Flux (2): Comparison of Molecular Mutation and Crossover Operators for Ligand-Based de Novo Design

Uli Fechner and Gisbert Schneider*

Johann Wolfgang Goethe-Universität, Institut für Organische Chemie und Chemische Biologie, Siesmeyerstrasse 70, D-60323 Frankfurt am Main, Germany

Received November 24, 2006

We implemented a fragment-based de novo design algorithm for a population-based optimization of molecular structures. The concept is grounded on an evolution strategy with mutation and crossover operators for structure breeding. Molecular building blocks were obtained from the pseudo-retrosynthesis of a collection of pharmacologically active compounds following the RECAP principle. The influence of mutation and crossover on the course of optimization was assessed in redesign studies using known drugs as template structures. A topological atom-pair descriptor grounded on potential pharmacophore points was used as a molecular descriptor, and the Manhattan distance between the template and candidate molecules served as a fitness function. Exclusive use of the crossover operator yielded few unique compounds and often resulted in premature convergence of the optimization process, whereas exclusive use of the mutation operator resulted in diverse high-quality structures. Combinations of crossover and mutation yielded the overall best results. The majority of the designed structures exhibit a chemically reasonable architecture; chiral centers are rare, and unfavorable connections of building blocks are infrequent. We conclude that this fragment-based design principle is suited as an idea generator for the automated design of novel leadlike molecules.

INTRODUCTION

Selection of a biological target is usually the initial step in drug discovery. This is followed by obtaining compounds (hits) that exhibit favorable binding behavior toward the particular biological target. One or more of these hits are then selected and subject to refinements. These alterations of the hit compounds aim at improving their suitability as drugs at the particular target: lowering inhibition constants, enhancing specificity, advancements with regard to their pharmacokinetic profile, and avoidance of toxicity to name just a few.[1] During this refinement phase, a compound is promoted from a mere hit to a lead candidate.

Different methods may be deployed to acquire hit compounds including high-throughput screening (HTS), virtual screening (VS), and computer-assisted de novo design. Basically, the concept of HTS and VS is to search for a suitable candidate by screening large compound libraries, thereby hoping to find a hit. In other words, each molecule of a given subset of chemical space is tested for a desired property. Computational de novo design, on the other hand, follows a different strategy to obtain hits: The search space is not restricted per se; instead, a directed path is followed during a search run for hits. Compounds along this path are tested for the respective desired property, and the outcome of these tests is used as an input to guide the proceeding search.

Each of these hit attainment approaches has its own advantages and disadvantages. Commonly, the HTS and VS screening libraries are collections of commercially available compounds and corporate databases. A search for hits is carried out on "known territory"; that is, the compounds are already known—it is the information about their binding behavior against the particular biological target that is sought. Retrieved hits are readily available or may be acquired from a vendor so that subsequent tests may be carried out rapidly. Candidate compounds suggested by a de novo design program are not readily available for further tests in the laboratory. Moreover, the synthesis might be elaborate, costly, and time-consuming—a difficulty de novo design has struggled with from its very early days.

Computational de novo design attempts to generate novel molecular structures with a desired pharmacological activity from scratch. Created candidate compounds are assessed by their match against a binding pattern of a particular biological target or set of targets. This binding pattern may be specified by either at least one known ligand (ligand-based design) or the three-dimensional (3D) structure of the target (structure-based design). Structure-based design is limited to targets where a high-resolution 3D structure is available. Its applicability thus excludes many membrane-bound drug targets including the large group of G-protein coupled receptors.[2] Protein flexibility constitutes another hassle and is currently not sufficiently tackled by structure-based de novo design approaches.[3] Notwithstanding, the deployment of structure-based methods has led to the successful generation of bioactive compounds.[4,5] Ligand-based design provides an alternative approach as it does not depend on the 3D structure of the biological target. Instead of that, the information of one or more known ligands of the particular biological target is drawn on to guide the search process. Spawn candidate compounds are evaluated by their similarity to these known reference ligands. The computation of chemical similarity relies on a meaningful choice of a descriptor and a similarity

* Corresponding author tel.: +49-69 798 24873; fax: +49-69 798 24880; e-mail: gisbert.schneider@modlab.de.

10.1021/ci6005307 CCC: $37.00
Published on Web 02/23/2007

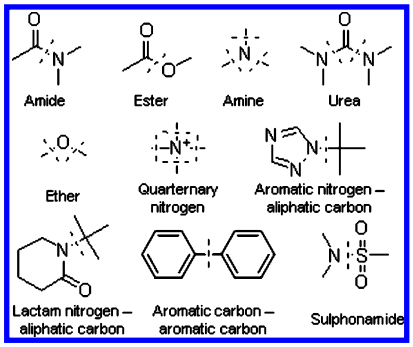

Figure 1. Ten RECAP bond cleavage types employed for (i) obtaining molecular fragments by virtual retrosynthesis of the drug data set and (ii) the virtual synthesis of these fragments to generate candidate compounds in a de novo design run of our program Flux.

coefficient.[6] More detailed information about different strategies that are being followed in the field of computational de novo design, their applicability, and their limitations can be found in a recent review.[7]

In this publication, we present the second version of our ligand-based de novo molecule generator Flux (Fragment-based Ligand bUilder reaXions). It is based on knowledge gained through the application of TOPAS[8] and our initial Flux version[9] but completely implemented from scratch. The basis of our molecule assembly concept remained unaltered: A restricted set of reaction schemes and fragments is employed to ease the chemical synthesis of proposed candidate compounds. We enriched our genetic operators by adding a new crossover operator to the already existing mutation operator. Furthermore, the second version of Flux implements an enhanced concept of our ligand-based fitness function and a universal molecular filter that facilitates the removal of compounds with undesired properties. We evaluated our molecule generator with four redesign studies using a peroxisome proliferator-activated receptor α (PPARα) agonist, a dihydrofolate reductase (DHFR) inhibitor, a 3-hydroxy-3-methylglutaryl coenzyme A (HMG-CoA) reductase inhibitor, and a 5-hydroxytryptamine 2 (5-HT_2) antagonist as templates.

METHODS

The Molecular Building Blocks. Each assembly process requires a set of building blocks. The assembling of compounds in Flux employs molecular building blocks that were yielded by virtual retrosynthesis of the COBRA data set with a limited set of reaction schemes. The COBRA data set is a collection of bioactive reference molecules for ligand-based library design compiled from recent scientific literature.[10] The data set is nonredundant and annotated by target information and activity data. We employed COBRA version 6.1, which comprises 7395 unique compounds. Our virtual retrosynthesis approach closely follows the concept of the REtrosynthetic Combinatorial Analysis Procedure (RECAP) published by Lewell and co-workers.[11] A total of 10 of the 11 RECAP bond cleavage types were applied, each of which was derived from a common chemical reaction (Figure 1). We did not include the olefin bond cleavage type because we observed in a series of preliminary design studies that its omission led to more advantageous candidate compounds (data not shown). In accordance with the original publication, ring bonds were not decomposed to leave cyclic structures intact. The original RECAP procedure prevented the application of virtual retrosynthesis, which gives rise to small fragments (including fragments up to four carbon atoms). However, we only avoided reactions that yielded single hydrogen atoms. The explicit consideration of such small fragments enriches the diversity of our building block collection in terms of size: Building blocks of different sizes allow for smaller and larger steps in chemical space, thereby equipping our molecule generator with the necessary components to adapt to variable characteristics of the fitness landscape. We selected a drug data set for fragmentation so that our building blocks are expected to be druglike as well. We anticipate that the assembly of such building blocks leads to candidate compounds that cover druglike areas of the chemical space.

Our program retroFlux takes a drug data set, applies the virtual retrosynthesis schemes, and outputs the fragmented data set. The output file of retroFlux can then be used as the stock of building blocks for the de novo design software Flux. The virtual retrosynthesis algorithm iterates over the 10 bond cleavage types. In each cycle, the current bond cleavage type is applied to all chemical entities in the data set of retroFlux: this includes uncleaved molecules as well as fragments that emerged from a virtual retrosynthesis by the same or another bond cleavage type. Resultant fragments may thus contain attachment sites produced from the application of different bond cleavage types. The deployment of one of the 10 bond cleavage types results in two, three, or four fragments, each of which bearing a new attachment site. Attachment sites were modified to maintain the information about the type of reaction that yielded it. The application of five of the 10 bond cleavage types ("amide", "ester", "aromatic nitrogen − aliphatic carbon", "lactam nitrogen − aliphatic carbon", and "sulfonamide") gives rise to fragments having different *fragment polarities* due to the "asymmetric" nature of these reaction schemes. For example, the virtual retrosynthesis of an amide bond leads to one fragment with an amine group and one fragment with a carboxyl group. Discrimination between fragment polarities is vital to carry out meaningful virtual synthesis. Compounds that could not be dissected at all were removed before saving the fragmented data set so that the stock of building blocks only holds chemical entities with at least one attachment site.

Exertion of different bond cleavage types gave rise to varied numbers of building blocks. We did not attempt to assimilate the number of building blocks per bond cleavage type by artificially changing the types of attachment sites. Our assembly concept is based on building blocks yielded by the application of virtual reaction schemes. Consistent implementation of this concept necessitates retaining the bond cleavage type whose application led to a particular attachment site. We deliberately accepted an overall smaller number of building blocks to increase the chance of synthetic feasibility.

The Molecular Mutation Operator. The molecular mutation operator algorithm starts by randomly selecting one of the available bond cleavage types. The remaining procedure can be subdivided into a *retrosynthesis part* and a subsequent *synthesis part*. In the retrosynthesis part, a given parental structure is exhaustively dissected by the chosen bond cleavage type, thereby building up a *reaction tree*

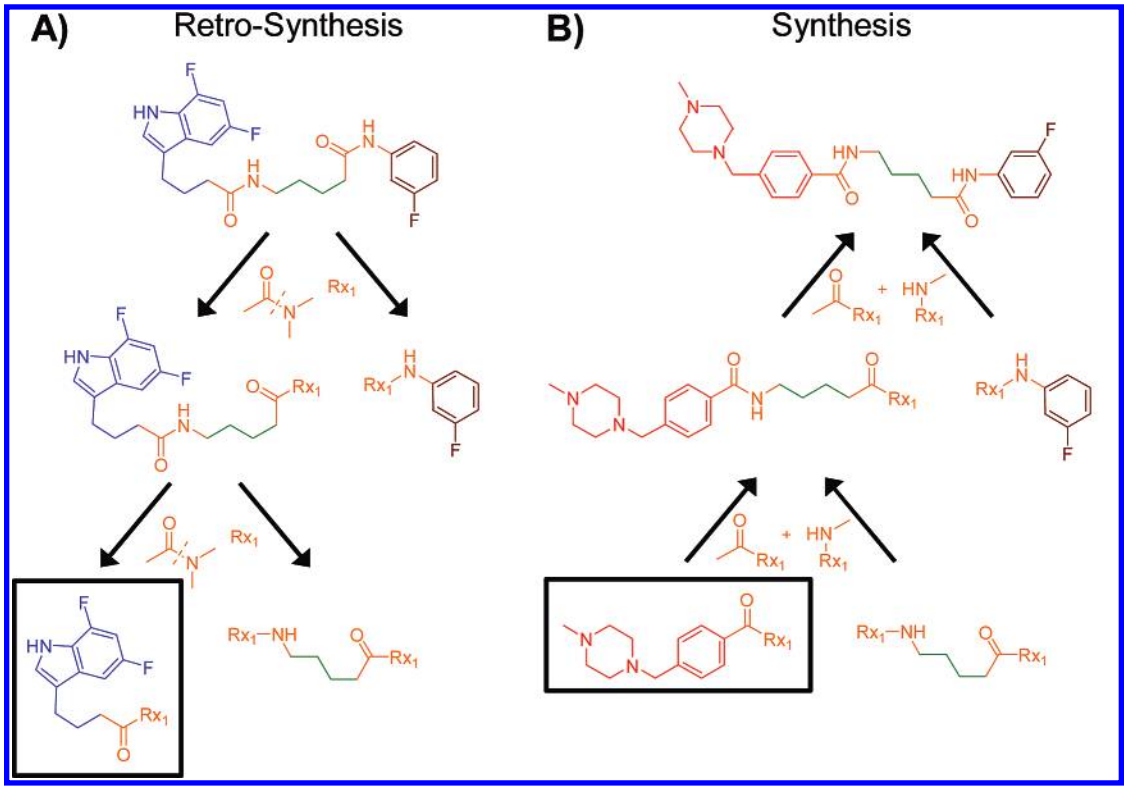

Figure 2. Exemplary application of the mutation operator. (A) Virtual retrosynthesis. The parental structure (tree root) is exhaustively dissected by virtual retrosynthesis using the amide bond cleavage type. Two applications of the amide reaction scheme lead to a reaction tree consisting of three levels and three leaves (gray rectangles). One leaf is randomly selected (rectangle with block border). (B) Virtual synthesis. The selected leaf is exchanged for a suitable building block from the stock. The child structure is then assembled by walking up the tree until the root is reached. Virtual synthesis is carried out en route using the same reaction scheme that was employed during creation of the reaction tree.

(Figure 2a). The root node of this tree is the parental structure; nonleaf nodes are intermediate fragments that can be further dissected, and leaf nodes are fragments that cannot be dissected any further using the selected bond cleavage type. Contingent upon the bond cleavage type and the particular molecule or fragment, successful application of a virtual retrosynthesis may give rise to two, three, or four child nodes. Creation of the reaction tree is followed by random selection of a leaf node. The retrosynthesis part is then concluded by substituting the fragment of this leaf node for a suitable fragment from the stock of building blocks. In the synthesis part, the child structure is synthesized by climbing up the reaction tree node-by-node starting from the leaf node whose fragment had been exchanged (Figure 2b). Fragments of sibling nodes (nodes having the same parent) play the role of educts in a virtual synthesis. The product of these reactions is associated with the parent node.

The Molecular Crossover Operator. The molecular crossover operator takes two parental structures and spawns two child structures. First, the algorithm randomly selects one of the available bond cleavage types. It is then assessed whether a virtual retrosynthesis using the chosen bond cleavage type can be carried out at least once for both parental structures. If this holds true, the algorithm proceeds; otherwise, another bond cleavage type is randomly selected until one is found whose application to both parental structures gives rise to products. It may well happen that two parental structures cannot be dissected by virtual retrosynthesis using the same reaction scheme. In this case, the algorithm terminates without generating offspring.

In analogy to the molecular mutation operator, selection of a suitable bond cleavage type is followed by a *retrosynthesis part* and a *synthesis part*. However, in contrast to the mutation operator, the crossover operator deals with two reaction trees in parallel: one for each parental structure. The retrosynthesis part starts with the exhaustive dissection of both parental structures thereby creating two corresponding reaction trees (Figure 3a). Then, a tree level between 2 and the maximum depth common to both reaction trees is randomly selected. A node on this level in one of the reaction trees is randomly chosen and substituted with a node on the same level of the other reaction tree. For such a node swap to take place both fragments have to share the same *polarity* (vide supra). The synthesis part is then identical to the mutation operator, the difference being that it is carried out twice: two child structures are virtually synthesized by walking up each of the two reaction trees starting from the node whose associated fragment has been exchanged (Figure 3b).

Growth and Shrinkage of Molecules. Application of the molecular mutation and crossover operator may lead to offspring structures with a different size than the original structure they emerged from (parental structure). Basically, this is caused by exchanging a node fragment after the creation of a reaction tree. More precisely, two cases can be discriminated. First, a substitute fragment may be of different size than the fragment it was substituted for. The second case may lead to a more significant size difference between a parental and an offspring structure. It occurs if the number of attachment sites of the original node fragment and its

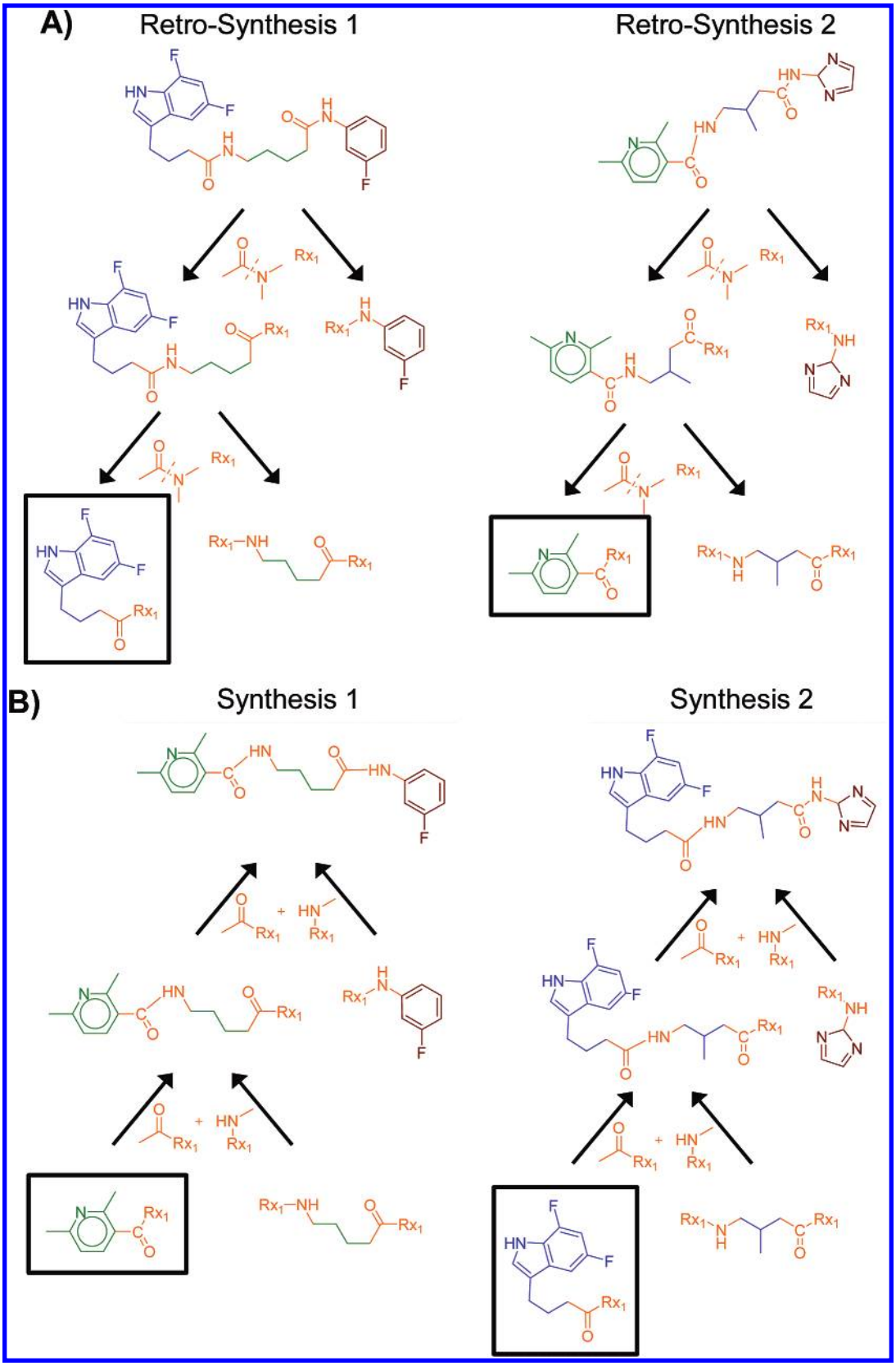

Figure 3. Illustration of the crossover operator. (A) Virtual retrosynthesis. The two parental structures (gray rectangles) are exhaustively dissected using the amide bond cleavage type. Two reaction trees are created. Then, a node fragment of one reaction tree is swapped for a node fragment on the same tree level of the other reaction tree (rectangles with black borders). (B) Virtual synthesis. Starting from the changed nodes, both reaction trees are synthesized resulting in two child structures (gray rectangles).

substitute fragment are not equal. If a substitute fragment has fewer attachment sites than the original fragment, synthesis of the reaction tree cannot be carried out until the root node is reached but terminates prematurely. This results in an offspring structure that is smaller than its parental structure (Figure 4a). On the other hand, a substitute fragment with more attachment sites than the original fragment leads to a completely synthesized reaction tree whose root node

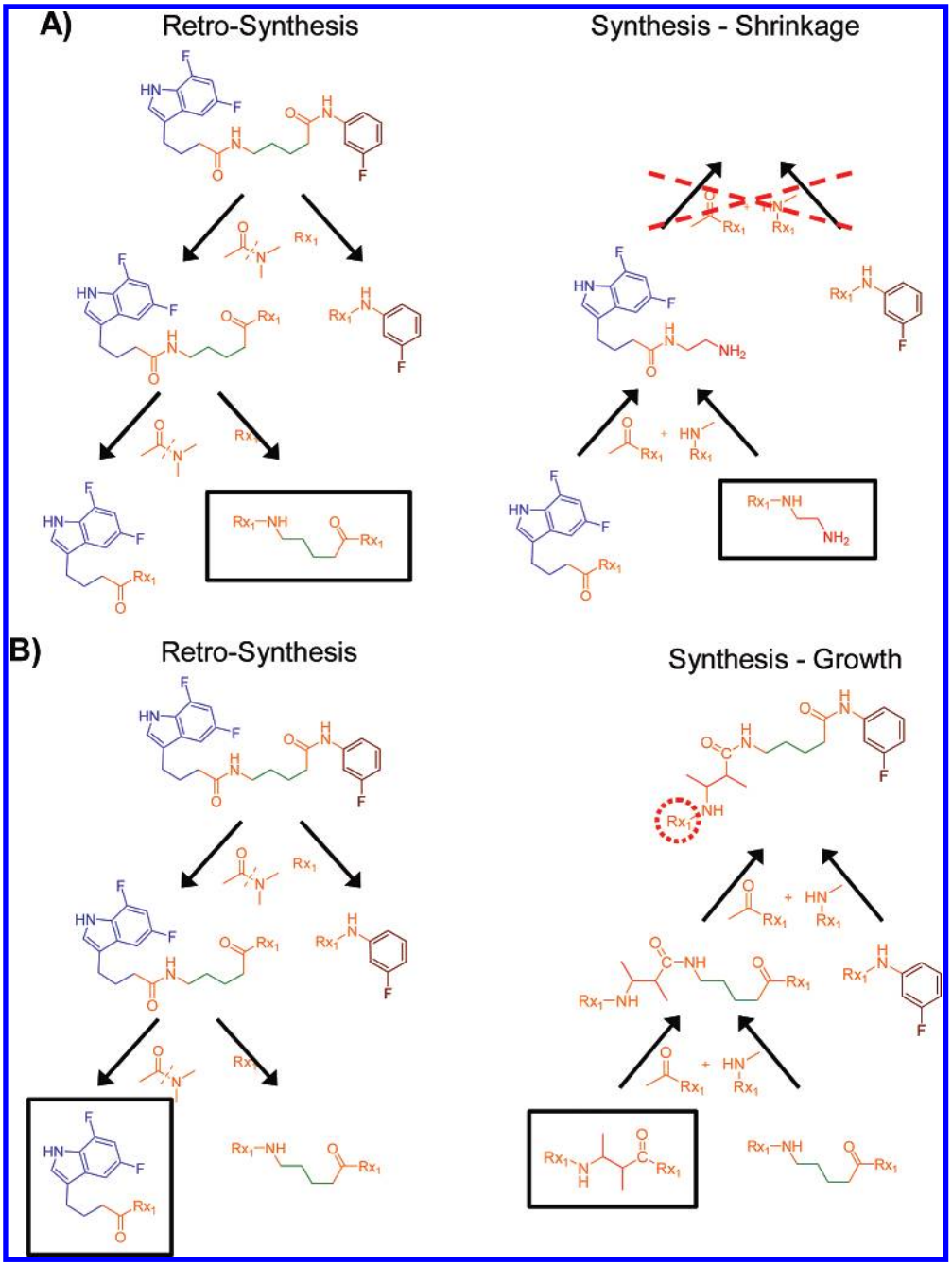

Figure 4. Depiction of two reaction trees to demonstrate the growth and shrinkage of molecules caused by the application of a molecular genetic operator. (A) Shrinkage of a molecule. The left reaction tree (Retro-Synthesis) shows the exhaustive virtual retrosynthesis of a parental structure (gray rectangle) by the amide reaction scheme. The randomly selected leaf node whose fragment is exchanged is marked by a rectangle with a black border. In the reaction tree on the right side (Synthesis − Shrinkage), the selected leaf node contains a substitute fragment. Whereas the original fragment of this leaf node contains two attachment sites, the substitute fragment only contains a single attachment site. The application of virtual synthesis results in a product (gray rectangle) not containing any attachment site. Therefore, no further virtual synthesis can be carried, out and this product is the final child structure. (B) Growth of a molecule. Creation of the left reaction tree (Retro-Synthesis) results from exhaustive dissection of the parental structure (gray rectangle) using the amide bond cleavage type. The fragment of the randomly selected leaf node (rectangle with black border) contains one attachment site. It is exchanged for a fragment with two attachment sites. Starting from the leaf node associated with the substitute fragment, the reaction tree is climbed up until the root node is reached (Synthesis − Growth). A virtual synthesis is applied at each tree level. The root node fragment (gray rectangle) still contains an attachment site (dotted red circle). Thus, the *attachment site saturation algorithm* (see text) is initiated, providing this fragment as an input.

has one or even more unsaturated attachment sites (Figure 4b). In this case, the *attachment site saturation algorithm* (vide infra) is executed, and the offspring structure exhibits growth compared to its parental structure.

The Attachment Site Saturation Algorithm. The *attachment site saturation algorithm* takes a molecule with at least one unsaturated attachment site as an input. It iterates over the atoms and stops at the first attachment site. Given the attachment site, the corresponding reaction scheme is determined. Depending on the reaction scheme, one, two, or three suitable building blocks are randomly selected from the stock. Then, a virtual synthesis is carried out using the molecule and the building block(s) as educts. This procedure is repeated until the molecule contains no further unsaturated

attachment sites. It is noteworthy that a randomly chosen building block from the stock may contain more than one attachment site, thereby contributing to further growth of the molecule.

Generation of a Random Molecule. Generation of a random molecule starts by randomly selecting a building block from the stock. All building blocks contain at least one attachment site. Thus, the *attachment site saturation algorithm* is executed providing the randomly chosen fragment as an input. After termination of the *attachment site saturation algorithm*, a chemically valid molecule is obtained.

Evolutionary Strategy Optimization. The search space of a de novo design campaign that aims for the generation of a drug is given by all compounds with a druglike weight that are theoretically possible. It is estimated that this comprises between 10^{60} and 10^{100} chemical structures.[12] Therefore, a simple enumeration of all candidate compounds is not feasible. Rather, efficient optimization algorithms are required to navigate through chemical space toward regions that are populated by suitable candidate compounds. We deployed an evolution strategy (ES) algorithm to this optimization task.

ES algorithms are grounded on the ideas published by Charles Darwin in 1859.[13] They are nondeterministic search algorithms that simulate the evolutionary pressure of selection and the evolutionary operators crossover and mutation. Deployment of an ES algorithm to an optimization task aims at finding the global optimum. In real-world applications, however, this target is rarely matched, and commonly one ends up finding a local optimum. We implemented a simplistic (μ, λ) ES omitting an adaptive-step size control.[14,15] The μ in the (μ, λ) ES indicates that the μ fittest individuals of one generation are selected as parents that produce offspring. λ specifies the number of children per generation. We did not implement an elitism strategy; that is, a parent of the current generation is not allowed to be part of the next generation. Enhancements of our basic ES algorithm will be implemented and subjected to de novo design applications in future work.

The basic ES algorithm starts at an arbitrary point in search space by assembling a random molecular structure (Figure 5). This structure is then used as a parent to spawn λ offspring—the initial population—by means of the molecular mutation operator. All individuals of the initial population are evaluated by our ligand-based fitness function. The fittest μ individuals are then selected as parents for the next generation. Application of the molecular crossover and mutation operator to the parental structures yields the offspring that make up the next generation. If identical offspring structures are spawned within the same generation, only one of them is kept. This can lead to a smaller number of offspring for a generation than λ. We did not enforce breeding of λ offspring structures due to limited numbers of building blocks for certain bond cleavage types. This sequence (breeding of a population by means of the mutation and crossover operators, evaluation of this population by our fitness function, and selection of the fittest μ parental structures) is repeated until a termination criterion holds true (Figure 5).

Two global parameter values have to be specified for our implementation of the ES algorithm: the number of parents per generation (μ) and the number of children per generation (λ), which were set to 10 and 100, respectively. Preliminary experiments led to the observation that our ES algorithm converged after approximately 50 generations (data not shown). To ensure proper convergence, we stopped the optimization after 75 generations and did not regard any additional termination criteria. As we did not apply an elitism strategy, our algorithm does not guarantee that the parental structure of the current generation is superior to parental structures of previous generations in terms of their fitness values. A comprehensive analysis of candidate compounds generated in a run of our de novo design program Flux may therefore include the parental structures of all generations.

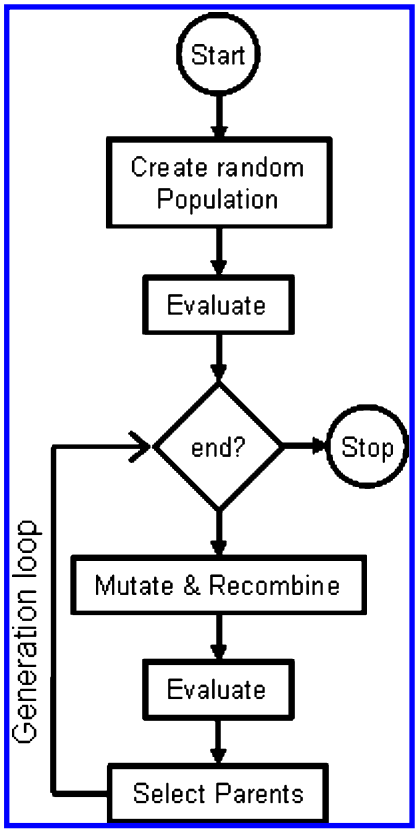

Figure 5. Flowchart of the evolution strategy (ES) optimization algorithm.

The Fitness Function. We assessed the potential suitability of candidate compounds by their similarity to a known reference molecule (template structure). Computation of such pairwise chemical similarity is based on (i) a chemical descriptor that spans a chemical space and (ii) a metric that defines distance in this space. The Flux software makes extensive use of the Chemistry Development Toolkit[16] (CDK) as a cheminformatics library. Every descriptor that implements the Java interface for CDK descriptors can be utilized for the chemical similarity calculations; this includes the 69 descriptors available in CDK at the time of writing. The Manhattan distance and the Tanimoto index are selectable as similarity metrics. The fitness function of Flux is flexibly configurable in accordance with the requirements. Basically, a fitness function in Flux is a weighted sum. Each term of such a sum is made up of a weighting factor, a template structure, and a descriptor. Weighting factors may also be negative. A combination of a positive weighting factor for one template structure and a negative weighting factor for

another one may allow for the targeting of particular receptor subtypes.

In this study, we employed a basic fitness function that was composed of a single template structure with a weighting factor of 1.0. The Chemically Advanced Template Search (CATS) descriptor was used to encode structures in a high-dimensional correlation vector.[17] It is a pharmacophore-based atom-pair descriptor whose distance calculations are topological, thereby avoiding the necessity of generating three-dimensional coordinates of molecules. Computation of the CATS descriptor may be configured by two parameters: the maximum distance in bonds between atom pairs that are considered and the type of scaling applied to the correlation vector. We employed standard parameters that were shown to be effective on average (maximum distance of 10 bonds, individual scaling of the pharmacophore type pairs).[18] Among a variety of distance metrics commonly used in cheminformatics, we chose the Manhattan distance. This decision arose from a study that scrutinized the interplay between several distance metrics and the CATS descriptor.[19] The Manhattan distance $D_{A,B}$ of two molecules A and B is given by

$$D_{A,B} = |\sum_{j=1}^{j=n} x_{jA} - x_{jB}| \quad (1)$$

where n is the total number of attributes of an object and x_{jA} the value of the jth attribute of object A. Results of the Manhattan distance are in $[0;+\infty]$, where zero refers to identity.

Molecular Filter. We implemented a molecular filter that facilitates the removal of candidate compounds that exhibit undesired characteristics. The molecular filter is composed of individual filters, each of which is based on a chemical descriptor with lower and upper boundary values. If the descriptor value of a given candidate compound is below or above the specified threshold values, the respective filtering criterion is violated. The user can set a maximum number of filters that may be violated by any compound ($\text{violation}_{\text{max}}$); the removal of a candidate compound takes place if more individual filters are violated than $\text{violation}_{\text{max}}$. Additionally, molecular properties that are considered to be critical may be enforced: a filter may be marked by a flag whose presence leads to the immediate removal of a candidate compound if the particular filter is violated—independent of the total number of individual filters that are violated by this candidate compound.

The molecular filter avails both descriptors that are present in CDK and other descriptors that implement the Java interface for CDK descriptors. Apart from this single restriction, the molecular filter is freely configurable according to the requirements of a particular de novo design task.

For this study, we configured a molecular filter that comprises a single filter. Candidate compounds that exhibit a molecular weight below 200 Da or above 750 Da were removed. The molecular filter removes structures after the creation of a generation, thereby potentially diminishing the number of its individuals. As a consequence, in Flux, an ES generation may contain less than λ children not only because of the breeding of identical offspring structures but also because of the application of the molecular filter.

Implementation Details. Both retroFlux and Flux were written in Java (developed by Sun Microsystems, Inc.; http://java.sun.com) and are based on a several Java libraries. CDK[16] is employed as a cheminformatics toolkit (http://cdk.sf.net). The input and output of chemical structures in SDF format[20] and the basic functionality for all structure manipulation are provided by CDK. Compounds are tested for uniqueness by means of canonical SMILES[21] generated by CDK. Our ligand-based fitness function and molecule filter are based on descriptors that implement the Java interface for descriptors defined by CDK.

The execution of retroFlux generates a binary file that contains the building blocks yielded by an exhaustive retrosynthesis of a given compound data set. The user can choose between two binary formats: The first one necessitates that all compounds are held in random access memory (RAM); the binary file consists of CDK molecule objects that are written to a file using Java's serialization mechanism. If the memory requirements of the complete stock of building blocks exceed the available RAM, the user may choose to use a structured query language (SQL) database for storage. We employed an HSQLDB to avoid the hassle of installing and maintaining an SQL database server. HSQLDB implements a SQL relational database engine in Java and comes with a corresponding Java Database Connectivity driver (http://hsqldb.org). The database engine of HSQLDB can be configured to run in the same Java virtual machine as the application that executes SQL queries. This so-called *in-process mode* of HSQLDB requires no configuration efforts from the user of the program; moreover, the *in-process mode* offers the additional advantage that SQL queries and their results are not sent over the network, thus potentially offering a speed increase for database scenarios that are characterized by many queries and small result sets.

The execution of a Flux run requires a comprehensive configuration file that specifies the name of the building blocks and output file, the virtual synthesis schemes, parameters of the optimization algorithm, the ligand-based fitness function and its reference structure(s), and details of the molecular filter that removes structures exhibiting undesired characteristics. Configuration files are in XML format to increase their readability and facilitate their parsing. The XML files are read and written with the aid of the Java Document Object Model library (http://jdom.org). Plots of fitness values over the number of generation are automatically created using JFreeChart (http://www.jfree.org/jfreechart). The CATS descriptor was computed by an in-house Java implementation. The optimization algorithm ES is implemented in a Java library termed "Cyclops" (developed by A. Schüller, University of Frankfurt, unpublished).

RESULTS AND DISCUSSION

Utilizing the experience gained in the implementation and application of the initial version of our ligand-based de novo design program Flux,[9] we developed a second refined version from scratch. The building blocks of Flux are obtained by the virtual retrosynthesis of a drug data set with a set of 10 bond cleavage types. The same set of reaction schemes was then employed to connect fragments to each other so that chemically valid candidate compounds are assembled. The potential binding affinity of candidate compounds is estimated by the chemical similarity between a known ligand

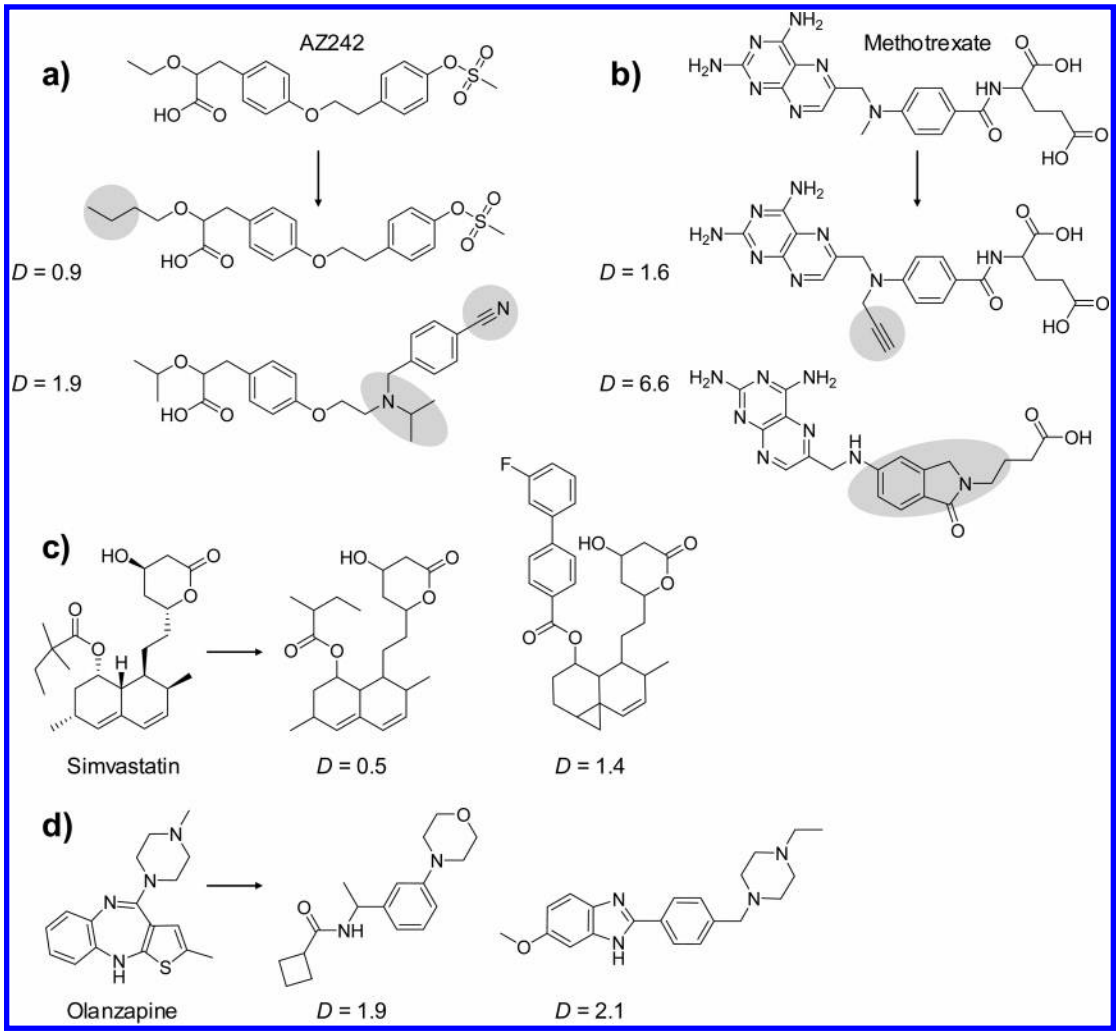

Figure 6. Examples of de novo designed molecules. In a and b, differences from the template structures are highlighted.

against the biological target of interest and the candidate compounds. This chemical similarity is based on the topological pharmacophore descriptor CATS and the Manhattan distance. A molecular filter was applied that removed candidate compounds exhibiting a molecular weight below 200 Da and above 750 Da. The principal efficiency of this second version of our molecule generator Flux was investigated by retrospective design studies against four different biological targets: we selected a PPARα agonist, a DHFR inhibitor, a HMG-CoA reductase inhibitor, and a 5-HT_2 antagonist as reference molecules. More specifically, the reference structures AZ242 (PPARα antagonist), methotrexate (DHFR inhibitor), simvastatin (HMG CoA reductase inhibitor), and olanzapine (5-HT_2 antagonist) were employed (structures are depicted in Figure 6 together with selected candidate compounds).

Virtual retrosynthesis of the 7395 unique compounds of the COBRA version 6.1 data set[10] by our program retroFlux resulted in 6156 unique fragments with at least one attachment site. A total of 2682 compounds of the COBRA data set were not dissected at all by any of the 10 bond cleavage types. These uncleaved compounds without any attachment site were removed from the output of retroFlux so that the set of 6156 fragments with at least one attachment site made up the stock of building blocks for the subsequent de novo design exercises. Figure 7 shows the distribution of the molecular weight of the stock of building blocks by means of a box plot. The median of the molecular weight is approximately 200 Da; the 25th and 75th percentiles are about 140 and 290 Da, respectively. The box plot also discloses that the weight of building blocks with higher

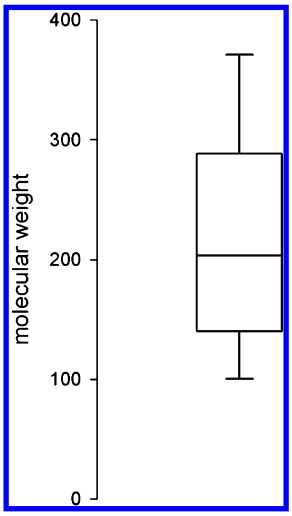

Figure 7. Box plot showing the molecular weight distribution of the building blocks yielded by virtual retrosynthesis of the COBRA drug data set using the bond cleavage types depicted in Figure 1. The line within the box indicates the median; the lower boundary of the box marks the 25th percentile, and the upper boundary gives the 75th percentile. Whiskers mark the 90th and 10th percentiles.

Table 1. The Number of Retrosynthesis Reactions and the Number of Resultant Unique Fragments That Were Yielded by Application of the Program RetroFlux to the COBRA Drug Data Set[a]

bond cleavage type	number of retrosynthesis reactions	number of unique fragments
amide	3194	2695
ester	879	853
amine	477	853
urea	231	293
ether	132	164
quarternary nitrogen	17	20
aromatic nitrogen−aliphatic carbon	531	634
lactam nitrogen−aliphatic carbon	379	477
aromatic carbon−aromatic carbon	1024	929
sulfonamide	508	598

[a] We applied 10 of the 11 RECAP bond cleavage types; the olefin bond cleavage type was omitted.

molecular weights are more scattered than those with lower ones. The mean of the molecular weight of all building blocks is 223 Da. Consequently, candidate compounds with a typical druglike weight of 500 Da are composed of 2.24 building blocks on the average.

Table 1 lists the frequency of virtual retrosyntheses being conducted and the number of unique fragments for each of the 10 bond cleavage types. The amide reaction was by far the most frequently carried out (3194), followed by the aromatic carbon−aromatic carbon reaction (1024), the ester reaction (879), the aromatic nitrogen−aliphatic carbon reaction (531), the sulfonamide reaction (508), and the amine reaction (471). Save for a few exceptions, the order of the frequency of virtual retrosynthesis execution corresponds to the order of the number of unique fragments generated by a particular bond cleavage type. The majority of fragments contained a single attachment site (4626), but we also obtained 1417 bivalent fragments, 102 fragments with three attachment sites, and 11 fragments with four or more attachment sites.

We then carried out de novo design runs using the four template structures that bind to 5-HT_2, DHFR, HMG, and PPARα. The parameters of the optimization algorithm, the ligand-based fitness function, and the molecular filter are described in the Methods section (vide supra). To facilitate a comparison between the molecular mutation operator and the molecular crossover operator, five different genetic operator configurations were applied to each template structure. The percentage of children of each generation that were bred by the mutation operator was gradually decreased by 25% starting from 100%. Children that were not spawned by the molecular mutation operator were generated by application of the molecular crossover operator. In other words, the first genetic operator configuration created 100% of the children of one generation using the molecular mutation operator and 0% using the molecular crossover operator (M100%;C0%). The second genetic operator configuration generated 75% of the children of one generation by usage of the molecular mutation operator and 25% by application of the molecular crossover operator (M75%;C25%). The third, fourth, and fifth genetic operator configurations are abbreviated by M50%;C50%, M25%;C75%, and M0%;C100%, respectively. The four template structures and the five genetic operator configurations made up 20 different Flux configurations. For each such Flux configuration, 100 runs of Flux were carried out.

Table 2. Molecular Weight of the Four Template Structures and the Average Molecular Weight and Corresponding Standard Deviation of the 50 Fittest Candidate Compounds Generated by Flux[a]

		50 fittest candidate compounds	
template	template structure	mean	standard deviation
5-HT_2	312	336	112
DHFR	454	579	100
HMG	418	554	137
PPARα	408	561	128

[a] The 50 fittest candidate compounds were selected irrespective of the particular genetic operator configuration (refer to the text for more details).

Table 3. Average Number of Building Blocks (BBs) and the Number of Expected Building Blocks (Expected BBs) for Reference Structures Binding to 5-HT_2, DHFR, HMG, and PPARα[a]

template	BBs	expected BBs
5-HT_2	2.32	1.51
DHFR	4.02	2.60
HMG	2.74	2.48
PPARα	3.68	2.51

[a] The 50 fittest candidate compounds independent from the particular genetic operator configuration were taken for the calculation of the mean (refer to the text for more details).

First of all, we analyzed the 50 fittest candidate compounds bred by each Flux configuration in terms of their molecular weight. Given a particular reference structure, we observed no significant differences of the average molecular weight and corresponding standard deviation between the five genetic operator configurations (data not shown). Therefore, we compiled a set of 50 candidate compounds for each reference structure; these 50 compounds exhibited the *overall* best fitness values among the set of 250 candidate compounds generated by the five different genetic operator configurations. On average, these 50 fittest candidate compounds have a molecular weight that is 110 Da higher than the molecular weight of the respective template structure (Table 2). Even so, the average molecular weight is still in a suitable range for an orally administered drug. In particular, the relatively low molecular weight of the 5-HT_2 reference ligand (312) is reflected in the average molecular weight of 336 of the designs.

Then, given a particular biological target, we scrutinized the 50 fittest candidate compounds generated by the five genetic operator configurations by means of the number of building blocks they were assembled from. Again, no substantial differences were observable between the individual genetic operator configurations (data not shown). Consequently—and as already done in the molecular weight analysis—we extracted the *overall* 50 fittest candidate compounds designed against a specific target from the set of 250 candidate compounds generated by the five different genetic operator configurations. Table 3 lists their average number of building blocks together with the expected number of building blocks. The expected number of building blocks results from a division of the average molecular weight of the 50 fittest candidate compounds by the average molecular weight of a building block (223). The average number of

Table 4. Number of Unique Candidate Compounds Generated by Flux for Reference Structures Binding to 5-HT_2, DHFR, HMG, and PPARα[a]

template	M100%;C0%	M75%;C25%	M50%;C50%	M25%;C75%	M0%;C100%
5-HT_2	334 695	216 639	164 620	101 330	5742
DHFR	335 449	265 645	187 062	111 348	6987
HMG	266 630	215 199	165 514	103 672	7036
PPARα	219 804	190 568	151 516	96 307	5530

[a] Individual columns refer to different percentage compositions of the molecular mutation and crossover operator. For instance, the column labeled M25%;C75% indicates a configuration where 25% of the children were spawned by application of the molecular mutation operator (M) and 75% of the children were created by usage of the molecular crossover operator (C). The entirety of candidate compounds generated by 100 runs of our de novo design software Flux was considered.

building blocks is higher than the expected number of building blocks for all four template structures. This may be taken as a hint that a larger set of small building blocks might be beneficial for the assembly of candidate compounds that are more similar to their reference structures in terms of molecular weight and composition. Certainly, a more refined stock of building blocks can be obtained from adding retrosynthesis schemes to the retroFlux procedure, for example, Michael reactions, Diels−Alder reaction schemes, Suzuki-type coupling, and others.

Basically, a genetic mutation operator introduces new information into a population, whereas a crossover operator recombines existing information. Our results corroborate this fact. Table 4 shows the number of unique compounds that were generated by the 100 runs of Flux for each particular combination of a template structure and a genetic operator configuration. Clearly, the more candidate compounds were spawned by the molecular crossover operator, the less unique the compounds obtained were. This decrease of unique compounds with an increasing usage of the crossover operator can be consistently observed for all four template structures. In particular, the comparably small number of unique compounds yielded by exclusive use of the crossover operator substantiates the fact that the crossover operator does not insert new information into an existing population.

We also observed that the course of optimization is influenced by the ratio of crossover versus mutation events. Independent from the particular biological target, our results clearly point out that 100% crossover did not yield optimal fitness values within the limits of our experiment. The mere use of recombining existing information—that is, using crossover as the only genetic operator—can only lead to successful optimization if optimal molecular building blocks are already present in the initial population. In our original design software TOPAS,[8] a (μ, λ) evolution strategy[14] was implemented that solely employed the molecular mutation operator. In this study, we acquired close-to-optimal results with Flux when using 100% mutation events, thereby corroborating the initial encouraging results obtained by TOPAS. On the basis of our results, a crossover versus mutation ratio of 50:50 seems to be a reasonable choice, as this led to best average fitness values for the majority of our design experiments. For the sake of conciseness, we are not able to show the course of optimization for all 20 combinations that arise from different crossover versus mutation ratios and reference structures. Instead, the design runs with the 5-HT_2 antagonist olanzapine serve as an example to demonstrate the inferiority of the sole usage of the crossover operator (Figure 8a), the close-to-optimal results of the exclusive deployment of the mutation operator (Figure 8c), and the superiority of a balanced crossover versus mutation ratio (Figure 8b).

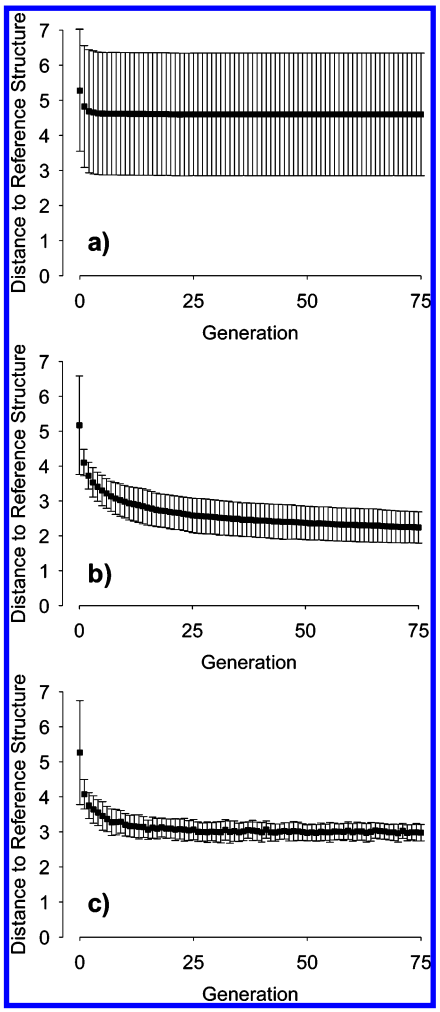

Figure 8. Average course of fitness values obtained from 100 design runs using the 5-HT_2 antagonist olanzapine as a template structure. Error bars give RMSD values. Mutation/crossover ratios were 0:100 (a), 50:50 (b), and 100:0 (c).

Figure 6 shows example designs that were generated by Flux; AZ242 (PPARα antagonist), methotrexate (DHFR inhibitor), simvastatin (HMG CoA reductase inhibitor), and olanzapine (5-HT_2 antagonist) were employed as template structures. The depicted candidate compounds were selected from the overall 50 fittest structures against a particular biological target, that is, the 50 fittest structures irrespective of the applied crossover versus mutation ratio. In the present study, the evaluation of Flux is grounded on its ability to perform redesign, that is, generating the template structure from the available building blocks.

In all four redesign attempts, very similar and several more distantly related structures were assembled. Note that the absolute distance values of the designs to their templates cannot be compared, which is due to the differences of the template structures.[22,23] With the exception of olanzapine, Flux succeeded in recreating the original molecule. This failure is explainable by the limited building block diversity and the fact that olanzapine is dissected only once by the RECAP rules. Figure 6 shows two designs for each template, one rather close to the reference structure and one more distant. The majority of the designed structures have a chemically reasonable architecture, and unfavorable connections of building blocks are infrequent. Still, this does not mean that the generated compounds are easily synthesizable. Compared to entirely atom-based molecule generators, however, the fragment-based concept implemented in Flux has a higher chance of coming up with feasible structures.

Overall, the generated molecules contain few stereocenters. This, however, is not inherent in the structure assembly algorithm of Flux but is determined by the number of stereocenters of the building blocks and the selection of the reaction schemes. Almost all of the employed reaction schemes do not create stereocenters. Moreover, the reactions that actually do create stereocenters will only introduce chirality if it was already present at those locations. The building blocks were obtained by virtual retrosynthesis of a drug data set. Typically, stereocenters are rare in drugs. It is our choice of reaction schemes and building blocks that leads to the rare presence of stereocenters in candidate compounds.

Several studies demonstrated that fragment-based de novo design is able to breed novel molecular entities that exhibit a desired binding behavior.[7,8,24,25] This, however, should not be regarded as the general outcome of a de novo design exercise, for it might be aimed too high. The automatically produced structures of a de novo design program cannot be expected to directly deliver new lead compounds. Rather, candidate compounds are able to induce "molecular ideas" that enrich a particular drug discovery project. A qualified medicinal chemist (the "expert") who is familiar with the requirements of a certain biological target and other factors influencing lead selection should be expected to judge better than a computer-based scoring function if it comes to the suitability assessment of candidate compounds. Such a human assessment is not only based on rational arguments but also on a "gut feeling" (as a part of expert knowledge) that cannot be modeled by algorithmic concepts. A recent de novo design program—the Molecule Evoluator[26]—directly takes advantage of the capabilities of medicinal chemists: it implements the concept of interactive evolution; that is, the user acts as a fitness function. The aim of computational de novo design is the combination of fast generation of candidate compounds by the computer and domain knowledge of medicinal chemists that brings out the best of both.

A single run of the first version of our de novo design software Flux[9] took approximately 150 min on a 2.4 GHz Intel Celeron with 2 GB of RAM. We regarded this time as an impediment for high-throughput de novo design. The second version of Flux presented in this work was completely written from scratch. One aim of the revised version was a major reduction of computation time, and we took great care to identify and optimize performance bottlenecks in the software. We were able to reduce the time for a single run on the same machine that was used for computations with the first version of Flux to approximately 30 min. In our view, this cutback of computation time by about 80% allows for a deployment of Flux to high-throughput de novo design and overcomes the obstacle formed by the first program version.

Our optimization algorithm—a nonadaptive evolution strategy—is a single algorithm from a plethora of optimization algorithms. The lack of adaptive parameters (e.g., step-size control) may make it difficult to escape local optima in search space so that resultant candidate compounds are suboptimal in terms of their fitness. It has not yet been clarified how rugged the fitness landscape de novo design has to deal with actually is.[27] Neither has the question been addressed to what extent local optima influence the outcome of a de novo design run.[28] There are optimization algorithms—for instance, adaptive evolution strategies,[15] particle swarm optimization,[29] and optimized particle swarm optimization[30]—that are considered to be more sophisticated and consequently more successful than the simplistic algorithm used in this research study. In particular, such elaborate optimization methods diminish the risk of premature convergence and facilitate to escape local optima. This work demonstrated that our optimization strategy was sufficient to reproduce the original template for three of the four template structures. In the case of the 5-HT_2 template, however, the optimization process terminated prior to reaching optimal fitness values. We are aware of this limitation in the current version of our de novo design software Flux. Subsequent research efforts will be directed at the implementation of additional optimization algorithms and the comparison thereof. Additionally, the freely configurable ligand-based fitness function and molecular filter provide potentialities that might be worthwhile following up. Only recently, a dynamic programming method has been proposed for structure-based de novo design,[31] which might also be applicable to ligand-based methods and represent an alternative to stochastic sampling.

The ultimate validation of a de novo design program is a prospective design study: a handpicked selection of the proposed candidate compounds is synthesized, and their biological activity is measured in the laboratory. We have subjected Flux to this validation procedure already (ongoing studies), and the outcome of such a prospective design study will be reported in a forthcoming publication.

ACKNOWLEDGMENT

The authors are grateful to Norbert Dichter for setting up the LSF Linux cluster and to Dr. Petra Schneider for compiling the COBRA data set. Andreas Schüller implemented the Java-based optimization library Cyclops. Dr. Karl-Heinz Baringhaus is thanked for valuable discussion of various aspects of de novo design techniques and comments on the Flux implementation. U.F. is thankful for a Ph.D. fellowship granted by Sanofi-Aventis Pharma Deutschland GmbH and Johann Wolfgang Goethe-University. This research was supported by the Deutsche Forschungsgemeinschaft (SFB 579, project A11) and the Beilstein-Institut zur Förderung der Chemischen Wissenschaften, Frankfurt am Main.

REFERENCES AND NOTES

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

(2) Lowrie, J. F.; Delisle, R. K.; Hobbs, D. W.; Diller, D. J. The Different Strategies for Designing GPCR and Kinase Targeted Libraries. *Comb. Chem. High Throughput Screening* **2004**, *7*, 495−510.
(3) Zhu, J.; Fan, H.; Liu, H.; Shi, Y. Structure-Based Ligand Design for Flexible Proteins: Application of New F-DycoBlock. *J. Comput.-Aided Mol. Des.* **2001**, *15*, 979−996.
(4) Böhm, H.-J. Computational Tools for Structure-Based Ligand Design. *Prog. Biophys. Mol. Biol.* **1996**, *66*, 197−220.
(5) Congreve, M.; Murray, C. W.; Blundell, T. L. Structural Biology and Drug Discovery. *Drug Discovery Today* **2005**, *10*, 895−907.
(6) Willett, P. Chemical Similarity Searching. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 983−996.
(7) Schneider, G.; Fechner, U. Computer-Based *de Novo* Design of Druglike Molecules. *Nat. Rev. Drug Discovery* **2005**, *4*, 649−663.
(8) Schneider, G.; Clement-Chomienne, O.; Hilfiger, L.; Schneider, P.; Kirsch, S.; Böhm, H.-J.; Neidhart, W. Virtual Screening For Bioactive Molecules by Evolutionary *de Novo* Design. *Angew. Chem., Int. Ed.* **2000**, *39*, 4130−4133.
(9) Fechner, U.; Schneider, G. Flux (1): A Virtual Synthesis Scheme for Fragment-Based De Novo Design. *J. Chem. Inf. Model.* **2005**, *46*, 699−707.
(10) Schneider, P.; Schneider, G. Collection of Bioactive Reference Compounds for Focused Library Design. *QSAR Comb. Sci.* **2003**, *22*, 713−718.
(11) Lewell, X. O.; Budd, D. B.; Watson, S. P.; Hann, M. M. RECAP − Retrosynthetic Combinatorial Analysis Procedure: A Powerful New Technique for Identifying Privileged Molecular Fragments with Useful Applications in Combinatorial Chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511−522.
(12) Lipinski, C.; Hopkins, A. Navigating Chemical Space for Biology and Medicine. *Nature* **2004**, *432*, 855−861.
(13) Darwin, C. *On the Origin of Species a Facsimile of the First Edition*; Harvard University Press: Cambridge, Massachusetts, 1975.
(14) Rechenberg, I. *Evolutionsstrategie '94*; Frommann-Holzboog: Stuttgart, Germany, 1994.
(15) Schwefel, H.-P. Deep Insight From Simple Models of Evolution. *Biosystems* **2002**, *64*, 189−198.
(16) Steinbeck, C.; Han, Y. Q.; Kuhn, S.; Horlacher, O.; Luttmann, E.; Willighagen, E. The Chemistry Development Kit (CDK): An Open-Source Java Library for Chemo- and Bioinformatics. *J. Chem. Inf. Comput. Sci.* **2003**, *43*, 493−500.
(17) 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.
(18) Fechner, U.; Schneider, G. Optimization of a Pharmacophore-Based Correlation Vector Descriptor. *QSAR Comb. Sci.* **2004**, *23*, 19−22.
(19) Fechner, U.; Schneider, G. Evaluation of Distance Metrics for Ligand-Based Similarity Searching. *ChemBioChem* **2004**, *5*, 538−540.
(20) Dalby, A.; Nourse, J. G.; Hounshell, W. D.; Gushurst, A. K. I.; Grier, D. L.; Leland, B. A.; Laufer, J. Description of Several Chemical Structure File Formats Used by Computer Programs Developed at Molecular Design Limited. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 244−255.
(21) Weininger, D. SMILES, a Chemical Language and Information System. 1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.
(22) Krumrine, J. R.; Maynard, A. T.; Lerman, C. L. Statistical Tools for Virtual Screening. *J. Med. Chem.* **2005**, *48*, 7477−7481.
(23) Schneider, G.; Schneider, P.; Renner, S. Scaffold-Hopping: How Far Can You Jump? *QSAR Comb. Sci.* **2006**, *12*, 1162−1171.
(24) 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.
(25) Firth-Clark, S.; Willems, H. M. G.; Williams, A.; Harris, W. Generation and Selection of Novel Estrogen Receptor Ligands Using the de Novo Structure-Based Design Tool, SkelGen. *J. Chem. Inf. Model.* **2006**, *46*, 642−647.
(26) Lameijer, E.-W.; Kok, J. N.; Bäck, T.; Ijzerman, A. P. The Molecule Evoluator. An Interactive Evolutionary Algorithm for the Design of Druglike Molecules. *J. Chem. Inf. Model.* **2006**, *46*, 545−552.
(27) Kauffman, S. A. *The Origins of Order: Self-Organization and Selection in Evolution*; Oxford University Press: New York, 1993.
(28) Schneider, G.; So, S.-S. *Adaptive Systems in Drug Design*; Landes Bioscience: Austin, TX, 2001.
(29) (a) Kennedy, J.; Eberhart, R. C. Particle Swarm Optimization. In *Proceedings of IEEE International Conference on Neural Networks*, Perth, Australia, Nov 27−Dec 1, 1995; IEEE: Piscataway, NJ, 1995; Vol. 4, pp 1942−1948. (b) 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, Oct 4−6, 1995; IEEE: Piscataway, NJ, 1995; pp 39−43.
(30) Meissner, M.; Schmuker, M.; Schneider, G. Optimized Particle Swarm Optimization (OPSO) and Its Application to Artificial Neural Network Training. *BMC Bioinf.* **2006**, *7*, 125−136.
(31) Degen, J.; Rarey, M. FlexNovo: Structure-Based Searching in Large Fragment Spaces. *ChemMedChem* **2006**, *1*, 854−868.

CI6005307