A Drug Candidate Design Environment Using Evolutionary Computation

M. İhsan Ecemiş, James Wikel, Christopher Bingham, and Eric Bonabeau, *Member, IEEE*

Abstract—**This paper describes the Candidate Design Environment we developed for efficient identification of promising drug candidates. Developing effective drugs from active molecules is a challenging problem which requires the simultaneous satisfaction of many factors. Traditionally, the drug discovery process is conducted by medicinal chemists whose vital expertise is not readily quantifiable. Recently, *in silico* modeling and virtual screening have been emerging as valuable tools despite their mixed results early on. Our approach combines the capabilities of computational models with human knowledge using a genetic algorithm and interactive evolutionary computation. We enable the chemist's expertise to play a key role in every stage of the discovery process. Our evolved structures are guaranteed to be within the chemistry space specified by the medicinal chemist, thereby making the results plausible. In this paper, we describe our approach, introduce a case study to test our methodology, and present our results.**

Index Terms—**Computational chemistry, drug design, drug discovery, genetic algorithms (GAs), *in silico* modeling, interactive evolutionary computation, lead optimization (LO), virtual screening.**

I. Introduction

With the advent of new technologies (such as Genomics, High Throughput Screening, Molecular Biology, etc.) it was expected that many new drugs would be discovered in the current decade. However, the number of New Molecular Entities (NMEs) approved by the U.S. Food and Drug Administration has declined in recent years. On the other hand, the R&D costs have risen dramatically [1], far outpacing sales. According to the consulting firm Bain & Company, the cost of bringing a new blockbuster drug to market is estimated to be about \$1.7 billion [2]. If we simply define the productivity of the pharmaceutical industry as the ratio of the NMEs to the drug development costs, there is arguably a crisis in the field. Fig. 1 shows the key steps in the drug discovery and development process. Hit identification used to be the bottleneck before the development of High Throughput Screening (HTS) in the 1990s. However, the huge increase in the number of compounds screened did not translate into many more drug candidates. This suggests that the bottleneck has simply shifted to the lead optimization (LO) phase [3]. During LO, the structures exhibiting encouraging activity and selectivity in screening are converted into clinical candidates. Therefore, one area where innovation could reduce R&D costs most significantly is the lead[1] identification and optimization phase. It is important to find new ways to effectively search for new drug candidates through large molecular spaces and make better decisions about *what*, rather than *how*, to synthesize and screen. *In silico* modeling and virtual screening are currently perceived as the most promising technologies for addressing these challenges.

Traditionally, chemists focused primarily on enhancing the affinity and selectivity of lead series during LO. However, the drug potential of a molecule also depends on factors such as absorption, distribution, metabolic stability, excretion, solubility, and toxicity, just to name a few. Fully optimizing for affinity and selectivity at the expense of other properties leads to a limited number of alternative solutions and high attrition rates in later (and more costly) stages [3]. Simply put, this presents a multidimensional optimization problem which should be handled in parallel instead of the traditional sequential manner.

In 2004, we developed a Candidate Design Environment (CDE), Mobius, for identification of potential drug candidates [4]. Our CDE is based on a genetic algorithm (GA) and interactive evolutionary computation (IEC). Mobius fosters interactions between computational and medicinal chemists by leveraging the strengths of both parties: *In silico* models of computational chemistry through the GA and the insight and experience of medicinal chemists through IEC.

GA is a technique inspired by natural evolution to find approximate solutions to optimization and search problems [5], [6]. It has been successfully applied to many disciplines including computational chemistry [7], e.g., in protein docking [8], library design [9], etc. On the other hand, IEC integrates subjective human evaluation into Evolutionary Computing in order to address problems where the fitness function is not easily quantifiable [10].

Prior work related to Mobius has focused on *de novo* drug design. Glen and Payne [11] implemented a GA to create molecules satisfying a range of constraints based on calculated molecular properties. Their algorithm uses a series of rules to produce realistic molecules in three dimensions. However, the final compounds which fit the constraints and satisfy all chemistry rules may still be unacceptable to the medicinal chemist. Many of the evolved structures require modification to produce synthesizable and realistic drug-like molecules.

Schneider *et al.* [12] developed a method to evolve fragment-based *de novo* molecular structures similar to a template structure. Compounds are created from $\sim 25\,000$ fragments through

Manuscript received October 12, 2006; revised May 30, 2007. First published February 8, 2008; current version published October 1, 2008.

M. I. Ecemiş, J. Wikel, and C. Bingham are with Coalesix, Inc., Cambridge, MA 02138 USA (e-mail: ecemis@coalesix.com; jim@coalesix. com; bing@coalesix.com).

E. Bonabeau is with Icosystem, Inc., Cambridge, MA 02138 USA (e-mail: eric@icosystem.com).

Digital Object Identifier 10.1109/TEVC.2007.913131

[1]*Lead* is a chemical structure with a confirmed activity and selectivity profile that warrants further investigation.

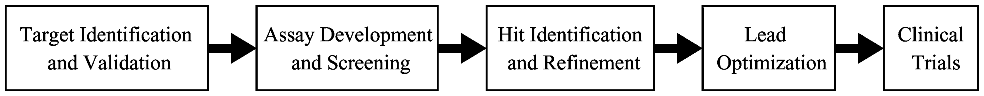

Target Identification and Validation
Assay Development and Screening
Hit Identification and Refinement
Lead Optimization
Clinical Trials

Fig. 1. Key steps in drug discovery and development process.

11 reaction schemes. The system derived molecules showing substantial bioactivity, though it was easily caught in a local optimum and was not able to perform final optimization due to the definitions of the building blocks. Goh *et al.* [13] evolved molecular structures to bind to a given protein target receptor. Their tree-based representation builds interesting molecules, similar to somewhat larger than known antiviral structures, which may prevent them from being good drug candidates. They also used a two-dimensional model of the target receptor which is not realistic.

Pegg *et al.* [14] applied a GA to create molecular structures as *acyclic* graphs of fragments. Their fitness function was composed of a docking score and drug-like properties. Their approach produced mixed results, partly because of the absence of an adequate performance metric. After Mobius, Lameijer *et al.* [15] reported an atom-based evolutionary method to design drug-like molecules, introducing a new representation of compounds and a new mutation operator. They applied IEC by having the *user* as the fitness function. In addition, the user can filter structures by their physical and chemical properties in order to evaluate only the more realistic ones. They reported some limited though promising results. However, many evolved structures seem difficult to synthesize since their system uses atom-based genetic operators but lacks chemical knowledge to ensure synthesizability.

Clerc *et al.* [16], [17] hybridized a classical GA with a knowledge discovery system based on K Nearest Neighbors Algorithm for optimizing catalyst libraries. In this problem, the fitness function is unknown and each formulation must be synthesized to get evaluated, which takes a long time. They started with a real catalyst library which is synthesized and tested. Then, the system evolved new virtual individuals and estimated the best ones in order to reduce the number of formulations to be evaluated empirically.

In multiobjective optimization domain, Babu *et al.* [18], [19] developed a novel differential evolution method and applied it to the optimization of nonlinear chemical processes. They showed that the performance of their approach is better than that of traditional direct search methods.

The rest of this paper is organized as follows. Section II describes our approach, Section III presents a project through which we quantified the performance of our system, and Section IV presents our conclusions.

II. Methods

Mobius facilitates the identification of a diverse set of preclinical drug candidates by combining computational models with expert knowledge. We assume that a set of compounds with encouraging drug potential are identified before using Mobius. This is a common case in the LO phase. The medicinal chemistry group starts a Mobius project by dissecting these compounds into key components. For each component, the user selects a library of fragments that could replace the original fragment, in order to create structures with potentially better properties. This process defines the search space in which the GA will seek the best solutions among a number of alternatives.

The next step is to determine the computational models to evaluate the GA-created structures as to their suitability to be drug candidates. The user adjusts the optimal value of each model and its relative importance. The selection of models and their optimal values depend on the therapeutic target chosen at the inception of the drug discovery process.

Once the search space and the fitness evaluator are specified, the user can start the search process, either from a random initial population or a set of potential compounds. Mobius evolves sets of compounds through its GA's crossover and mutation operators described later in this section. Compounds are created by selecting fragments for each component and optimizing the *in silico* criteria defined by the user.

The user can run the GA for a specific number of generations or indefinitely, in which case he should monitor its progress and stop it when the best population score exceeds a certain threshold or is no longer improving. When the GA stops, the compounds generated in the last population are presented to the user. The user evaluates the top 12 compounds in the population by providing negative or positive feedback. This enables human input on objectives that are not readily quantifiable, such as synthetic tractability. The feedback provided may reinforce or change the direction in which the GA is heading. At this time, the user can also review his prior decisions and make necessary adjustments, e.g., change the search space by adding or removing fragments, add or remove models, change model optimal values, or their relative importance, etc.

All the data generated by Mobius are available for detailed analysis later, such as reranking all compounds by different fitness criteria. The user may also decide to have some of the promising compounds synthesized. Synthesized compounds may provide the computational chemistry group with data to improve their predictive models. The entire process continues until diverse sets of optimized preclinical drug candidates are obtained. Fig. 2 shows a schematic diagram of the workflow described above.

A. Representation

Our chromosome is a fixed-length vector of genes where each gene expresses a fragment. The fixed-length nature of the chromosome may appear to be limiting the search space, but we will show that one can easily create a vast compound space with it. We call a *Blueprint* the recipe for making a compound from a

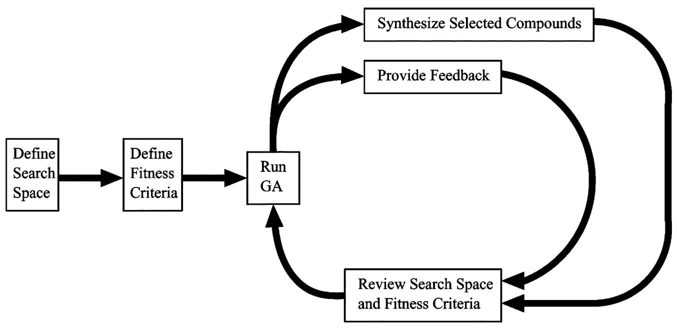

Synthesize Selected Compounds
Provide Feedback
Define Search Space
Define Fitness Criteria
Run GA
Review Search Space and Fitness Criteria

Fig. 2. High-level schematic diagram of Mobius's workflow.

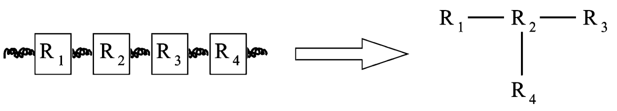


Fig. 3. A simple *blueprint* in Markush notation corresponding to a chromosome of four genes. The branching information is defined in the blueprint and only the fragments are changing in order to create new compounds.

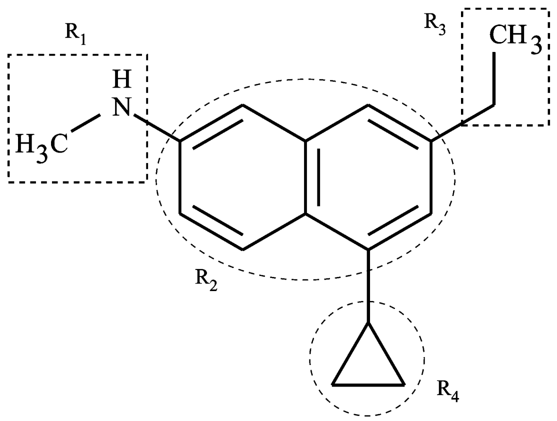


Fig. 4. A sample structure generated from the *blueprint* shown in Fig. 3.

chromosome. Fig. 3 shows a simple blueprint in Markush notation[2] corresponding to a chromosome of four genes. Since Markush structures are commonly used by medicinal and computational chemists, defining our search space with Markush notation proved to be very intuitive for our target users. A blueprint may contain a mix of fixed structures (*scaffold*) and variable structures (*R-groups*, corresponding to our genes). There must be at least one R-group, otherwise, the blueprint defines a single compound. The blueprint may consist entirely of R-groups with no fixed structure, as in Fig. 3. The chemist's expertise is crucial for an effective blueprint definition. We propose that creating molecules based on the fragment-based Markush notation yields more plausible structures compared with other approaches.

For each R-group, the user creates a distinct list of fragments, either from a library provided by Mobius, by importing structures from a file, or by using ChemAxon Ltd.'s drawing tool Marvin Sketcher, which is fully integrated into Mobius. Fragment libraries for various functional groups are at the disposal of most chemists. A fragment can contain any number of atoms, even no atom at all (we denote this case by a single Hydrogen atom). Our fragment definition also includes *connection sites*, where each fragment may be bonded to other fragments in order to build compounds, e.g., in Fig. 3, R_2 has three bonds, so all fragments in the list of substitutes for the corresponding gene need to define at least three connection sites. On the other hand, a single connection site suffices for R_1, R_3, and R_4. If a list contains only one substitute for a gene, that gene behaves as a *scaffold*. Fig. 4 shows a sample compound created by substituting all R-groups in Fig. 3 with specific fragments.

B. Mutation and Crossover

We mutate a compound by randomly selecting one of its genes and mutating it. For example, Fig. 5 shows how the compound shown on Fig. 4 may be mutated. In this example, the third gene is selected for mutation. Each gene's probability of being selected is directly proportional to the number of substitutes available to that gene, e.g., a gene with 50 substitutes is ten times more likely to be selected for mutation than a gene with five substitutes. We give more weight to genes with more substitutes to guarantee their alternative fragments are adequately sampled. If each gene had the same probability of being selected, the fragments for the gene with five substitutes would end up being oversampled compared with the gene with 50 substitutes.

Our mutation operator alters a gene by replacing its current fragment by another substitute randomly selected from the corresponding list of fragments. Each fragment in the list has the same probability of being selected to replace the original fragment.

The crossover operator blends the characteristics of a pair of parent chromosomes to create two new offspring. We first generate a random number between one and the total number of genes in the chromosome. This is the number of genes we will swap between the parents. We randomly select that many genes in the parents and cross them over to create the offspring, as shown in Fig. 6.

C. Fitness Evaluation

A long list of (often conflicting) factors influence the drug potential of a given compound. We assume that appropriate computational models exist to assess each factor. A model can be as simple as a molecular weight calculator or as complex as a protein docking model. It can also be a classification model, e.g., an activity model that returns *Active* or *Inactive*. So far, we have successfully integrated a number of computational models into Mobius such as structure property plugins (e.g., Calculator Plugins by ChemAxon, Ltd.), command line tools (e.g., ROCS by OpenEye Scientific Software, Inc.), proprietary models of big pharmaceutical companies through Web Services, etc. Mobius is completely agnostic to the models used.

Each computational model result is normalized to a value between 0 and 1 using a piecewise linear function. Fig. 7 shows an example of molecular weight normalization. Traditionally,

[2]Markush notation is a way of concisely describing a number of compounds by identifying a fixed core structure and listing some *functionally equivalent* variable structures (R-groups). Markush formulas are often used in patent claims since their generic nature makes the claims as broad as possible.

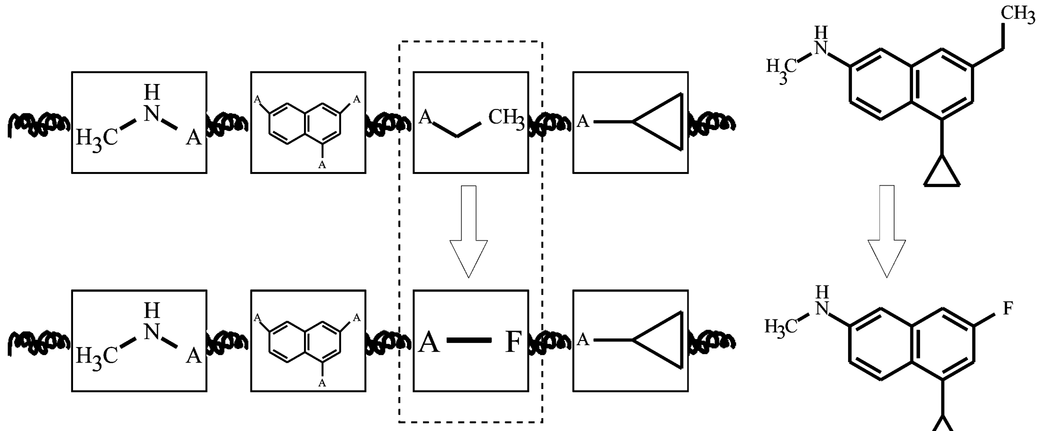


Fig. 5. Mutation operator. The third gene (CH_2CH_3, *Ethyl*) is selected for mutation and then replaced by a substitute (F, *Fluoro*) from the list of corresponding fragments. The letters A denote the connection points needed to build the compounds from the chromosomes. The original and mutated compounds are shown on the top and bottom right, respectively.

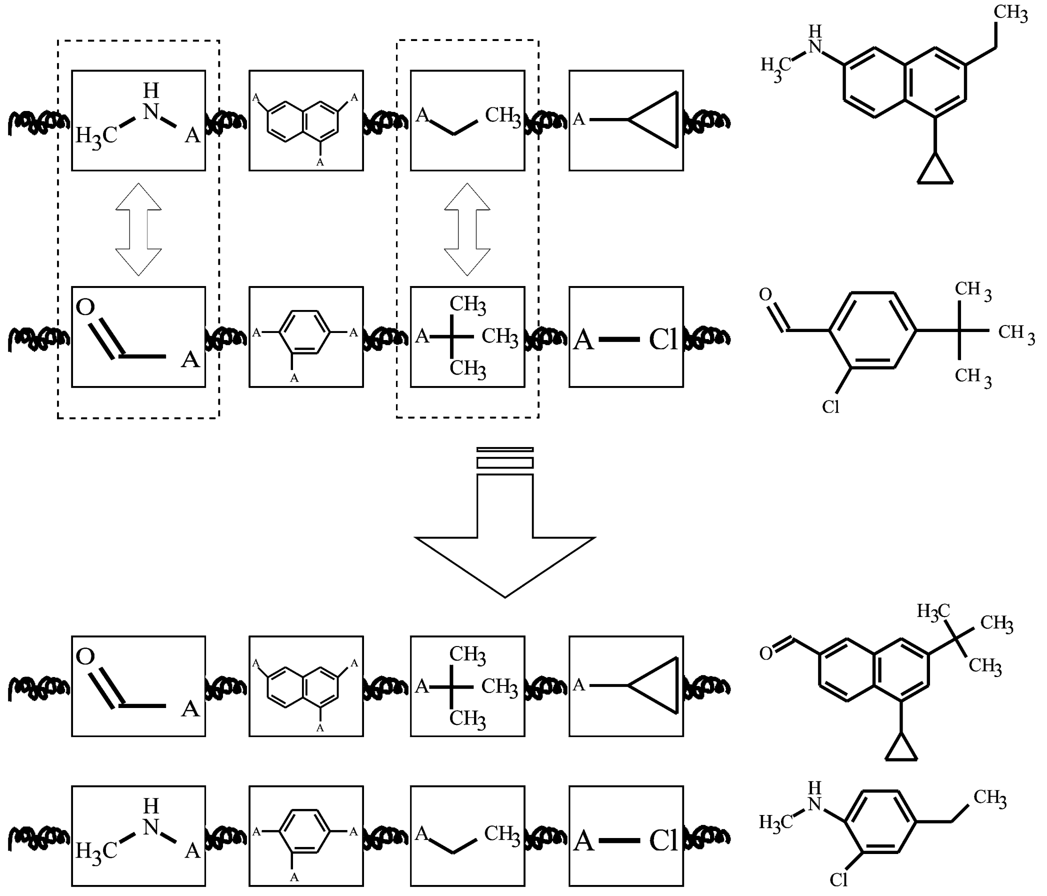


Fig. 6. Crossover operator. The first and third genes of the parent chromosomes (shown on top) are swapped to create two offspring (shown in the bottom). The letters A denote the connection points needed to build the compounds from the chromosomes. The parent and offspring compounds are shown on the top and bottom right, respectively.

chemists have set hard cutoff points to select which compounds to consider for optimization. They sometimes move their subjective thresholds if too many or too few compounds get through. This approach completely eliminates compounds which are just below or above the cutoff points. It is not uncommon that those compounds could induce ideas that would lead to a different optimization path. We found it more intuitive for medicinal chemists to define ranges where they would like to have a model value be (the *preferred* range) or absolutely not be (the *unacceptable* range). Within Mobius, the user defines the components in the function and adjusts the preferred and unacceptable ranges. If a model returns a value within the preferred range, it is normalized to 1 and if it returns a value within the unacceptable range, it is normalized to 0—no matter

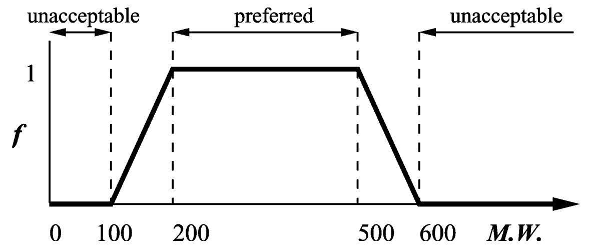


Fig. 7. Sample piecewise linear function to normalize Molecular Weight of a structure.

what the actual value is. Model values between a preferred and unacceptable range are interpolated linearly, as shown in Fig. 7.

The overall fitness is the weighted sum of normalized fitness values divided by a penalty term

$$F = \frac{\sum w_i f_i}{P \sum w_i}$$

where w_i and f_i are the weight and normalized score of the ith computational model. The penalty term P is the product of individual penalty terms p_i

$$P = \prod p_i, p_i = \begin{cases} 1 + w_i, & \text{if } f_i = 0 \\ 1, & \text{otherwise} \end{cases}.$$

We introduced the penalty term for the following reason. In the absence of the penalty term, if a compound fails half of the models (normalized score f_i of 0) and succeeds the rest (normalized score f_i of 1), it will be assigned an overall fitness score of 0.5 (assuming all models are weighted equally). Another compound that achieves a normalized score f_i of 0.5 from all the models will also be assigned an overall fitness score of 0.5. This is not a desirable outcome for the medicinal chemist; a compound that completely fails a few models should get a lower score since it can not be improved easily. Individual penalty terms are also proportional to the weights of the models so that failing important models brings a larger penalty.

Algorithm

Our GA starts with a population of N unique random compounds or a set of user-specified compounds. Since the GA's search procedure is based on stochastic operations, different initial conditions evolve into different near-optimal solutions. If the user ran several experiments starting the GA with random compounds each time, he would get different solutions providing worthwhile diversity. On the other hand, if a number of potential solutions are already available (e.g., from a hit series), the user can *seed* the GA with them. These user-specified initial compounds would expedite the search process, which is important if the compound space is vast. This biased initial population also reduces the diversity and produces solutions closer to what the user may want to see (e.g., more compounds similar to the initial hit series).

During evolution, all compounds in a population are sorted according to their fitness. A number N_e of top ranking individuals (so-called *elites*) are directly copied to the next generation.

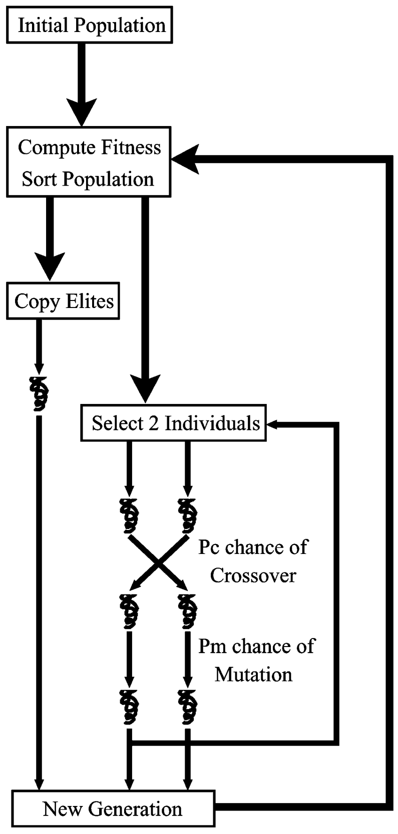


Fig. 8. High-level schematic diagram of the algorithm.

Then, we start the breeding steps by selecting two individuals for reproduction. The selection probability is inversely proportional to the rank of each individual: The probability of selecting the best individual is proportional to N, the probability of selecting the second best individual is proportional to $N-1$, etc.

Next, we crossover the selected individuals with a probability of P_c, and then mutate each resulting individual with a probability of P_m. We add the results to the next generation if they do not already appear there. We repeat these breeding steps until the next generation has N unique compounds. A diagram of the algorithm is shown in Fig. 8.

D. *User Interaction*

Mobius empowers the medicinal chemist to lead the search process by capturing his expert knowledge through a user-friendly graphical interface. The user can stop the progress of the GA at any time and redirect the search. There are five broad categories of user interaction.

- Blueprint: The medicinal chemist expresses his approach to solving the LO problem by determining the Markush structure. This is the first step of a project. The formulation of the blueprint may originate from a variety of sources, such as a hit series, HTS analysis, patent literature, etc.
- Fragments: The medicinal chemist defines the search space by selecting the list of fragments for each gene (R-group). Mobius does not create any compound which uses a fragment not chosen by the user. This ensures most generated structures are plausible from a chemical standpoint.

- Models: Mobius facilitates the adoption and usage of computational models by medicinal chemists. The user can define and redirect the search process by adjusting the normalization ranges and weights. Our users have devised various strategies for exploring diverse regions of the compound space. An example is to start with one or two models having a tight preferred range and high weight, and then to progressively tighten the preferred range of other models.
- Feedback: Structures created by Mobius are periodically presented to the chemist for evaluation. This mechanism coalesces human expertise into the search algorithm to satisfy objectives that are not quantifiable. The user can rate the compounds with positive or negative scores. Currently, our user interface supports the following values of the user rating $r : -1, 0.25, 0.5, 0.75$, and 1. A more precise scoring capability is not usually significant in the IEC paradigm. If the user rating is positive, at each generation we *inject* the rated compound into the population with a probability

$$P_i = \exp\left(\frac{-n}{\lambda r}\right)$$

  where n is the number of generations since the feedback is given and λ is a user defined constant (1000 by default). Positively rated compounds are in effect until the following feedback session. If $r = -1$ (negative feedback), we do not let the rated compound participate in any subsequent population. The list of *banned* compounds is in effect until the user resets the GA.
- Direct manipulation: The user can alter a compound created by Mobius by replacing one or more of its fragments with substitutes. For example, while evaluating a compound, the user may have a new idea and wonder how the fitness score would change if some part of the compound were different. Through direct manipulation, the user can get the model results and fitness score for the modified compound. He can also change the direction of the evolution, since user-modified compounds directly participate in the breeding for the subsequent generation.

III. Case Study

In this section, we present a project for evaluating the performance of our approach. Our goal is to show that Mobius can efficiently discover promising compounds that satisfy objective criteria, subjective human requirements, or both. For this, we used a published study and constructed a blueprint which spans not only the published compounds but also many more structures which could be potential solutions to the published problem. First, we describe our system. Second, we show the results of running the GA with the fitness function alone. In this case, Mobius is able to quickly find the optimal solutions in the search space. Finally, we present the impact of user interaction on the solutions evolved by the GA.

A. Setup

For the systematic study of our algorithm, we chose quinolone structures displaying antibiotic activity published by Klopman *et al.* [20], [21]. These articles describe the structure-activity relationships of 161 specific quinolones. Of particular relevance is the reporting of the activity of these structures against the clinical strains of *mycobacterium avium*, the cause of tuberculosis and the most frequent bacterial complication of AIDS. Based on the reported structures, we designed the Markush representation shown in Fig. 9. We call this blueprint Ciprofloxacin, or Cipro, after a well-known antibiotic featured among the reported 161 quinolones. Our Markush notation has five R-groups for which we defined the following number of fragments[3]: 11, 4, 6, 483, and 10 (in ascending R-group number order). Our search space spans 1 275 120 unique compounds, 132 of which are reported structures with known activity data. We could easily extend this space to more than 10^7 compounds by adding other obvious substitutes to each R-group. However, we kept the space relatively small so that we could generate and evaluate all compounds in order to measure the performance of our algorithm.

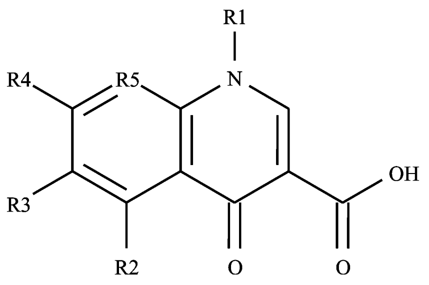


Fig. 9. *Cipro* blueprint designed after reported quinolones.

We selected the following properties as part of our fitness criteria: Molecular weight (MW), octanol/water partition coefficient $\log P$, octanol/water distribution coefficient $\log D$ (at $\text{pH} = 7.4$), polar surface area (PSA), rotatable bond count (RotBonds), hydrogen bond donor inclination (HDonors), hydrogen bond acceptor inclination (HAccept), acid-ionization constant (pKa), and base-ionization constant (pKb). In our experiments, we used ChemAxon Ltd.'s Calculator Plugins to compute these properties.

Unless noted otherwise, we used the following preferred and unacceptable normalization ranges:

$$f_{\text{MW}} = \begin{cases} 0, & \text{if MW} \leq 250 \\ 1, & \text{if } 350 \leq \text{MW} \leq 375 \\ 0, & \text{if MW} \geq 425 \end{cases}$$

$$f_{\log P} = \begin{cases} 0, & \text{if} \log P \leq -2 \\ 1, & \text{if} -1.5 \leq \log P \leq -0.5 \\ 0, & \text{if} \log P \geq 0.5 \end{cases}$$

$$f_{\log D} = \begin{cases} 0, & \text{if} \log D \leq -2 \\ 1, & \text{if } -1.5 \leq \log D \leq -0.5 \\ 0, & \text{if} \log D \geq 0.5 \end{cases}$$

$$f_{\text{PSA}} = \begin{cases} 0, & \text{if PSA} \leq 65 \\ 1, & \text{if } 70 \leq \text{PSA} \leq 80 \\ 0, & \text{if PSA} \geq 95 \end{cases}$$

[3]The list of fragments are available upon request from the authors. Even though defining fragments appears to be a tedious process, a number of functionally equivalent fragment libraries is readily available to most medicinal chemists. Therefore, fragment definition usually consists of loading fragment libraries into Mobius.

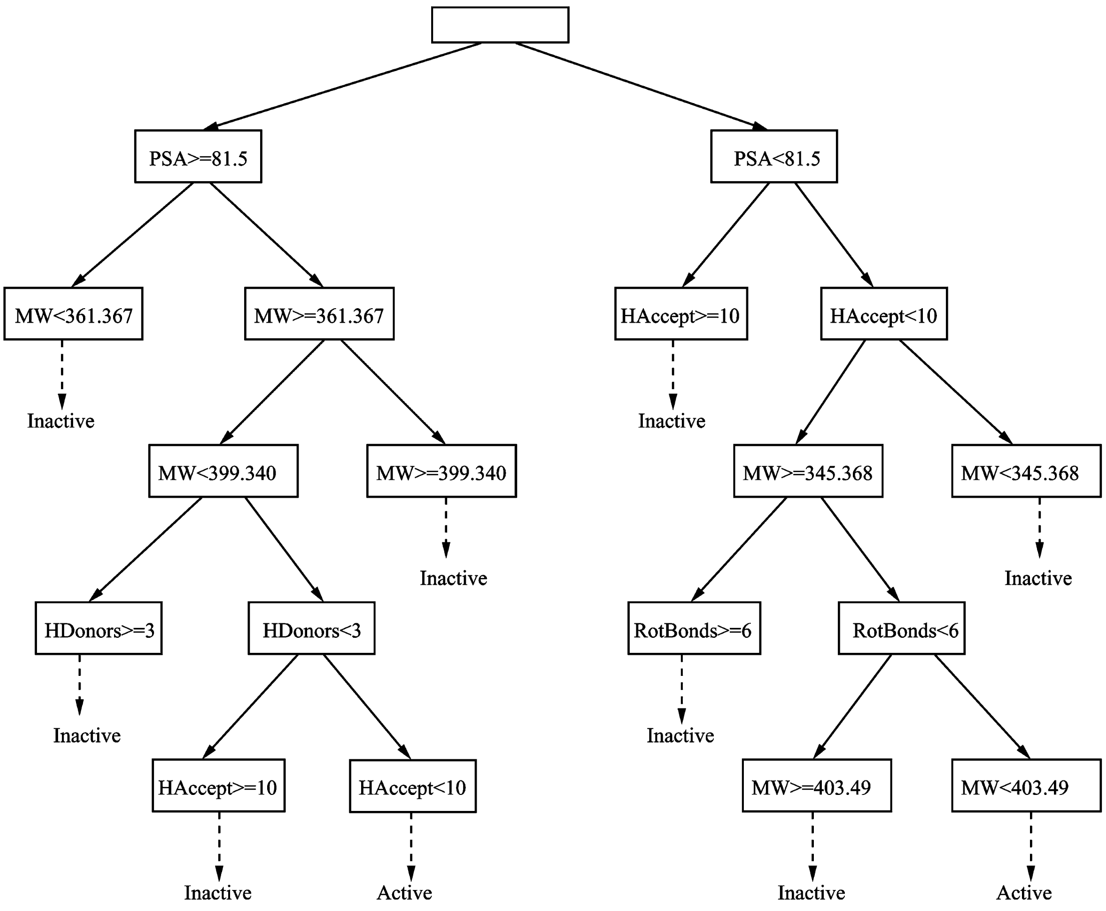


Fig. 10. Our QSAR model to predict the bioactivity of compounds generated by Mobius in Cipro study.

$$f_{\text{RotBonds}} = \begin{cases} 0, & \text{if RotBonds} \leq 1 \\ 1, & \text{if } 2 \leq \text{RotBonds} \leq 3 \\ 0, & \text{if RotBonds} \geq 6 \end{cases}$$

$$f_{\text{HDonors}} = \begin{cases} 1, & \text{ifHDonors} \leq 2 \\ 0, & \text{if HDonors} \geq 5 \end{cases}$$

$$f_{\text{HAccept}} = \begin{cases} 0, & \text{if HAccept} \leq 7.5 \\ 1, & \text{if } 8.5 \leq \text{HAccept} \leq 9 \\ 0, & \text{if HAccept} \geq 11 \end{cases}$$

$$f_{\text{pKa}} = \begin{cases} 0, & \text{if pKa} \leq 5 \\ 1, & \text{if } 5.25 \leq \text{pKa} \leq 5.5 \\ 0, & \text{if pKa} \geq 5.75 \end{cases}$$

$$f_{\text{pKb}} = \begin{cases} 0, & \text{if pKb} \leq 7 \\ 1, & \text{if } 7.25 \leq \text{pKb} \leq 9.25 \\ 0, & \text{if pKb} \geq 10 \end{cases} .$$

We interpolated the fitness scores for model values that fall between those ranges as mentioned above. All model weights were set to their default value of 0.5.

In addition to these nine property models, using the reported biological data we constructed a simple statistical model to predict the biological activity of a compound from its properties. We declared, for our study, compounds with MIC50[4] $\leq 2\ \mu\text{g/ml}$ are active, which produced 53 *known* actives and 108 *known* inactives.[5] This quantitative structure-activity relationship (QSAR) model is a decision tree resulting from a method known as recursive partitioning,[6] trained and cross validated to predict one of two responses: Active or Inactive. Fig. 10 shows our model. Note that there is no link between the normalization ranges of the previous nine property models and our QSAR model. In real-world experiments, the QSAR models are typically developed by the computational chemists and their implementations are not reviewed by the medicinal chemist. On the other hand, the medicinal chemist adjusts the normalization ranges. Even though our test fitness function does not include any computationally intensive models, it is similar in nature to those used in real-world experiments.

Table I shows the *confusion matrix* of our QSAR model for the 132 reported compounds in our search space. Our model makes false predictions as is the case in most real-world projects. Its accuracy and precision are 87.1% and 80.9%, respectively.

In order to validate the results presented Sections III-B–C, we enumerated all compounds in our search space and evaluated them with this fitness function. We found that 15 structures

[4]MIC50 stands for Minimum Inhibitory Concentration required to inhibit the growth of 50% of organisms and is a measure of bioactivity.

[5]According to this assumption Ciprofloxacin, with $\text{MIC50} = 4\ \mu\text{g/ml}$, is presumed *inactive*. Another example, Sparfloxacin, with $\text{MIC50} = 1\ \mu\text{g/ml}$, is considered *active*.

[6]We used the JMP application by SAS Institute, Inc. to build our model. Details are beyond the scope of this paper.

TABLE I
CONFUSION MATRIX OF OUR MODEL SHOWING THE NUMBER OF COMPOUNDS IN EACH ACTUAL AND PREDICTED CATEGORY

		QSAR Model	
		Active	Inactive
	Active	**38**	8
Known	Inactive	9	**77**

achieved a top score of 0.998. Even though identifying any of those top compounds is the primary goal for our search algorithm, there are other compounds with slightly lower scores that could look promising to a medicinal chemist. Therefore, for our study, we set a threshold of 0.95 and found 139 compounds with a higher score. We call those 139 compounds *exceptional* compounds. Quick identification of these exceptional compounds is also important for a real-world application like Mobius.

The *exhaustive* search of our Cipro Blueprint took 40 hours to complete on a 2.66 GHz Pentium 4 PC. It could take much longer in a real-world case. For example, one could have a 10^9-compound space with an elaborate docking model. Assuming a computer cluster evaluates one compound per second, an exhaustive search of that space would take approximately 32 years to complete.

B. GA Experiments

In our first set of experiments, we ran the GA without any user interaction. Our goal is to show that our GA is able to discover compounds that optimally satisfy multiple criteria specified by computational models. We used a random initial population and the following parameters unless otherwise noted:

- the population size $N = 100$;
- the number of elites $N_e = 2$;
- the probability of crossover $P_c = 0.8$;
- the probability of mutation $P_m = 0.1$.

Fig. 11 shows the best and average population scores as a function of the number of generations. The best fitness score is above 0.9 after only 12 generations, 0.95 at the 23rd generation and a top-score compound $(F = 0.998)$ is found at the 42nd generation. The average population fitness increases steadily in the first 25 generations, and then fluctuates as expected.

In order to investigate further, we allow the GA to run even after a top-score compound was found. Although the best score cannot improve at that stage, we would like to ascertain whether the GA discovers other known actives or exceptional compounds. Fig. 12 shows that the GA discovers them indeed. We have already identified 11 exceptional compounds in 42 generations when a top-score compound is found. We identified 41 exceptional compounds in 100 generations, and 120 (86.3% of the total) in 600 generations. At the same time, we identified 37 known actives out of 53. Note that we did not expect to identify all of the known actives due to an inherent problem with the fitness function used in these experiments. Fig. 13 shows

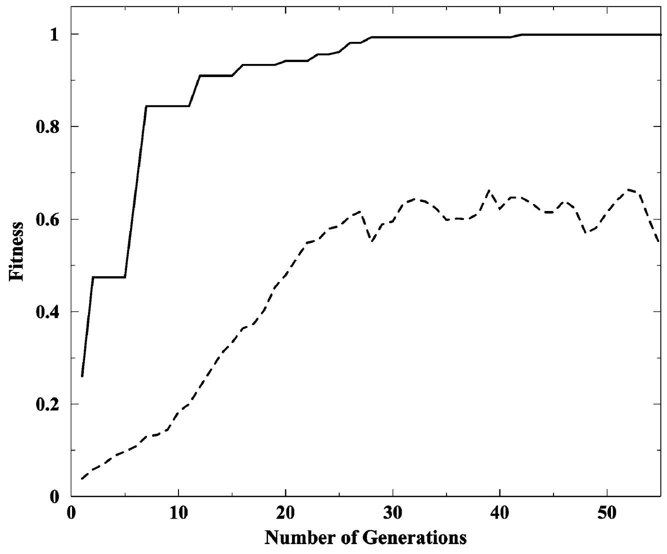


Fig. 11. Best fitness (solid line) and average population fitness (dashed line) as a function of number of generations.

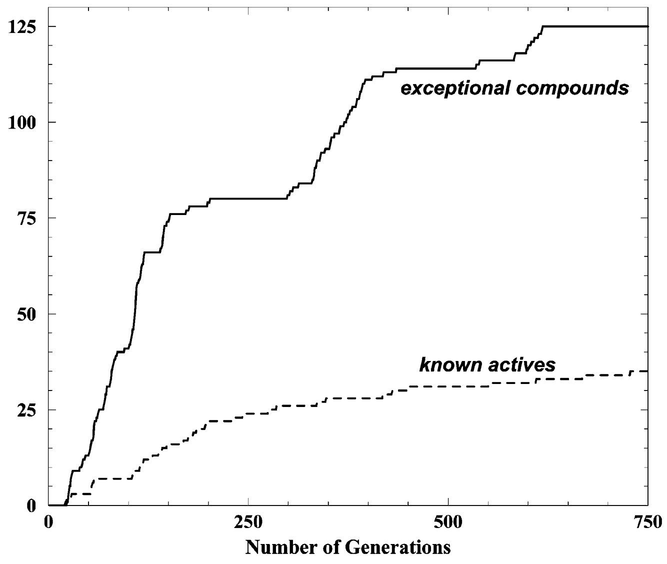


Fig. 12. The number of exceptional compounds and known actives identified as a function of number of generations.

the fitness histogram of known actives and inactives. There are only 31 known actives with a fitness larger than 0.6. Correctly identifying more than 30 known actives is reassuring. Depicting the number of generations as the performance metric may be relevant for a computational scientist but not for a medicinal chemist. A better metric is the amount of time the user waits for the results or the computational resources allocated; these may not be proportional to the number of generations. We timed all functions in our breeding algorithm and found that the time spent in creating new compounds (which includes mutation, crossover, and fusing fragments) is insignificant compared with running the models on them. If one used realistic docking tools besides our simple property models, then almost all computational time would be spent on model execution. Since Mobius

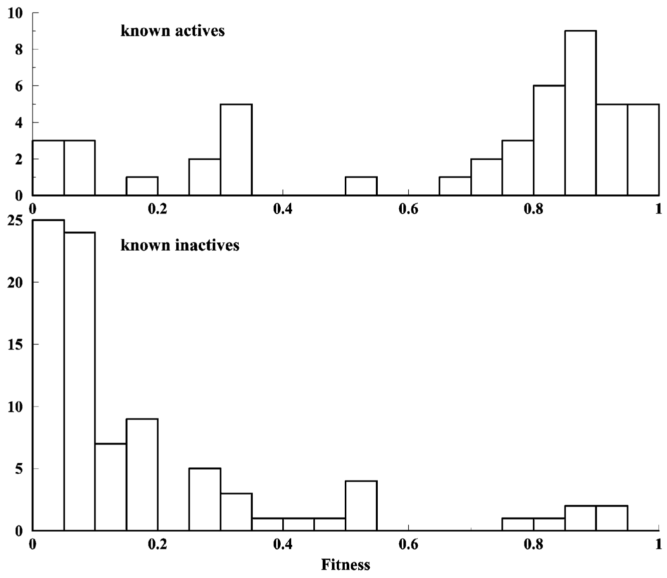


Fig. 13. Fitness histogram of known actives and inactives.

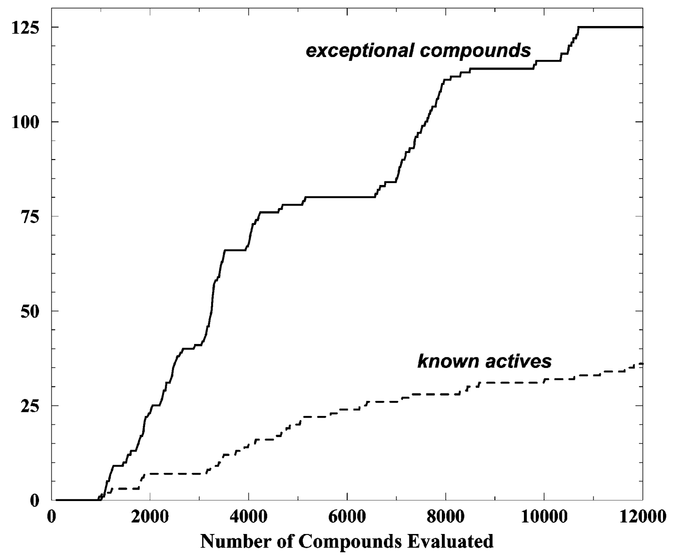


Fig. 14. The number of exceptional compounds and known actives identified as a function of the number of compounds evaluated.

stores the model results of all evaluated compounds in a database for quick lookup,[7] the amount of time the computational resources are used is approximately proportional to the number of unique compounds generated.

Our GA ensures that all compounds in a given generation are unique but does not ensure uniqueness. Overall, as the algorithm converges, it creates fewer and fewer new compounds per generation. For example, in the first experiment. the 20th generation has only 34 new compounds. After 200 generations, we create around 15 new compounds per generation, and this number drops to less than 10 after 1000 generations. Thus, the running time of successive generations becomes faster and faster.

Fig. 14 shows the number of known actives and exceptional compounds identified, as a function of the number of compounds evaluated by the computational models. We evaluated only 1526 compounds to find a top-score compound in the 42nd generation: only 0.12% of the search space. When 0.5% of the search space is evaluated (6383 compounds), we have already identified 80 exceptional compounds (57.6% of the total). This number increases to 125 (89.9% of the total) when 1% of the search space is evaluated.

Since the performance of a GA is influenced by its initial population, we repeated the experiment above 50 times starting with different random populations. Fig. 15 shows the statistics of these experiments in terms of best population score. On average, the best score gets above 0.95 after evaluating about 700 compounds—only 0.05% of the search space. In the worst case among these 50 runs, we had to evaluate 0.12% of the search space to find a compound with fitness > 0.95, and 0.28% of the search space to identify a top-score compound.

The identification of exceptional compounds was also very efficient, as shown in Fig. 16. On average, we identified 99.5 ex-

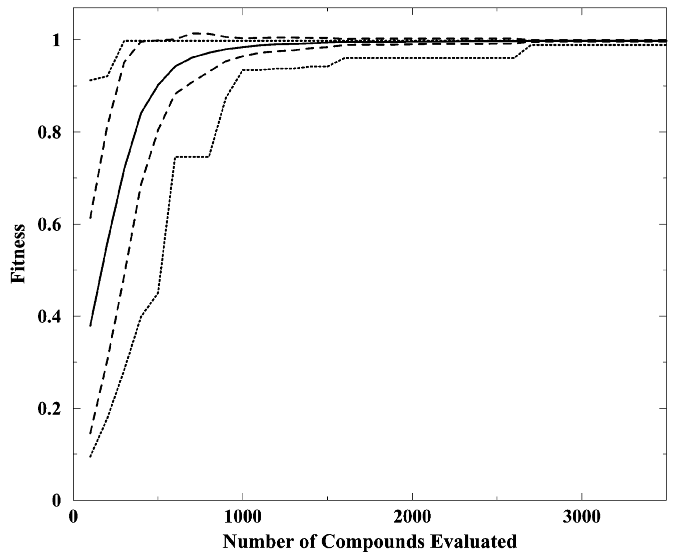


Fig. 15. Best population score versus the number of compounds evaluated. Solid line shows the mean of 50 runs. Dashed lines are one standard deviation away from the mean. The minimum and maximum best population scores of these 50 runs are shown with dotted lines.

ceptional compounds (71.6% of the total, best run is 128, worst run is 57) after evaluating 0.5% of the search space. This number increased to 118.8 (85.5% of the total, best run is 135, worst run is 76) when we evaluated 1% of the search space.

Fig. 17 shows the identification of known actives. On average, 22.3 known actives (42.1% of the total) could be identified after evaluating 0.5% of the search space. When 1% of the search space is evaluated, we identified 35.3 known actives (66.6% of the total).

Finally, we investigated the performance of our GA under various parameter settings. We did not observe significant changes in the results when we swept the following parameter ranges: $1 \leq N_e \leq 4$ and $0.7 \leq P_C \leq 0.9$. The only parameter which proved to be crucial was P_m, as shown on Fig. 18. Smaller

[7]We have successfully integrated Mobius with Oracle, MySQL, Derby, and HSQL databases. In this particular study, we used an in-memory cache for best performance.

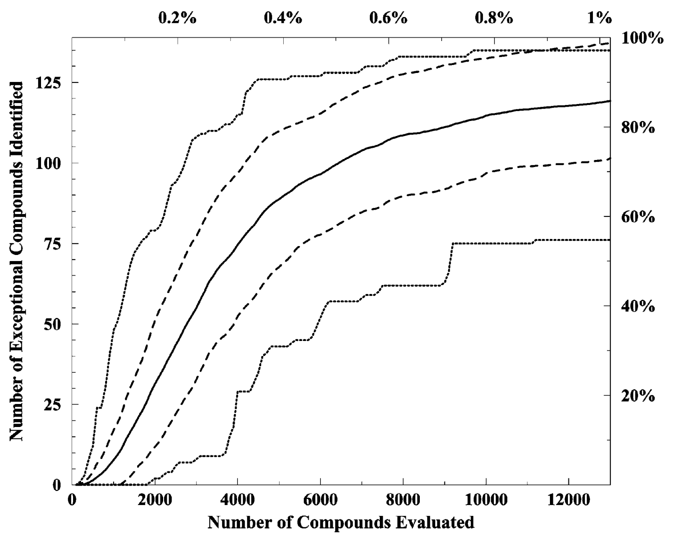


Fig. 16. Number of *exceptional* compounds identified versus the number of compounds evaluated. Solid line shows the mean of 50 runs. Dashed lines are one standard deviation away from the mean. The minimum and maximum *exceptional* compounds identified are shown with dotted lines. The tick labels on the top and on the right show the percentage of the search space evaluated and the *exceptional* compounds identified, respectively.

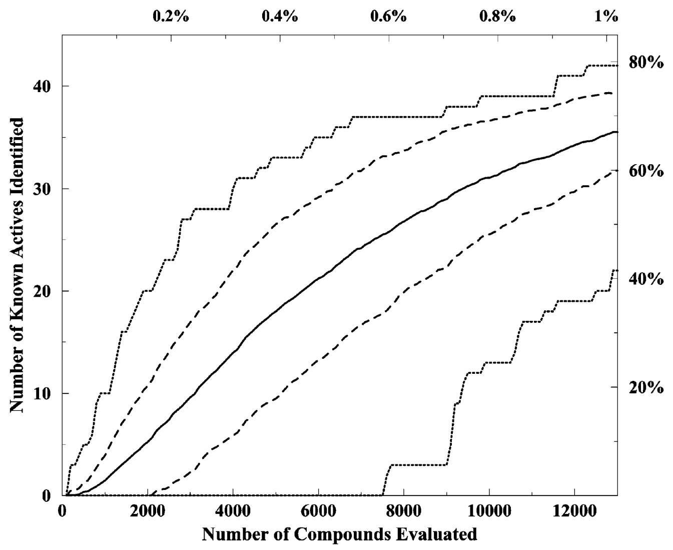


Fig. 17. Number of known actives identified versus the number of compounds evaluated. Solid line shows the mean of 50 runs. Dashed lines are one standard deviation away from the mean. The minimum and maximum known actives identified are shown with dotted lines. The tick labels on the top and on the right show the percentage of the search space evaluated and the known actives identified, respectively.

mutation probabilities increased the rate at which the exceptional compounds were identified. However, changing P_m did not considerably affect the progress of the best population score or when a top-score compound is identified. This suggests that once the GA converges to a good solution, high levels of mutation hinder the creation of other quality alternatives by permitting more random *jumps* in the compound space (even though more unique compounds are created per generation).

C. User Interaction Experiments

In Section II-E, we described five broad categories of user interaction in Mobius. In this section, we present experiments showing the impact of user interaction under the Models and Feedback categories. The other categories are not suitable for systematic experiments since they change the nature of the problem.

As mentioned above, under the Models category, the user can redirect the GA to a different set of solutions by adjusting the normalization ranges and weights of computational models. We tested this interaction by first running the GA with certain model settings until it converged to some near-optimal solutions. We then altered those settings, resumed the GA with the population produced with the old settings, and observed the new solutions.

Fig. 19 shows the best and average population scores as a function of the number of generations for this experiment. We started the GA with the model settings mentioned in Section III-A, and the GA discovered a near-optimal solution in eight generations (not a top-score compound however).

Then, we stopped Mobius after 25 generations and changed the model ranges for PSA (polar surface area) as follows:

$$f_{\mathrm{PSA}} = \begin{cases} 0, & \text{if PSA} \leq 110 \\ 1, & \text{if } 110 < \text{PSA} \leq 120 \\ 0, & \text{if PSA} > 120 \end{cases}.$$

We also emphasized the importance of PSA by increasing its weight to 0.9 and lowering all other model weights to 0.1. At that moment, the current compounds did not fit well to these new model settings and the best population score dropped to 0.39. However, when we resumed the GA, Mobius quickly discovered new solutions and the best population score doubled within 30 generations. We looked at the average number of heteroatoms[8] in order to quantify the difference the new model settings made. Fig. 20 shows the population average of the number of heteroatoms as a function of generations. The number of heteroatoms drop when the GA uses the initial model settings but then increases when we increased the preferred range for PSA from $[70, 80]$ to $[110, 120]$ and its prominence from 1 to 9, relative to other models. With this change, the new solutions comprise more heteroatoms in order to gain polar surface area. This result confirms that this user interaction redirects our GA to a different location in the compound space.

Next, we present the impact of user feedback on the solutions evolved by Mobius. Ultimately, the goal of the GA is to find compounds with the highest fitness score as defined through the computational models. However, the models are not by themselves sufficient to completely set apart good drug candidates from compounds bound to fail. The medicinal chemist has invaluable expertise which, if captured, could complement the models and lead the search algorithm to superior solutions.

For this experiment, we provided feedback to Mobius through the user rating feature mentioned in Section II-E. During our early experiments without user interaction, we observed that most near-optimal solutions contained a chlorine

[8]In organic chemistry, a heteroatom is any atom that is not carbon or hydrogen.

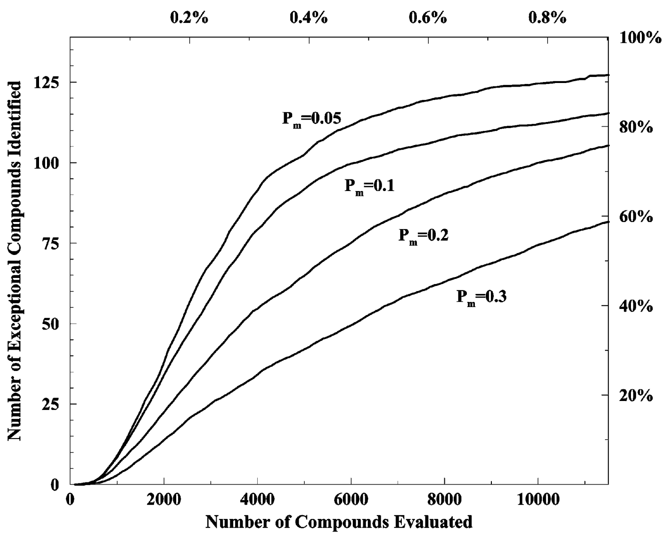


Fig. 18. Average number of *exceptional* compounds identified over 50 runs versus the number of compounds evaluated for different values of P_m. The tick labels on the top and on the right show the percentage of the search space evaluated and the *exceptional* compounds identified, respectively.

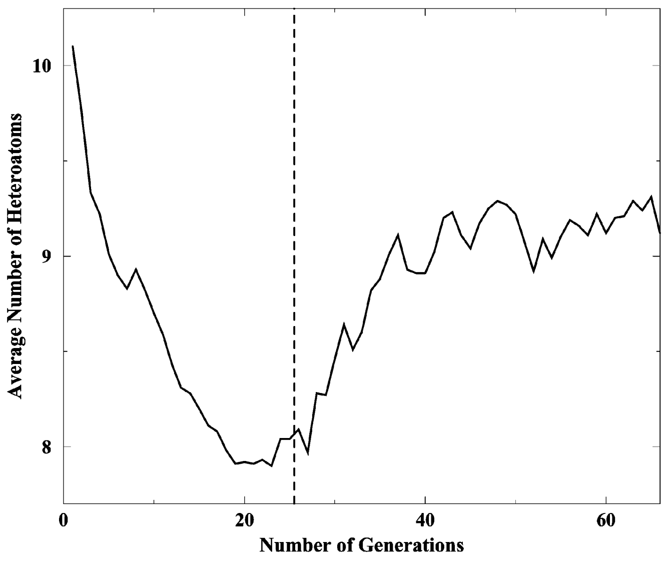


Fig. 20. Average number of heteroatoms as a function of number of generations. The dashed line demarcates where we changed the PSA model settings.

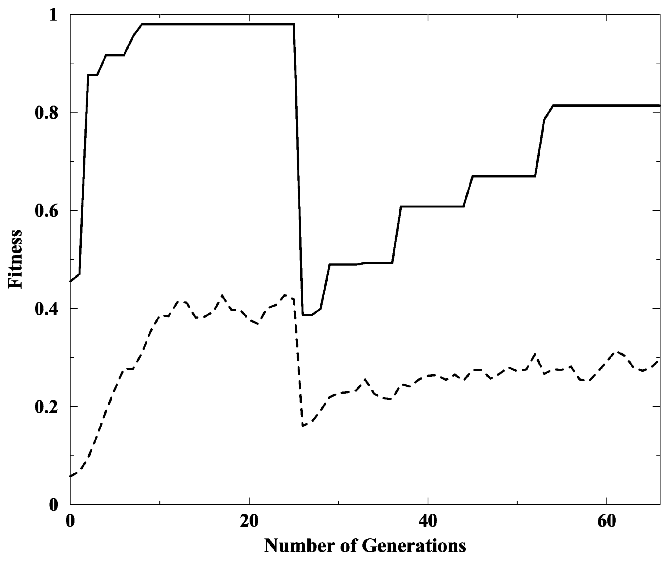


Fig. 19. Best fitness (solid line) and average population fitness (dashed line) as a function of number of generations. We changed the model settings after the 25th generation, hence the sudden drop in fitness.

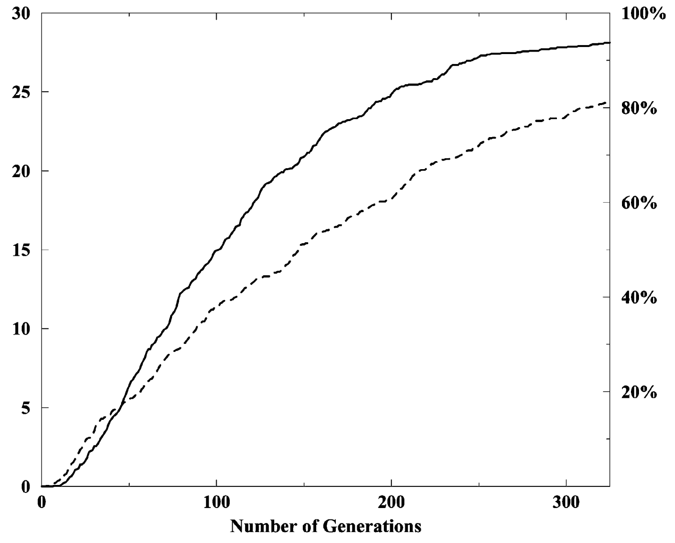


Fig. 21. Number of exceptional Cl-lacking compounds identified versus number of generations. Solid and dashed lines shows the mean of 20 runs with and without user feedback, respectively. The tick labels on the right show the percentage of the exceptional Cl-lacking compounds identified.

(Cl) atom. We confirmed this observation by counting only 30 Cl-lacking compounds among the 139 exceptional compounds. What if the medicinal chemist knew that the Cl atom would hurt the chances of succeeding in the clinical trials but the computational chemist could not incorporate this information in the computational models? Our solution is to enable the medicinal chemist to rate the compounds through his subjective instincts. In this case, he can rate the compounds according to whether they include Cl or not.

We tested Mobius by stopping the GA every 25 generations and providing feedback. At each feedback session, we reviewed the top 12 compounds in the population and assigned a user rating r of 1 to those lacking a Cl atom and -1 to those including one or more Cl atoms. Fig. 21 shows the average of 20 runs in terms of the number of exceptional Cl-lacking compounds identified, with and without user feedback. The user feedback did not make a significant difference in the first 50 generations. This was expected since the total number of compounds rated is only 24 (compared with $\sim$ 1800 compounds created). After the fourth feedback session (the 100th generation), it is clear that user ratings helped Mobius to discover more of the exceptional Cl-lacking compounds. This result should satisfy the medicinal chemist as he sees more of his preferred compounds as solutions.

IV. Discussion

We have presented our approach for efficient identification of promising drug candidates. Combining *in silico* models, parallel optimization techniques, and expert knowledge, Mobius can identify more and better drug candidates, faster than conventional methods; and hence lower the current high attrition rates during costly clinical trials. Any gain in this early phase of drug discovery is crucial for the pharmaceutical industry.

We have also presented a case study through which we tested our method. The results are promising: Mobius evaluated only a small fraction of a reasonably large search space and identified not only the best compound in that space but also a substantial percentage of other promising compounds. Our algorithm is very robust with respect to the particular choice of parameters.

Besides the case study presented here, we also applied our approach to a retrospective LO project in collaboration with a large pharmaceutical company. The original project had lasted three years, following the traditional approach, and failed to produce a drug candidate, mostly due to the sequential optimization strategy and to focusing on a few subjective alternatives instead of exploring multiple avenues. With the help of the medicinal chemistry group who worked on the project, we defined a vast search space spanning $\sim 5.6 \times 10^{12}$ compounds. To build our fitness function, we used proprietary models developed for the target molecule in addition to the drug-like property calculators presented here. We injected the original hit compound into the population during evolution as if it had a positive feedback score. After running our GA for about a week, we generated three promising series that the original team did not come close to finding. It could have taken the traditional approach decades to discover these series.

Mobius differs from previous approaches mainly through its genotype representation. Our approach does not fit into *de novo* drug design since the user defines a fixed-sized (though very large) chemistry space. The generated molecules are thus not outside of what the medicinal chemist defined, ensuring plausible outcome. Most computational methods *creating* structures in vast compound space end up with obviously unacceptable solutions. This discourages the medicinal chemist and prevents him from adopting new computational techniques. In general, user interactions are an essential part of Mobius's workflow. Expert feedback from the medicinal chemist provides direction to the search algorithm.

Mobius depends on *in silico* models, and the availability and predictive accuracy of these models are variable. They are based upon biological models that are inherently noisy. Without user feedback, our GA's performance is limited by the most accurate model used in the fitness function. In general, medicinal chemists are skeptical of computational tools, even though the tools have been improving considerably in recent years. Our preliminary results leveraging the medicinal chemists' feedback suggest a huge increase in productivity, but this requires a mindset change in the overall drug discovery process. With the chemist's intuition as part of the search algorithm more models can be useful.

Medicinal chemists are also hesitant to use computational tools because of their poor user interfaces. We expect Mobius to change that since it provides a common, simple, and consistent interface to all the models needed for a given LO project. Meanwhile, computational chemists may enjoy wider use of their models.

ACKNOWLEDGMENT

The authors thank S. Malinchik and J. Potter for their contribution to the proof-of-concept version of this work, they thank A. Schacht, M. Fisher, and M. Wiley for their help in formulating some of the concepts in our approach, and they also thank M. Wang and D. Johnson for the feedback they provided on Mobius.

REFERENCES

[1] J. A. DiMasi, R. W. Hansen, and H. G. Grabowski, "The price of innovation: New estimates of drug development costs," *J. Health Econ.*, vol. 22, no. 2, pp. 151–185, 2003.

[2] P. Landers, "Cost of developing a new drug increases to about $1.7 billion," *The Wall Street Journal*, Dec. 2003.

[3] K. H. Bleicher, H.-J. Böhm, K. Müller, and A. I. Alanine, "Hit and lead generation: Beyond high-throughput screening," *Nature Reviews Drug Discovery*, vol. 2, no. 5, pp. 369–378, 2003.

[4] E. Bonabeau, C. Anderson, B. Orme, P. Funes, O. Bandte, M. Sullivan, S. Malinchik, and J. Rothermich, "Methods and systems for interactive evolutionary computing (IEC)," U.S. 7043463, May 9, 2006.

[5] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis With Applications to Biology, Control and Artificial Intelligence*. Cambridge, MA, USA: MIT Press, 1992.

[6] D. B. Fogel, *Evolutionary Computation: Toward a New Philosophy of Machine Intelligence*. Piscataway, NJ: IEEE Press, 1995.

[7] G. Jones, "Genetic and evolutionary algorithms," in *Encyclopedia of Computational Chemistry*, P. von Rague, Ed., Chichester, U.K., 1998.

[8] G. Jones, P. Willett, and R. Glen, "Molecular recognition of receptor sites using a genetic algorithm with a description of desolation," *J. Molecular Biol.*, vol. 245, pp. 43–53, 1995.

[9] V. J. Gillet, W. Khatib, P. Willett, P. J. Fleming, and D. V. S. Green, "Combinatorial library design using a multiobjective genetic algorithm," *J. Chem. Inf. Comput. Sci.*, vol. 42, no. 2, pp. 375–385, Mar.-Apr. 2002.

[10] H. Takagi, "Interactive evolutionary computation: Fusion of the capabilities of EC optimization and human evaluation," *Proc. IEEE*, vol. 89, pp. 1275–1296, 2001.

[11] R. C. Glen and A. W. R. Payne, "A genetic algorithm for the automated generation of molecules within constraints," *J. Computer-Aided Molecular Design*, vol. 9, no. 2, pp. 181–202, 1995.

[12] G. Schneider, M.-L. Lee, M. Stahl, and P. Schneider, "De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks," *J. Computer-Aided Molecular Design*, vol. 14, no. 5, pp. 487–494, 2000.

[13] G. K.-M. Goh and J. A. Foster, "Evolving molecules for drug design using genetic algorithms via molecular trees," in *Proc. GECCO*, L. D. Whitley, D. E. Goldberg, E. Cantú-Paz, L. Spector, I. C. Parmee, and H. Beyer, Eds., 2000, pp. 27–33.

[14] S. C.-H. Pegg, J. J. Haresco, and I. D. Kuntz, "A genetic algorithm for structure-based de novo design," *J. Computer-Aided Molecular Design*, vol. 15, no. 10, pp. 911–933, 2001.

[15] E.-W. Lameijer, J. N. Kok, T. Bäck, and A. P. IJzerman, "The molecule evoluator. An interactive evolutionary algorithm for the design of drug-like molecules," *J. Chem. Inf. Model.*, vol. 46, no. 2, pp. 545–552, 2006.

[16] F. Clerc, M. Lengliz, D. Farrusseng, C. Mirodatos, S. R. M. Pereira, and R. Rakotomalala, "Library design using genetic algorithms for catalyst discovery and optimization," *Rev. Scientific Instrum.*, vol. 76, pp. 2208–2208, Jun. 2005.

[17] S. R. M. Pereira, F. Clerc, D. Farrusseng, J. C. van der Waal, T. Maschmeyer, and C. Mirodatos, "Effect of the genetic algorithm parameters on the optimization of heterogeneous catalysts," *QSAR Combinatorial Sci.*, vol. 24, no. 1, pp. 45–57, 2005.

[18] B. V. Babu and R. Angira, "Modified differential evolution (MDE) for optimization of non-linear chemical processes," *Comput. Chem. Eng.*, vol. 30, no. 6–7, pp. 989–1002, 2006.

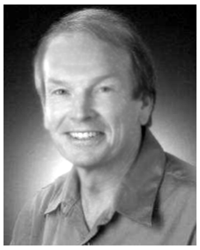
[19] B. V. Babu and R. Angira, "Optimization of process synthesis and design problems: A modified differential evolution approach," *Chem. Eng. Sci.*, vol. 61, no. 14, pp. 4707–4721, 2006.

[20] G. Klopman, S. Wang, M. R. Jacobs, S. Bajaksouzian, K. Edmonds, and J. J. Ellner, "Anti-mycobacterium avium activity of quinolones: In vitro activities," *Antimicrobial Agents and Chemotherapy*, vol. 37, no. 9, pp. 1799–1806, 1993.

[21] G. Klopman, D. Fercu, T. E. Renau, and M. R. Jacobs, "N-1-tert-butyl-substituted quinolones: In vitro anti-mycobacterium avium activities and structure-activity relationship studies," *Antimicrobial Agents and Chemotherapy*, vol. 40, no. 11, pp. 2637–2643, 1996.

M. İhsan Ecemiş received the B.S. and M.S. degrees in theoretical physics from Bilkent University, Ankara, Turkey, in 1993 and 1995, respectively, and the Ph.D. degree in cognitive and neural systems from Boston University, Boston, MA, in 2001.

He has extensive experience with complex systems, distributed adaptive algorithms, machine learning and pattern recognition methods, optimization techniques, nonlinear system analysis, data mining, and agent-based modeling. During his physics studies, he proposed a novel computational method to explain electron transport in semiconductor chips. His doctorate research focused on learning, vision and sonar recognition in mobile robotics and he specialized in the application of neural networks to the control of mobile robots and classification problems. During his dissertation work, he invented an inexpensive, simple, fast, and robust object recognition system using ultrasonic sensors. He successfully deployed his system in commercial and military settings through his collaboration with iRobot Corporation. As a Scientist at Icosystem Corporation, he has been actively involved in developing and applying algorithms to the solution of diverse scientific and business problems involving complex systems. He designed and implemented distributed control strategies for a swarm of robots under a DARPA project that resulted in complex, emergent, and intelligent swarm behaviors both in software and hardware. In 2004, he founded Coalesix, Inc., Cambridge, MA, and developed Mobius, a software application aimed at improving the drug discovery cycle. He is currently working at Coalesix, Inc., a division of Icosystem Corporation, as the V.P. of Technology and at Icosystem Corporation as the Director of Innovative Science and Technology.



James Wikel was born on August 29 1947, in Beckley, WV. He received the B.S. and M.S. degrees in chemistry from Marshall University, Huntington, WV, in 1969 and 1971, respectively.

He is currently the Chief Technology Officer of Coalesix, Inc., a division of Icosystem, Inc., Cambridge MA, founded in 2004. He retired from Eli Lilly & Company in 2004 as head of the Department of Structural and Computational Sciences, Discovery Chemistry Research and Technologies Division of Lilly Research Laboratories, Eli Lilly & Company, Indianapolis, IN. He joined Lilly in 1971 as an Organic Chemist and moved into the emerging computational chemistry area in 1989 as a founding member of that department created to maximize the corporate investment in a Cray-2 Supercomputer. He has been actively engaged in pharmaceutical research for over 33 years as both a laboratory scientist and as a scientific manager and has 34 peer-reviewed scientific publications and 47 issued U.S. patents. The subject matter included in these patents and publications describe three molecules that underwent clinical evaluation as drug candidates, enviroxime, enviradene, and frentizole, and one successfully marketed agricultural product, BEAM. As a computational chemistry scientist, he has published in a broad range of topics with expertise in QSAR studies and algorithm development. He established the QSAR group at Lilly and initiated the development of proprietary predictive methods. He lead a group of information technologists, computer scientists, and computational chemistry scientists from within the global Lilly Research Labs scientific community and in partnership with external collaborators to deliver computational methods across the organization via a web based integration framework application.

Mr. Wikel is a member of the American Chemical Society.

Christopher Bingham received the B.A. degree in computer science (*summa cum laude*) from Dartmouth College, Hanover, NH, in 1992, doing thesis work on genetic algorithms and machine learning.

His commercial experience began at Apple Computer, where he was the Chief Architect of the Macintosh Finder during the development of MacOS System 8. Subsequently, he was a key member of the web browser team at Netscape Communications, helping to create Netscape Navigator 2, 3, and 4. After Netscape, he focused on high-performance dynamic web server technology for a number of years. Among other projects, he created a free web-based e-mail system for the Disney Internet Group that had over one million users, and as System Architect of FairMarket, Inc., he developed large-scale dynamic-pricing e-commerce systems for clients such as Microsoft, Wal-Mart, American Express, and hundreds more. In 2004, he joined Coalesix, Inc., to oversee the technical architecture and development process of Mobius, a software system applying interactive evolution to pharmaceutical lead optimization. He is currently Vice President of Software Engineering at Coalesix, Inc., a division of Icosystem Corporation, as well as Director of Software Development and Architecture at Icosystem Corporation.

Mr. Bingham is a member of Phi Beta Kappa. He received the 1991 John G. Kemeny Computing Prize.

Eric Bonabeau (M'02) received the Ph.D. degree in theoretical physics from Paris-Sud University, Paris, France

He is an alumnus of Ecole Polytechnique and Ecole Nationale Supérieure des Télécommunications, both in France. He is founder and CEO of Boston-based Icosystem Corporation, Cambridge, MA. He is one of the world's leading experts in complex systems and distributed adaptive problem solving. His book *Swarm Intelligence* has been a scientific bestseller for eight years and provided the inspiration for another bestseller, Michael Crichton's *Prey*. His numerous articles in Harvard Business Review and MIT Sloan Management Review have all been exploring the limits of human decision making in a complex, decentralized, and unpredictable world. His commercial experience includes years of research and development in U.S. and European telecommunications and software companies. He sits on the advisory board of a number of Fortune 500 corporations. Prior to his current position, he was the CEO of Eurobios, then a joint venture with Cap Gemini Ernst & Young applying the science of complex adaptive systems to business issues. He has been a Research Director with France Telecom R&D, an R&D Engineer with Cadence Design Systems, Lowell, MA, and the Interval Research Fellow at the Santa Fe Institute. He is Co-Editor-in-Chief of the journals *Advances in Complex Systems* (World Scientific) and the ACM *Transactions on Adaptive and Autonomous Systems* (ACM Press) and serves as a member of the editorial board of several scientific journals. In addition to *Swarm Intelligence* and more than 100 scientific articles, he is the coauthor of *Self-Organization in Biological Systems* (Princeton University Press), a bestselling biology textbook.