0%
quartiles	2.0%−3.3%	25.3%−36.0%	33.3%−38.7%	34.7%−52.7%
pop. size × gen.		1 × 150	15 × 10	1 × 150

[a] "pop. size × gen." denotes the best combination of population size and number of generations as determined by the maximal number of active compounds found in an optimization ("max. # actives"). The number of actives is given as the median of 100 repetitions along with the lower and upper quartiles. "max. % actives" gives the maximal percentage of found actives compared to the number of evaluations. A value of 100% is reached when finding exactly 150 active compounds in an overall optimization.

from 300 to 150. The goal of the optimization study with COBRA was to determine whether binary activity data are sufficient for optimization and whether the optimization parameters obtained with the *Ugi data set* would also work on a different data set. Again, random initial population selection, and the inclusion of four known actives within the initial population were considered. The four actives were chosen randomly from all 228 factor Xa inhibitors.

Overall, the optimization results resembled those obtained with the *Ugi data set* (Table 6, Figure 6). Best results could be achieved with PSO (1 particle, 150 iterations) on the MOE 2D descriptors with a success rate of 56%. All three algorithms performed effectively with binary activity data and greatly outperformed a Random Search. It should be noted that a small algorithmic modification to the memory update rule of PSO was necessary to adapt to a binary fitness function (see "Materials and Methods"). However, this change did not influence results obtained with the *Ugi data set* in any way.

The greatest difference to the results obtained with the *Ugi data set* is the remarkably poor performance of SA, especially in setups without the inclusion of known actives in the initial population. SA proved to be quite sensitive for changes of the step size parameter. With a separately optimized step size parameter, we were able to produce results as good as PSO with SA on COBRA as well. Yet, we decided not to use a different step size setting for demonstration purposes. In a prospective screening study a prior optimization of a step size parameter will most likely be infeasible. The effect is less dramatic for optimizations with included initial actives. In accordance to the results obtained with the *Ugi data set*, SA preferred high numbers of generations and is thus−in this implementation−less suited for an application in low-throughput screening campaigns. ES again favored fewer iterations and greater populations. ESs performed slightly worse than PSO for the same reason as SA, namely a suboptimal step size. However, their population-based character rendered them much more robust against changes of the step size parameter. PSO−again preferring more iterations and smaller population sizes−performed best on COBRA in all cases. The constriction swarm demonstrated an excellent optimization behavior due to its autoadaptive step size control which makes it an ideal out-of-the-box optimization strategy.

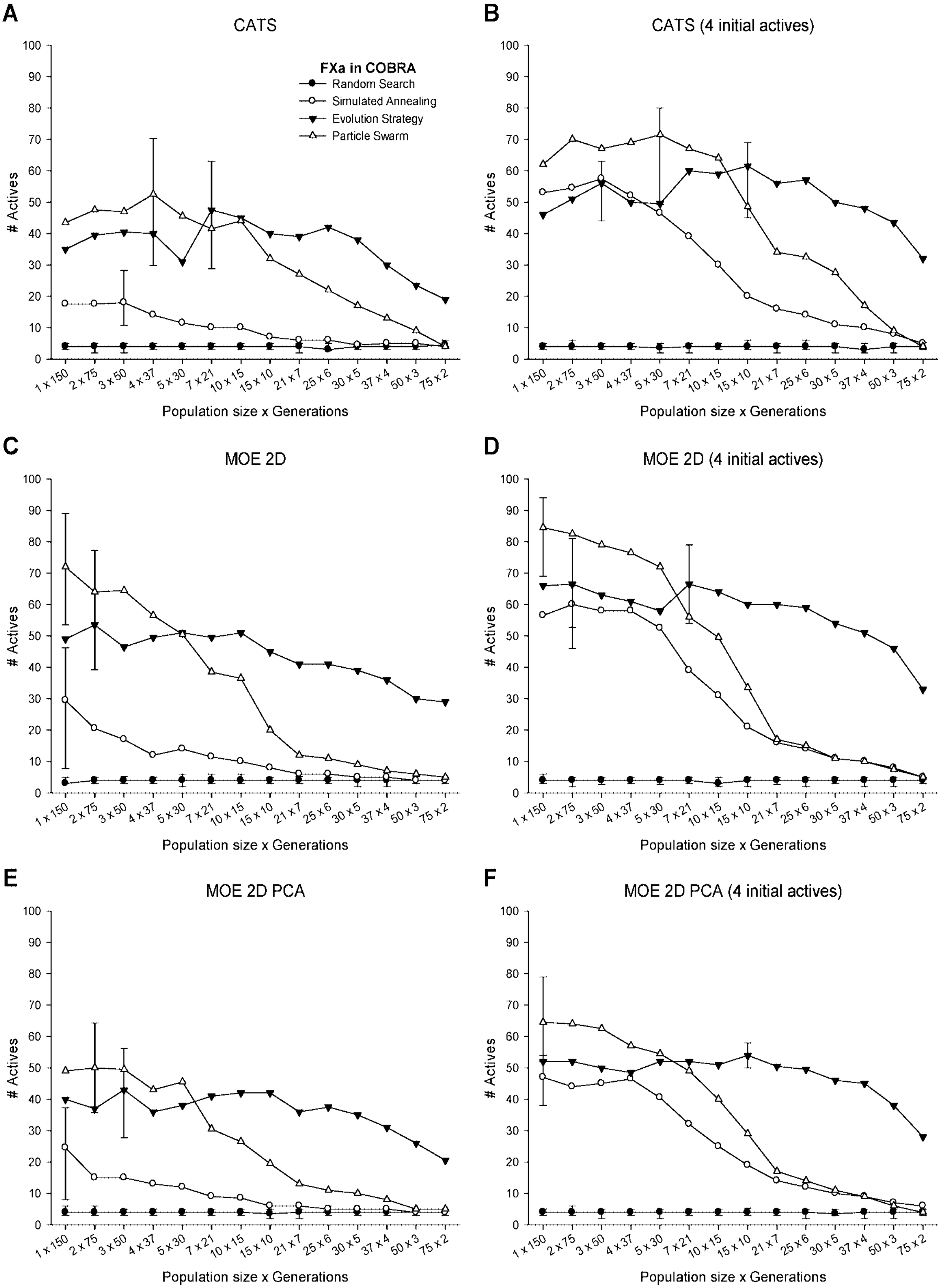

Figure 6. Comparison of different combinations of population size by number of generations for optimizations of factor Xa inhibitors of COBRA. The number of found active molecules (# actives) is plotted as medians of 100 repetitions. Error bars indicate lower and upper quartiles and are given for Random Search and the optimal combinations of the other optimization algorithms.

SOM analysis of the three descriptor spaces revealed a distinct distribution of factor Xa inhibitors (Figure 7). 20 × 20 SOMs with a toroidal topology were trained for 1,000,000 iterations with all 8311 COBRA molecules. This time, the SOM of the CATS descriptor space was trained using the Euclidean distance metric instead of the Manhattan metric, and descriptor values were not autoscaled. The percentage of active compounds against factor Xa is projected on the map. The percentage was determined by calculating the ratio of the number of active compounds to the number of inactive compounds assigned to a specific neuron. Light areas indicate a prevalence of inactive compounds, dark areas indicate prevalence of active compounds, and plus ("+") signs denote empty neurons. Thus, dark areas on the maps not only

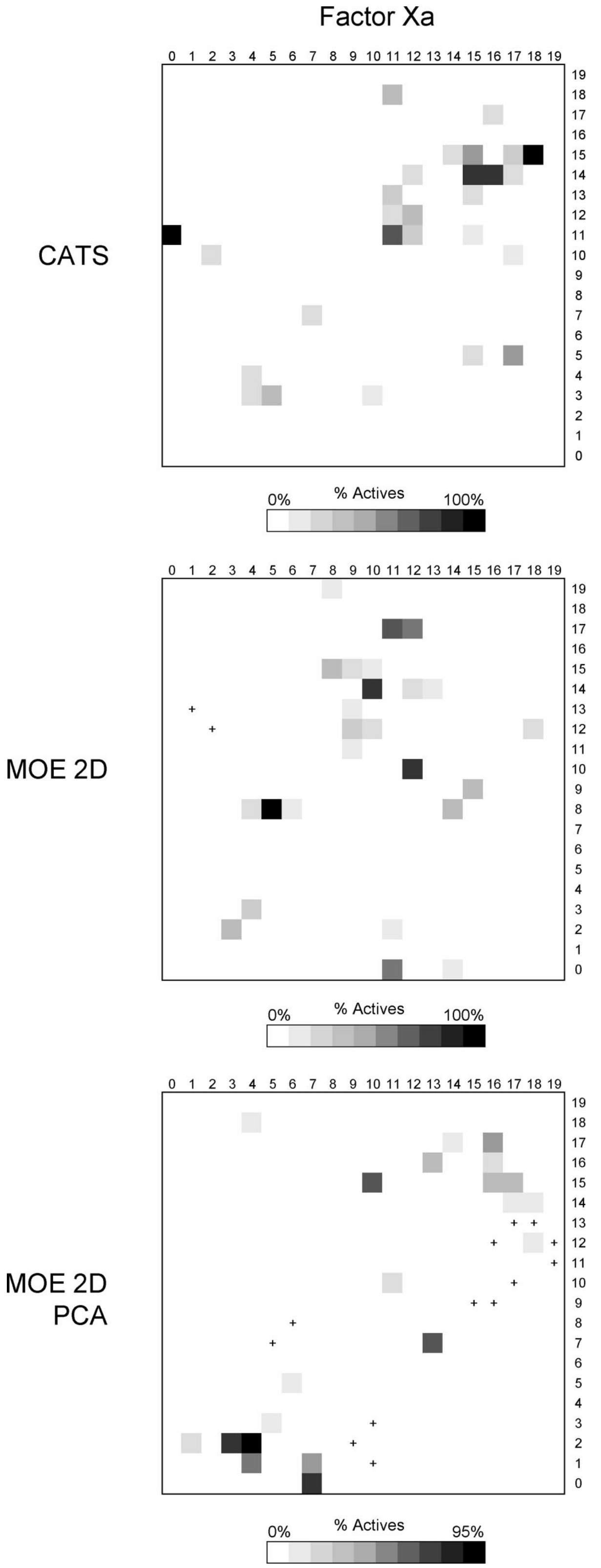

Figure 7. Toroidal self-organizing maps (SOM) of the high-dimensional descriptor spaces of COBRA. The percentage of active compounds against factor Xa is projected on the map. The percentage was determined by calculating the ratio of the number of active compounds to the number of inactive compounds assigned to a specific neuron. Light areas indicate prevalence of inactive compounds, and dark areas indicate prevalence of active compounds. Plus signs denote empty neurons.

indicate the presence of active compounds but also indicate their relative frequency compared to inactive compounds. Similar to SOMs trained on the *Ugi data set*, active compounds formed clusters on the maps, which facilitates optimization. However, in contrast to the SOMs of the *Ugi data set*, several dense clusters with up to 100% actives were found. The corresponding areas in search space seem to be highly enriched with active factor Xa inhibitors. When one of those dense clusters is hit by a search algorithm, the further optimization will proceed more readily in the activity-enriched neighborhood. This might explain the surprisingly good result of retrieving roughly 50% actives among only 150 evaluated library compounds (where 75 active molecules represent 33% of all factor Xa actives contained in COBRA). While the Ugi data set represents a focused library comprised of combinatorial reaction products designed toward inhibition of serine proteases, the Collection of Bioactive Reference Analogues (COBRA) is more diverse and contains compounds directed against more than 300 biological targets. We have already demonstrated earlier that compounds of COBRA directed against distinct target receptors are partially separable by SOM analysis.[55,74]

CONCLUSIONS

We demonstrated the usefulness of different optimization algorithms for adaptive focused library design under the constraint of limited compound and testing resources and provided an in-depth analysis of the internal workings of two such algorithms. Already over a decade ago Weber et al. reported successful optimization of a library of 160,000 molecules with a budget of only 400 compounds.[26] They employed a Genetic Algorithm to yield submicromolar thrombin inhibitors. Here, we compared three mutually inherently different optimization algorithms: Simulated Annealing, Evolution Strategy, and Particle Swarm Optimization. With suitable setup parameters all three algorithms were able to produce good results. However, the Evolution Strategy was in best agreement with our secondary optimization constraint: As we had low-throughput screening in mind, an optimization algorithm should preferably operate with few cycles and large populations. Only the Evolution Strategy generally fulfilled this criterion. Particle Swarm Optimization performed sufficiently well with intermediate numbers of iterations, which makes it still suitable for low-throughput experiments. It proved to be the most robust out-of-the-box optimization strategy. Required search parameters were either autoadaptive (step size control), or the default values worked acceptably in our experiments (cognitive and social memory constants). Further efforts are required to find suitable algorithmic modifications in order to diminish PSO’s need for higher numbers of iterations.

All three optimization algorithms required particular modifications to produce best results. Among those the special treatment of inactive compounds and a global *tabu* rule had the greatest impact.

We further demonstrated that focused library optimization without any prior knowledge about the biological target is possible. The prerequisites were a sufficient abundance of active compounds in the screening library (at least 2%) and a reasonable clustering of the actives in the chemical search space applied. This finding might be helpful for the evaluation of the suitability of HTS results for subsequent lead optimization.

ACKNOWLEDGMENT

We thank Dr. Lutz Weber for the *Ugi data set* and Michael Meissner for providing the PSO code. Matthias Rupp and Dr. Ulrich Fechner are thanked for fruitful discussion. This work was supported by the Beilstein-Institut zur Förderung der Chemischen Wissenschaften, the German Research Foundation (DFG) Sonderforschungsbereich 579 (project A11.2), and the Fonds der Chemischen Industrie.

Supporting Information Available: Pseudocode of our implementations of the applied optimization algorithms “Random Search”, “Simulated Annealing”, “Evolution Strategy”, and “Particle Swarm Optimization”. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Bleicher, K. H.; Böhm, H.; Müller, K.; Alanine, A. I. Hit and Lead Generation: Beyond High-Throughput Screening. *Nat. Rev. Drug. Discovery* **2003**, *2*, 369–378.

(2) Bohacek, R. S.; McMartin, C.; Guida, W. C. The Art and Practice of Structure-Based Drug Design: A Molecular Modeling Perspective. *Med. Res. Rev.* **1996**, *16*, 3–50.

(3) Lipinski, C.; Hopkins, A. Navigating Chemical Space for Biology and Medicine. *Nature* **2004**, *432*, 855–861.

(4) Walters, W. P.; Stahl, M. T.; Murcko, M. A. Virtual Screening - an Overview. *Drug Discovery Today* **1998**, *3*, 160–178.

(5) Eglen, R. M.; Schneider, G.; Böhm, H. High-Throughput Screening and Virtual Screening: Entry Points to Drug Discovery. In *Virtual Screening for Bioactive Molecules*; Böhm, H., Schneider, G., Eds.; Wiley-VCH: Weinheim, New York, 2000; Vol. 10, pp 1−14.

(6) Drewry, D. H.; Young, S. S. Approaches to the Design of Combinatorial Libraries. *Chemom. Intell. Lab. Syst.* **1999**, *48*, 1–20.

(7) Bajorath, J. Integration of Virtual and High-Throughput Screening. *Nat. Rev. Drug. Discovery* **2002**, *1*, 882–894.

(8) Miller, J. L. Recent Developments in Focused Library Design: Targeting Gene-Families. *Curr. Top. Med. Chem.* **2006**, *6*, 19–29.

(9) Lahana, R. How Many Leads from HTS. *Drug Discovery Today* **1999**, *4*, 447–448.

(10) Schneider, G. Trends in Virtual Combinatorial Library Design. *Curr. Med. Chem.* **2002**, *9*, 2095–2101.

(11) Schneider, G.; Böhm, H. Virtual Screening and Fast Automated Docking Methods. *Drug Discovery Today* **2002**, *7*, 64–70.

(12) Mason, J. S.; Good, A. C.; Martin, E. J. 3-D Pharmacophores in Drug Discovery. *Curr. Pharm. Des.* **2001**, *7*, 567–597.

(13) Winkler, D. A. The Role of Quantitative Structure-Activity Relationships (QSAR) in Biomolecular Discovery. *Brief. Bioinf.* **2002**, *3*, 73–86.

(14) Schneider, G.; Fechner, U. Computer-Based De Novo Design of Drug-Like Molecules. *Nat. Rev. Drug. Discovery* **2005**, *4*, 649–663.

(15) Sheridan, R. P.; Kearsley, S. K. Using a Genetic Algorithm To Suggest Combinatorial Libraries. *J. Chem. Inf. Comput. Sci.* **1995**, *35*, 310–320.

(16) Weber, L. Current Status of Virtual Combinatorial Library Design. *QSAR Comb. Sci.* **2005**, *24*, 809–823.

(17) Holland, J. H. *Adaptation in Natural and Artificial Systems*; University of Michigan Press: Ann Arbor, MI, 1975.

(18) Brown, R. D.; Martin, Y. C. Designing Combinatorial Library Mixtures Using a Genetic Algorithm. *J. Med. Chem.* **1997**, *40*, 2304–2313.

(19) Gillet, V. J.; Willett, P.; Bradshaw, J.; Green, D. V. S. Selecting Combinatorial Libraries to Optimize Diversity and Physical Properties. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 169–177.

(20) Gillet, V. J.; Khatib, W.; Willett, P.; Fleming, P. J.; Green, D. V. S. Combinatorial Library Design Using a Multiobjective Genetic Algorithm. *J. Chem. Inf. Comput. Sci.* **2002**, *42*, 375–385.

(21) Gillet, V. J.; Willett, P.; Fleming, P. J.; Green, D. V. S. Designing Focused Libraries Using MoSELECT. *J. Mol. Graphics Modell.* **2002**, *20*, 491–498.

(22) Illgen, K.; Enderle, T.; Broger, C.; Weber, L. Simulated Molecular Evolution in a Full Combinatorial Library. *Chem. Biol.* **2000**, *7*, 433–441.

(23) Jamois, E. A.; Lin, C. T.; Waldman, M. Design of Focused and Restrained Subsets from Extremely Large Virtual Libraries. *J. Mol. Graphics Modell.* **2003**, *22*, 141–149.

(24) Mandal, A.; Johnson, K.; Wu, C. F. J.; Bornemeier, D. Identifying Promising Compounds in Drug Discovery: Genetic Algorithms and Some New Statistical Techniques. *J. Chem. Inf. Model.* **2007**, *47*, 981–988.

(25) Singh, J.; Ator, M. A.; Jaeger, E. P.; Allen, M. P.; Whipple, D. A.; Solowe, J. E.; Chowdhary, S.; Treasurywala, A. M. Application of Genetic Algorithms to Combinatorial Synthesis: A Computational Approach to Lead Identification and Lead Optimization. *J. Am. Chem. Soc.* **1996**, *118*, 1669–1676.

(26) Weber, L.; Wallbaum, S.; Gubernator, K.; Broger, C. Optimization of the Biological Activity of Combinatorial Compound Libraries by a Genetic Algorithm. *Angew. Chem., Int. Ed. Engl.* **1995**, *34*, 2280–2282.

(27) Yokobayashi, Y.; Ikebukuro, K.; McNiven, S.; Karube, I. Directed Evolution of Trypsin Inhibiting Peptides Using a Genetic Algorithm. *J. Chem. Soc., Perkin Trans. 1* **1996**, *20*, 2435–2437.

(28) Shanmugasundaram, V.; Maggiora, G. M.; Lajiness, M. S. Hit-Directed Nearest-Neighbor Searching. *J. Med. Chem.* **2005**, *48*, 240–248.

(29) Zheng, W.; Hung, S. T.; Saunders, J. T.; Seibel, G. L. PICCOLO: A Tool for Combinatorial Library Design Via Multicriterion Optimization. *Pac. Symp. Biocomput.* **2000**, 588–599.

(30) Rogers-Evans, M.; Alanine, A. I.; Bleicher, K. H.; Kube, D.; Schneider, G. Identification of Novel Cannabinoid Receptor Ligands Via Evolutionary De Novo Design and Rapid Parallel Synthesis. *QSAR Comb. Sci.* **2004**, *23*, 426–430.

(31) Agrafiotis, D. K. Stochastic Algorithms for Maximizing Molecular Diversity. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 841–851.

(32) Zheng, W.; Cho, S. J.; Waller, C. L.; Tropsha, A. Rational Combinatorial Library Design. 3. Simulated Annealing Guided Evaluation (SAGE) of Molecular Diversity: A Novel Computational Tool for Universal Library Design and Database Mining. *J. Chem. Inf. Comput. Sci.* **1999**, *39*, 738–746.

(33) Kirkpatrick, S.; Gelatt, C. D.; Vecchi, M. P. Optimization by Simulated Annealing. *Science* **1983**, *220*, 671–680.

(34) Metropolis, N.; Rosenbluth, A. W.; Rosenbluth, M. N.; Teller, A. H.; Teller, E. Equations of State Calculations by Fast Computing Machines. *J. Chem. Phys.* **1953**, *21*, 1087–1092.

(35) Salamon, P.; Sibani, P.; Frost, R. *Facts, Conjectures, and Improvements for Simulated Annealing (SIAM Monographs on Mathematical Modeling and Computation)*; Society for Industrial and Applied Mathematic: Philadelphia, PA, 2002.

(36) Agrafiotis, D. K.; Lobanov, V. S. Ultrafast Algorithm for Designing Focused Combinational Arrays. *J. Chem. Inf. Comput. Sci.* **2000**, *40*, 1030–1038.

(37) Le Bailly de Tilleghem, C.; Beck, B.; Boulanger, B.; Govaerts, B. A Fast Exchange Algorithm for Designing Focused Libraries in Lead Optimization. *J. Chem. Inf. Model.* **2005**, *45*, 758–767.

(38) Rechenberg, I. *Evolutionsstrategie -Optimierung technischer Systeme nach Prinzipien der biologischen Evolution*; Frommann-Holzboog: Stuttgart, 1973.

(39) Rechenberg, I. *Evolutionsstrategie′94*; Frommann-Holzboog: Stuttgart, 1994.

(40) Schwefel, H. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*; Interdisciplinary systems research; Birkhäuser Verlag: Basel, 1977, p 26.

(41) Schneider, G.; Wrede, P. The Rational Design of Amino Acid Sequences by Artificial Neural Networks and Simulated Molecular Evolution: De Novo Design of an Idealized Leader Peptidase Cleavage Site. *Biophys. J.* **1994**, *66*, 335–344.

(42) Schneider, G.; Lee, M. L.; Stahl, M.; Schneider, P. De Novo Design of Molecular Architectures by Evolutionary Assembly of Drug-Derived Building Blocks. *J. Comput.-Aided. Mol. Des.* **2000**, *14*, 487–494.

(43) Fechner, U.; Schneider, G. Flux (1): A Virtual Synthesis Scheme for Fragment-Based De Novo Design. *J. Chem. Inf. Model.* **2006**, *46*, 699–707.

(44) Eberhart, R. C.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In *Proceedings of the Sixth International Symposium on Micromachine and Human Science*; Nagoya, Japan, 1995; pp 39−43.

(45) Kennedy, J.; Eberhart, R. C. Particle Swarm Optimization. In *Proceedings of IEEE International Conference on Neural Networks*; Piscataway, NJ, 1995; pp 1942−1948.

(46) Cedeño, W.; Agrafiotis, D. K. Using Particle Swarms for the Development of QSAR Models Based on K-Nearest Neighbor and Kernel Regression. *J. Comput.-Aided. Mol. Des.* **2003**, *17*, 255–263.

(47) Shen, Q.; Jiang, J.; Jiao, C.; i Lin, W.; Shen, G.; Yu, R. Hybridized Particle Swarm Algorithm for Adaptive Structure Training of Multilayer Feed-Forward Neural Network: QSAR Studies of Bioactivity of Organic Compounds. *J. Comput. Chem.* **2004**, *25*, 1726–1735.

(48) Shen, Q.; Jiang, J.; Jiao, C.; Huan, S.; Shen, G.; Yu, R. Optimized Partition of Minimum Spanning Tree for Piecewise Modeling by Particle Swarm Algorithm. QSAR Studies of Antagonism of Angiotensin II Antagonists. *J. Chem. Inf. Comput. Sci.* **2004**, *44*, 2027–2031.

(49) Lin, W.; Jiang, J.; Shen, Q.; Shen, G.; Yu, R. Optimized Block-Wise Variable Combination by Particle Swarm Optimization for Partial Least Squares Modeling in Quantitative Structure-Activity Relationship Studies. *J. Chem. Inf. Model.* **2005**, *45*, 486–493.

(50) Meissner, M.; Schneider, G. Protein Folding Simulation by Particle Swarm Optimization. *Open Struct. Biol. J.* **2007**, *1*, 1–6.

(51) Meissner, M.; Schmuker, M.; Schneider, G. Optimized Particle Swarm Optimization (OPSO) and Its Application to Artificial Neural Network Training. *BMC Bioinformatics* **2006**, *7*, 125.

(52) Schüller, A.; Fechner, U.; Renner, S.; Franke, L.; Weber, L.; Schneider, G. A Pseudo-Ligand Approach to Virtual Screening. *Comb. Chem. High-Throughput Screening* **2006**, *9*, 359–364.

(53) Ugi, I.; Meyr, R.; Fetzer, U.; Steinbrückner, C. Versuche Mit Isonitrilen. *Angew. Chem.* **1959**, *71*, 386.

(54) Ugi, I.; Steinbrückner, C. Über Ein Neues Kondensations-Prinzip. *Angew. Chem.* **1960**, *72*, 267–268.

(55) Schneider, P.; Schneider, G. Collection of Bioactive Reference Compounds for Focused Library Design. *QSAR Comb. Sci.* **2003**, *22*, 713–718.

(56) CLIFF (Chemical Library: Interconversion of File Formats), version 1.14; Molecular Networks GmbH: Nägelsbachstrasse 25, 91052 Erlangen, Germany, 2002.

(57) Gasteiger, J.; Rudolph, C.; Sadowski, J. Automatic Generation of 3D-Atomic Coordinates for Organic Molecules. *Tetrahedron Comput. Methodol.* **1990**, *3*, 537–547.

(58) CORINA 3D Structure Generator, version 3.20; Molecular Networks GmbH: Nägelsbachstrasse 25, 91052, Erlangen, Germany, 2005.

(59) Molecular Operating Environment (MOE), version 2004.03; Chemical Computing Group: 1010 Sherbrooke St. West, #910, Montreal, Canada, H3A 2R7, 2004.

(60) Lance, C. E.; Butts, M. M.; Michels, L. C. The Sources of Four Commonly Reported Cutoff Criteria: What Did They Really Say. *Org. Res. Meth.* **2006**, *9*, 202–220.

(61) Schneider, G.; Neidhart, W.; Giller, T.; Schmid, G. "Scaffold-Hopping" by Topological Pharmacophore Search: A Contribution to Virtual Screening. *Angew. Chem., Int. Ed.* **1999**, *38*, 2894–2896.

(62) Fechner, U.; Schneider, G. Optimization of a Pharmacophore-Based Correlation Vector Descriptor for Similarity Searching. *QSAR Comb. Sci.* **2004**, *23*, 19–22.

(63) Schneider, G.; So, S. *Adaptive Systems in Drug Design*, 1st ed.; Landes Bioscience: Georgetown, 2002.

(64) Spall, J. C. *Introduction to Stochastic Search and Optimization*; John Wiley & Sons, Inc.: New York, NY, U.S.A., 2003.

(65) Clerc, M.; Kennedy, J. The Particle Swarm - Explosion, Stability, and Convergence in a Multidimensional Complex Space. *IEEE Trans. Evol. Comput.* **2002**, *6*, 58–73.

(66) Kohonen, L. *Self-Organization and Associative Memory*; Springer-Verlag: Heidelberg, 1984.

(67) Schneider, G.; Wrede, P. Artificial Neural Networks for Computer-Based Molecular Design. *Prog. Biophys. Mol. Biol.* **1998**, *70*, 175–222.

(68) Horvath, D.; Jeandenans, C. Neighborhood Behavior of in Silico Structural Spaces with Respect to in Vitro Activity Spaces - a Novel Understanding of the Molecular Similarity Principle in the Context of Multiple Receptor Binding Profiles. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 680–690.

(69) Maggiora, G. M. On Outliers and Activity Cliffs - Why QSAR Often Disappoints. *J. Chem. Inf. Model.* **2006**, *46*, 1535–1535.

(70) Peltason, L.; Bajorath, J. SAR Index: Quantifying the Nature of Structure-Activity Relationships. *J. Med. Chem.* **2007**, *50*, 5571–5578.

(71) Whitlow, M.; Arnaiz, D. O.; Buckman, B. O.; Davey, D. D.; Griedel, B.; Guilford, W. J.; Koovakkat, S. K.; Liang, A.; Mohan, R.; Phillips, G. B.; Seto, M.; Shaw, K. J.; Xu, W.; Zhao, Z.; Light, D. R.; Morrissey, M. M. Crystallographic Analysis of Potent and Selective Factor Xa Inhibitors Complexed to Bovine Trypsin. *Acta Crystallogr., Sect. D: Biol. Crystallogr.* **1999**, *55*, 1395–1404.

(72) Noeske, T.; Sasse, B. C.; Stark, H.; Parsons, C. G.; Weil, T.; Schneider, G. Predicting Compound Selectivity by Self-Organizing Maps: Cross-Activities of Metabotropic Glutamate Receptor Antagonists. *ChemMedChem* **2006**, *1*, 1066–1068.

(73) Lewin, B. *Genes VIII*, International Ed.; Pearson Prentice Hall, Pearson Education, Inc.:Upper Saddle River, NJ 07458, 2004.

(74) Schneider, G.; Schneider, P. Navigation in Chemical Space: Ligand-based Design of Focused Compound Libraries. In *Chemogenomics in Drug Discovery*; Kubinyi, H., Müller, G., Eds.; Wiley-VCH: Weinheim, 2004; pp 341−376.

CI8001205