Automated Pharmacophore Query Optimization with Genetic Algorithms—A Case Study Using the MC4R System

Lei Jia,[†] Jinming Zou,[§] Sung-Sau So,[‡] and Hongmao Sun*,[‡]

Department of Chemistry, New York University, New York, New York 10003, and Department of Discovery Chemistry, Hoffmann-La Roche, Nutley, New Jersey 07110

Received March 7, 2007

Due to the recent availability of high quality small molecule databases, such as ZINC and PubChem,[1,2] virtual screening is playing an even more important role in identifying biologically relevant molecules in drug discovery campaigns. The success of pharmacophore-based virtual screening (PBVS) relies largely on the accuracy and specificity of the pharmacophore query employed. Deriving a pharmacophore query from a single structure inevitably introduces uncertainty, and the derived query is unlikely to be optimal against every collection of input compounds, especially when it is desired to discriminate among compounds with similar chemical structures. In this study, we present an optimization approach empowered by genetic algorithms (GA) to enhance the accuracy and specificity of a primary pharmacophore query. The example utilized is the human melanocortin type 4 receptor (hMC4R), for which the pharmacophore query was built on the basis of the structure of a rigid cyclic peptide agonist.[3] The optimized query is shown to be capable of identifying 37 positive hMC4R agonists with no false positives from a training set containing 55 agonists and 51 nonagonists. This represents a significant improvement from the initial query which exhibited a 37/32 hit rate. The final, optimized query is challenged with a testing set comprising of 55 hMC4R agonists and 50 nonagonists and achieves a hit rate of 33/8, that improved from 40/31. The impact of GA controlling parameters, including mutation rate, crossover rate, fitness function, population size, and convergence criterion, on performance of optimization are examined and discussed.

INTRODUCTION

An ongoing challenge faced by the pharmaceutical industry is the efficient identification of initial biologically active compounds, generally called "hits", that are of high quality. High throughput screening (HTS) played a major role in hit identification for over a decade, but the rise of HTS has so far done little to enrich the pipelines of drug companies.[4] Efforts to control the rate of decision errors associated with HTS, namely "false positives" and "false negatives", have typically resulted in bottlenecks. There are many series of such errors that can be examined, such as technical or procedural problems during the assay, or the properties of the screened compounds themselves, such as low solubility or poor stability.[5] The time and resource costs of refining the results of HTS must be added to the high costs of the HTS process itself, which include collecting and maintaining screening libraries of thousands to millions of compounds, the reagents for the assay, and maintaining the robot systems. For these reasons, hit generation through HTS is losing popularity to other technologies, especially at the smaller pharmaceutical companies. Under these circumstances, the value of virtual screening (VS) has become more and more recognized. VS offers not only economic advantages but also a way to access millions of commercially available druglike small molecules that are not in corporate inventories.

Two of the major structure-based VS are molecular docking and pharmacophore-based approaches. Difficulties in handling the flexibility of the target protein, and the high sensitivity of the Lennard-Jones-like scoring function against interatomic distances, limit the application of the docking approach. On the other hand, pharmacophore-based virtual screening (PBVS) demonstrates great potential for identifying hits from virtual libraries and can be applied when the experimental structure of the target is not available. PBVS is based on pharmacophore queries that summarize the chemical features possessed by a biologically active compound or shared by a number of such compounds and searches for hits whose pharmacophores align with part or all of these predefined features. Two key factors that affect the performance of a PBVS are the accuracy of the pharmacopheric query and the quality of the conformational database of small molecules. The quality of a conformational database is defined by its overall size and by the completeness of the conformational ensemble generated for each small molecule. Small molecule databases, such as ZINC, contain multiple millions of commercially available compounds which have been passed through filters to remove problematic molecules and which have undergone other quality control protocols to correct the chirality and protonation state of certain atoms.[1,6] The program Omega has proven to be an efficient and powerful tool for enumeration of conformations.[6,7] Therefore, the quality of conformational databases is no longer the biggest hurdle for PBVS. On the other hand, little has been discussed on how to assess or improve the quality of a pharmacopheric query. In this study, we illustrate

* Corresponding author phone: (973)562-3870; fax: (973)235-6084; e-mail: Hongmao.sun@roche.com.
[†] New York University.
[‡] Hoffmann-La Roche.
[§] Current address: Locus Pharmaceuticals, Blue Bell, PA.

10.1021/ci700089w CCC: $37.00
Published on Web 06/08/2007

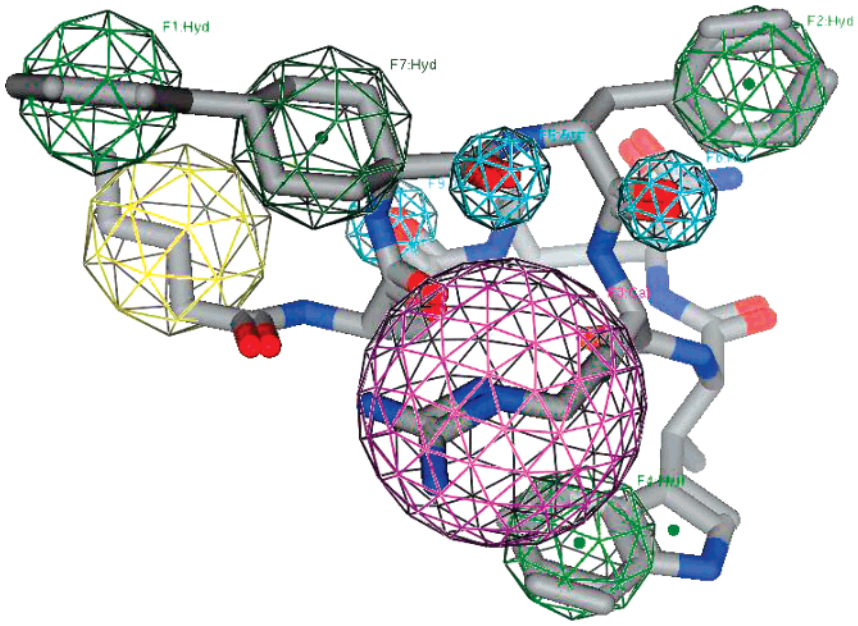

Figure 1. The initial pharmacophore query derived from a cyclic peptide agonist of hMC4R. F1 (Hyd), F2 (Hyd), F3 (Cat), and F4 (Hyd) are essential features, while F5 (Any), F6 (HBA), F7 (Hyd), F8 (HBA), and F9 (HBA) are optional features.

a method to optimize a query against a large training set through the use of genetic algorithms (GA).

A pharmacophore query derived from a single conformation of a bioactive compound is not precise in terms of the coordinates of the chemical features and their tolerance radii, because executing a small perturbation to any feature will result in numerous queries into which the original compound can map. Optimization of a pharmacophore query, therefore, is defined as determining the best combination of the coordinates and the tolerance radii of the chemical features, such that the resulting query can predict the activities of the training compounds. We employed GA to tackle this multiple parameter optimization problem.

GA is a simple and elegant search technique to find approximate solutions to optimization problems, based on the principles of Darwinian evolution.[8−10] It can be considered as a form of evolution that occurs on a computer. In nature, most organisms evolve by means of two primary processes: natural selection and sexual reproduction. The first process determines which members of a population survive to reproduce, and the second ensures mixing and recombination among the genes given to their offspring. GA applies evolutionary procedures including selection, mutation, and crossover to a set of inheritable features so that an optimized set can be approached. Crossover and mutation are genetic operators, which generate solutions that both inherit characteristics from the parents and create new features, while a fitness function determines which individual survives to the next generation, thus defining the direction of selection and optimization. GA has been successfully applied to many problem domains,[11−15] including drug discovery.[16−21]

As a test subject for our optimization procedure, we utilized a pharmacophore query of human melanocotin-4 receptor (MC4R),[22,23] a member of the G-protein-coupled receptor (GPCR) superfamily.[24] MC4R is expressed in the central nervous system[25,26] and is a theraputic target for obesity[27−30] and male erectile dysfunction.[31−33] Melanocyte-stimulating-hormone and agouti related protein are a natural agonist and an antagonist, respectively, of MC4R.[34−36] A large number of peptide and small molecule agonists and antagonists that target the MC4R have been designed and synthesized.[37,38] Currently, an experimental structure of MC4R is not available, and therefore, ligand-based approaches, such as pharmacophore modeling, have become the methods for rational drug discovery.

METHOD

Initial Pharmacophore Query. The initial pharmacophore query was constructed on the basis of a cyclic peptide agonist of hMC4R. This peptide, penta-c[Asp-APC-dPhe-Arg-(2S,-3S)-β-methylTrp-Lys]-NH_2, was known to be rigid, potent, and selective.[3] The query, created by MOE Pharmacophore Query Editor,[39] consisted of nine chemical features, among which four features were considered essential, including three hydrophobes (F1, F2, and F4) representing three hydrophobic side chains, APC, dPHE, and TRP, and one positive charge center (F3) representing ARG, and five features were considered optional, including three hydrogen bond acceptors (HBA), a hydrophobe, and a feature of "ANY" (Figure 1). Besides the four essential features, at least one of the five optional features must be present in a hit.

Training Set and Testing Set. The number of both agonists and antagonists of hMC4R has expanded dramatically in recent years. These newly published hMC4R related compounds comprise a well-suited training set and allow an opportunity to refine and improve our pharmacophore query.

A preliminary database consisted of 211 small molecules for which biological assay results were available. The EC_{50} values to hMC4R were used as classification criteria—molecules with EC_{50} smaller than 100 nM are entered as an active entry and those with EC_{50} larger than 1000 nM are entered as an inactive entry. This yields a collection of 110 active molecules and 101 inactive molecules. The molecules were built with formal charges assigned to corresponding atoms under a neutral aqueous solution condition followed

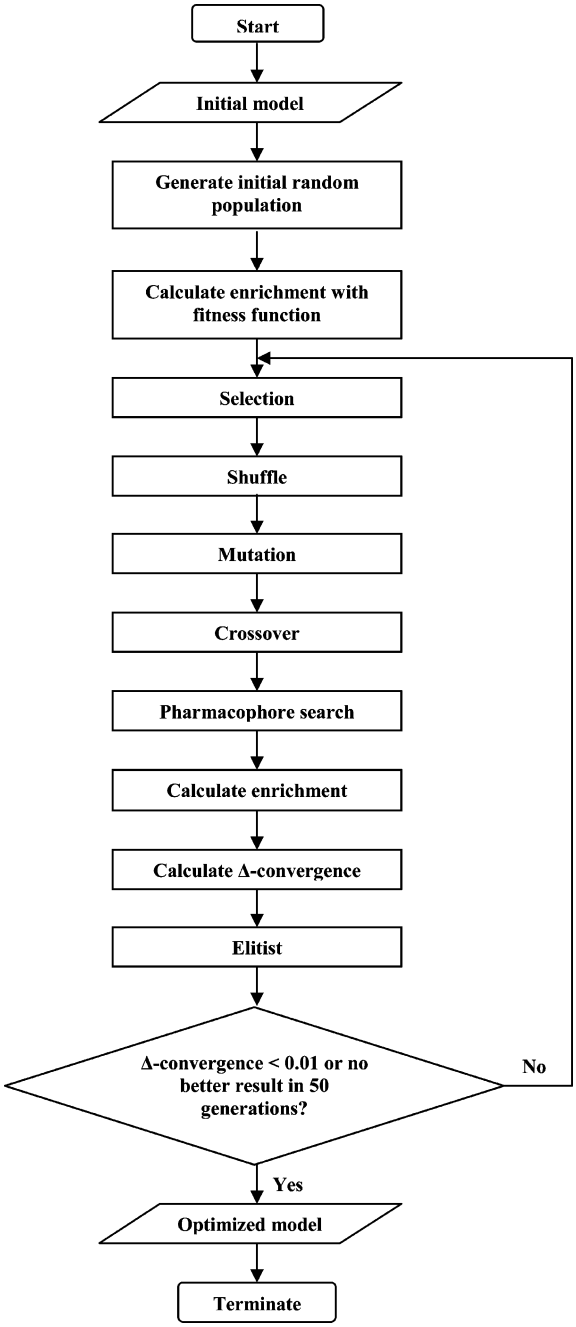

Figure 2. The flowchart of the genetic algorithm.

by energy minimization in MOE using the MMFF94s force field.[40] The data set was randomly split into a training set containing 55 active and 51 inactive molecules and a testing set containing 55 active and 50 inactive molecules.

The program Omega[41] was employed to generate conformers for each molecule in the data sets. The MMFF94s force field was applied for energy calculations. The cutoff energy was set to 5 kcal/mol, as suggested from our previous work.[6] To allow the generation of conformers for all the compounds in the databases, the maximum number of rotatable bonds and flexible rings systems were expanded to 40 and 10, respectively. A heavy-atom root-mean-square deviation (rmsd) of 0.6 Å was used to determine the acceptance of newly generated conformers. A maximum of 400 low energy conformers was allowed for each molecule. By this procedure, we obtained a training set with 34 369 conformers and a testing set with 34 424 conformers.

Pharmacophore Query Optimization with Genetic Algorithms. An application based on GA was developed in house in the C programming language to carry out the pharmacophore query optimization. As indicated in the flowchart (Figure 2), the optimization process started with generating an initial population of individual queries. They were created by executing a random perturbation on the initial 9-point pharmacophore query. Each individual query was then used to screen the compounds in the training set, and each query was assigned a numerical evaluation of its merit by a fitness function. Nine different fitness functions were tested, in an attempt to balance the capability of finding more true positives and eliminating false positives. In the end, the quadratic fitness function (eq 5) outperformed the other eight fitness functions.

Once all individual queries in the population had been evaluated, their fitness values were used as the basis for selection. Selection and inheritance are two different forces. Selection is implemented by eliminating low-fitness individuals, while inheritance is implemented by making multiple copies of high-fitness individuals. Roulette selection was applied in our study, where the chance of an individual query surviving to the next generation was proportional to its fitness, reflecting the concept of "survival of the fittest". The individual solutions in the generation following the selection were shuffled to avoid any position dependency.

Mutation and crossover are both genetic operators to increase genetic diversity. Mutation is the altering of one or more gene values in a feature set from its initial state, which may result in new solutions. Mutation helps to prevent the searching from stagnating at any local optimum. Crossover combines two parent chromosomes to produce a couple of daughter chromosomes. The idea behind crossover is that the new chromosome may be better than both of the parents if it takes the best characteristics from each of the parents.

The nine fitness functions which were designed and examined in this study were as follows:

$$\text{enrichment} = \frac{p}{n+1} \tag{1}$$

$$\text{enrichment} = \frac{p}{n+1} - \frac{n}{t+1} \tag{2}$$

$$\text{enrichment} = \frac{p^2 + (t_n - n)^2}{n^2 + (t_p - p)^2} \tag{3}$$

$$\text{enrichment} = \frac{p}{\sqrt{n+1}} \tag{4}$$

$$\text{enrichment} = \frac{p^2}{n+1} \tag{5}$$

$$\text{enrichment} = \frac{t^2 - n^2}{n+1} \tag{6}$$

$$\text{enrichment} = p^2 - n^2 \quad (7)$$

$$\text{enrichment} = \frac{p^3}{n+1} \quad (8)$$

$$\text{enrichment} = \frac{p}{\log(n+1)+1} \quad (9)$$

where p, n, and t are the numbers of active, inactive, and total hits, respectively, t_p is the total number of active molecules, and t_n is the total number inactive molecules in a data set.

During reproduction, random mutation and one-point crossover were allowed with different rates. For mutation, both the position and tolerance radius of a feature were allowed to be randomly changed. The evolution process was terminated after one of the two criteria was met, i.e., either the delta convergence was less than 0.01 or no higher enrichment was reached in n generations ($n = 20$ or 50). A delta convergence was defined as the following:

$$\text{delta convergence} = \sqrt{\left|\frac{\sum_1^{\text{pop}} (\text{enrichment})^2}{\text{pop}} - \left(\frac{\sum_1^{\text{pop}} \text{enrichment}}{\text{pop}}\right)^2\right|}$$

where pop is the size of a population. In addition, an elitist strategy was applied to prevent the current best query from being destroyed through mutation and crossover until a better solution emerged.

MOE Pharmacophore Search. Pharmacophore query construction and pharmacophore mapping were carried out by using MOE.[39] The Polar-Charged-Hydrophobic (PCH) scheme was selected for annotating the ligands. To accelerate the searching process, a molecule was reported as a hit, and the searching process was terminated for this molecule, once a conformation of the molecule was identified as a hit. The searching speed was further enhanced by calculating the ligand annotations for all molecules beforehand.

Computations were carried out on an HP workstation xw8000 with a 3.06 GHz CPU.

RESULTS

Investigating the Efficiency of the GA Optimization. To enhance the efficiency and performance of a GA optimization, several controlling parameters need to be tuned, including mutation rate (2%), crossover rate (80%), population size (50), convergence criteria (20 generations), and fitness function ($p^2/(n+1)$). Rather than exploring the best combination of these controlling parameters through design of experiment (DOE), we examined the effects of each individual parameter on the performance of GA, while the other parameters were kept at their original values as shown in the parentheses (Table 1 and Figure 3).

Fitness Function. The direction of an optimization process using GA is determined by its fitness function. Once a problem is transformed from problem domain to model domain, a GA cannot alter the problem itself. Instead, it can only recognize a good solution, defined by a fitness function, when one appears. Here the goal is to identify the best query to maximize the number of real positives while minimizing the false positives in a training set. Nine different functions were designed and examined, from the simplest fitness function of $p/(n+1)$, where $(n+1)$ was utilized to avoid zero as a denominator, to more sophisticated functions, such as $p/(\log(n+1)+1)$. The nine different fitness functions

Table 1. Hits, Enrichment, and Running Time under Different Testing Conditions[a]

conditions	active/inactive hits	enrichment
Mutation Rate		
0.1%	34/13	82.6
0.5%	22/5	80.7
2%	40/13	114.3
6%	29/0	841.0
10%	22/0	484.0
Crossover Rate		
60%	46/27	75.6
80%	40/13	114.3
100%	22/2	161.3
Population Size		
50	40/13	114.3
200	33/4	217.8
400	32/3	256.0
Converge Generation		
20	40/13	114.3
50	29/0	841.0

[a] We investigated the following conditions: 1. mutation rate (2%), 2. crossover rate (80%), 3. population size (50), 4. converge steps (20), 5. chemical feature mutation (off). One criterion is changed at a time while keeping other criteria as the value shown in parentheses. ($p^2/(n+1)$) was used as the fitness function.

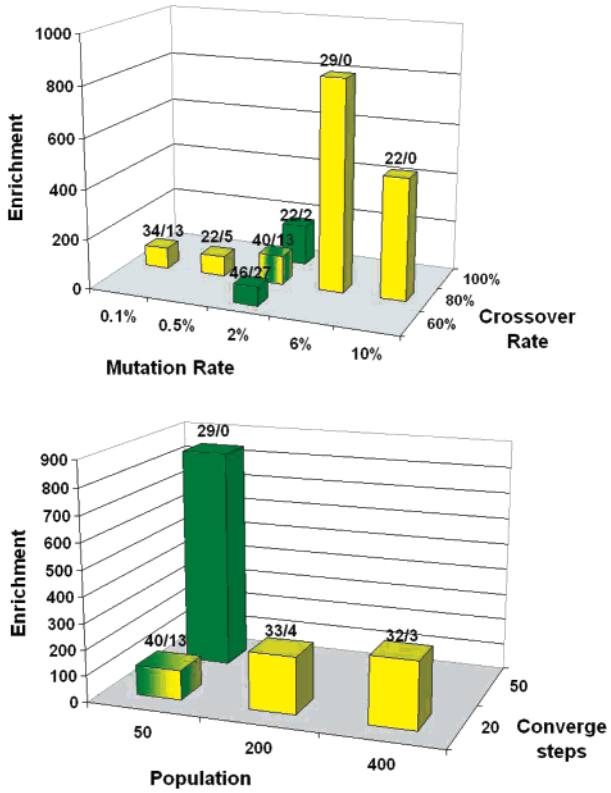

Figure 3. Cross comparison of the enrichments with (A) different mutation rates and crossover rates. Comparisons are under such conditions: population size = 50, converge steps = 20, (B) different population size and convergence steps. Comparisons are under such conditions: mutation rate = 2%, crossover rate = 80%. Chemical feature mutation = off, fitness function = $p^2/(n+1)$. The positive/negative hit numbers are shown on the top of each bar.

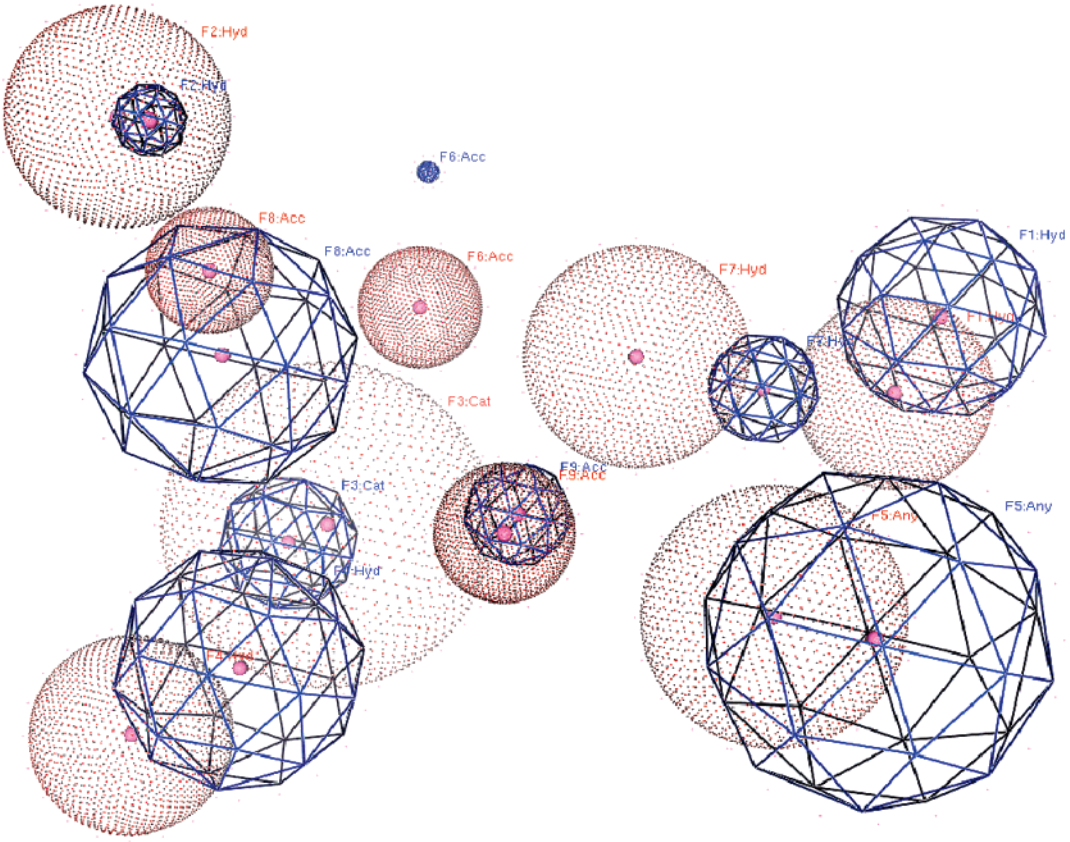

Figure 4. Superimpose of the initial and finally optimized pharmacophore queries. The initial query was represented in red-dotted spheres, and optimized query in blue lined polyhedrons. The chemical features are marked by the corresponding spheres.

were tested under two different mutation rates, 0.5% and 2%. When positive and negative have both the first order on the nominator and denominator, false positives were successfully controlled to a minimal level but at a cost of losing a number of real hits. Increment of the order of positive on the nominator immediately elevates the number of real hits; however, false positives increased dramatically at the same time. Among all nine fitness functions, $p^2/(n + 1)$ yielded the best result with a 22/5 hit rate under a 0.5% mutation rate and a 28/1 hit rate under a 2% mutation rate (Table 1).

Mutation and Crossover Rates. Setting $p^2/(n + 1)$ as the fitness function, other parameters were examined sequentially. Increasing the mutation rate from 0.1% to a moderate number of 6.0% resulted in continuous improvement of both enrichments and hit rates. At a 6.0% mutation rate, a hit rate of 29/0 was achieved, improved from a hit rate of 34/13 with a 0.1% mutation rate, 22/5 with a 0.5% mutation rate, and 40/13 with a 2.0% mutation rate. A mutation rate higher than 6.0% caused the loss of real hits. Increasing the crossover rate from 60% to 80% and to 100% resulted in better enrichments, but a 100% crossover rate exhibited a great reduction of both real and false positives, giving a hit rate of 22/2 (Table 1 and Figure 3A).

Population Size and Convergence Criterion. At a mutation rate of 2.0% and crossover rate of 80%, a substantial improvement was observed by increasing the population size from 50 to 200 (Table 1). Further increasing the population size from 200 to 400 did not create a significant improvement but consumed much longer computational time. A less strict convergence criterion, such as terminating the searching if no better solution emerged in 20 generations, made the process more efficient but at the risk of missing the opportunity to find better queries. A more strict convergence criterion, terminating searching only when no better result was formed in 50 generations, resulted in a much better query in terms of enrichment, a change from 114 to 841, but at the cost of more generations to converge and thus more computing time (Table 1 and Figure 3B).

Automatic Pharmacophore Query Search under the Optimized GA Conditions. After examining the above-mentioned running conditions of the GA, we settled upon an optimized GA running condition combination with a fitness function of $p^2/(n + 1)$, a mutation rate of 6.0%, a crossover rate of 100%, a population size of 200, and a converge generation of 50, to optimize the initial hMC4R pharmacophore query. An optimized pharmacophore query was able to achieve a hit rate of 37/0 after 95 h and 20 min computing on a single CPU of 3.06 GHz. The final query demonstrated a significant improvement from the initial query which had a hit rate of 37/32. Using this optimized pharmacophore query to search the testing set, we obtained a hit rate of 33/8, improved from a hit rate of 40/31 by the initial query.

Optimized Pharmacophore Query. Figure 4 illustrates the alignment of the optimized pharmacophore query with the initial query. Among the four essential chemical features, the tolerance radius of F2 (Hyd) and F3 (Cat) were significantly decreased, that of F4 (Hyd) was moderately increased, and that of F1 (Hyd) remained the same size. The coordinates of all the features changed. Reduced tolerance radius of a chemical feature reflects the increased specificity of that feature.

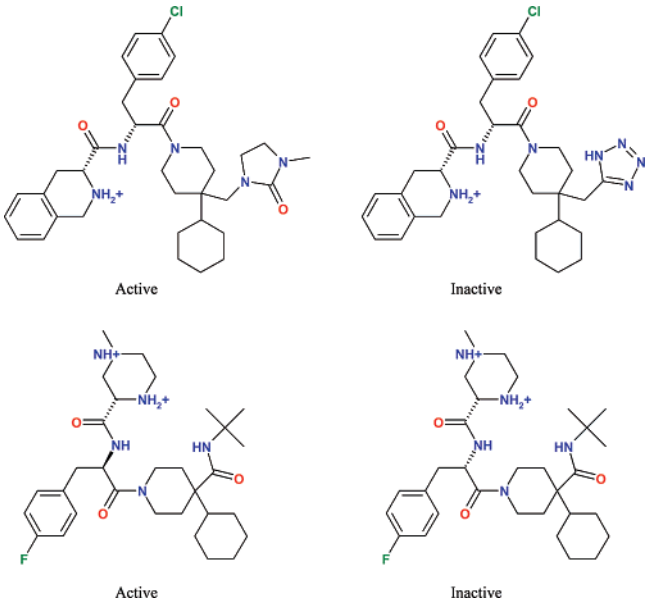

Figure 5. Structural comparison of a pair of the active and inactive molecules in the training set. Molecules are structurally similar but with different biological activities.

DISCUSSION

No matter how accurate a template structure is, the process of converting a structure into a pharmacophore query introduces uncertainties. Multiple valid queries can be created from the same structure by mutating the coordinates and tolerance radius of each of the chemical features or excluded volumes. It is of high value to design a method to refine an original query built from a single structure into an accurate and specific one capable of effectively discriminating positives from structurally similar negatives in a training set. A systematic or exhaustive search of all possible queries is impossible in this regard, instead, we employed GA to search for an optimal arrangement of chemical features in space with suitable tolerance radius.

To optimize a query is to determine the coordinates and tolerance radius of each chemical feature, so that the optimized query is applicable not only to the original template compound but also to a large collection of positive and negative test compounds. As with other optimization approaches, the first challenge for pharmacophore query optimization is to design a suitable training set. To shape a pharmacophore query into a specific and accurate final query requires a well-selected training set, consisting of compounds with similar chemical structures but different biological activities. A training set containing only negative compounds that are structurally dissimilar to the actives will not help to improve the accuracy and specificity of the query to be optimized. For example, assume that a query requires a positive charge center as an essential feature, then a training set containing only neutral compounds as negatives will contribute little to query improvement, because no matter how to adjust the coordinates and tolerance radius of the features in the query, it will be classified as negative.

The guideline for designing the training set and testing set in this study was to select active and inactive compounds that were structurally similar (Figure 5). If the optimized query is capable of discerning the subtle variations in structures among active and inactive compounds, the query is considered accurate and specific. It is not trivial for a single query to distinguish close analogues, because each compound is represented by hundreds of conformations, and if any one of the conformations of an inactive compound can map with the query it will be misclassified. A related issue is the energy cutoff in conformation enumeration. Considering the Boltzmann distribution of energy levels and ease of converging, the energy cutoff was set to a relatively low value of 5.0 kcal/mol. It is not desirable to identify a hit in a high-energy conformation, because the energy required to shift the population toward this conformation will result in a lower binding affinity. Our results indicated that the final query was able to distinguish the active compounds from the inactive analogues as represented by their ensemble of low-energy conformations.

Table 2. Hits and Running Time under Different Fitness Functions[a]

	fitness function			
	hits (active/inactive)		time[b]	
mutation rate	0.5%	2%	0.5% (min)	2% (min)
1	5/0	9/0	135	720
2	3/0	13/0	380	900
3	37/14	41/10	990	1260
4	14/1	37/9	765	960
5	22/5	28/1	530	920
6	47/38	48/39	270	650
7	40/16	45/32	690	660
8	47/39	37/10	660	940
9	48/42	43/22	730	750

[a] The order of fitness functions is following the Method. [b] Running time was reflected to the same computing machine.

GA is an efficient and powerful multiparameter optimization algorithm. A significantly improved solution generally emerges from a few generations of evolution. A long existing challenge for GA applications is the characterization of a solution as a single number, which might be extremely difficult for complicated systems. In this study, a fitness function was designed to evaluate the quality of each pharmacophore query. The fitness function should reflect the balance between sensitivity of the query—its ability to identify true positives, and specificity—its ability to eliminate false positives. In our design, specificity of a query was emphasized more than sensitivity, because a less specific pharmacophore query may generate a large number of false positives on screening a large database like ZINC and consequently cause a waste of resources. On the other hand, missing some true positives is affordable (actually, no screening technology can claim to be free from false negatives) if the purpose of screening is to identify at least a few hits to supply a good starting point for a new project. Fitness function 6 was the most sensitive one, capable of identifying 48 agonists at a 2.0% mutation rate, while functions 1 and 2 were the most specific, giving no false positives (Table 2). Function 5 illustrated the best balance between sensitivity and specificity, exhibiting a hit rate of 22/5 at a 0.5% mutation rate and 28/1 at a 2.0% hit rate.

Tuning the controlling parameters is the method by which one manipulates the genetic diversity along the GA process. A slow change of diversity indicates low efficiency due to a slow convergence, while a quick drop of genetic diversity risks causing stagnation in a local optimum.

Mutation and crossover increase diversity, while natural selection uses diversity in a population to produce adaptation. To maximize the diversity of the original population, each individual query was created by randomly disturbing both coordinates and tolerance radius of each feature in a query to be optimized. Increment of the population size expanded the searching space by increasing the overall diversity, but, at certain point, redundancy of randomly created queries started to build up and ultimately cancelled out the benefits of increased population size. In this study, a population size of 200 seemed sufficient to explore the solution space efficiently.

Mutation increases the genetic diversity of a population, thus preventing early convergence. A search at a low mutation rate tends to be less efficient due to the large possibility of being trapped at local optima, while too high a mutation rate turns a GA search into a random search, which is also inefficient. According to our results, a mutation rate around 6.0% yielded the best results for query optimization.

Initiating from a diverse population, a GA converges when most of the population is identical. How fast a GA will converge is determined by multiple factors, including the choice of fitness function, mutation rate, and the definition of convergence. Two different convergence criteria were defined in this study. Delta convergence is achieved when the diversity of a generation is minimized. In this study, delta convergence was reached only at low mutation rate GA searches. Considering the high demand of computational resources to assess the fitness of each individual query and the complexity of the fitness landscape, a less tight converging criterion was applied in this study—a GA was considered converged if no better solution emerged in a number of generations. If computational resource is not a limitation, a GA can theoretically converge to a definite value that is close to the global optimum. However, even if the resources are limited, a significantly improved solution can be achieved efficiently, as demonstrated in this study.

Selection supplies the pressure for a GA to converge. A different selection scheme assumes a different path and efficiency of convergence. In the Roulette selection scheme, every solution has an opportunity to survive, but the better solutions with higher normalized fitness are given greater chances to survive to the next generation.

The last decision to make in a GA is when to terminate the search. In an ideal situation, the population will converge to a definite solution after certain generations of evolution, but in reality, especially for a complex system where there exist multiple degenerate solutions, a delta convergence can be hardly reached at a moderate mutation rate, as shown in this study. At the same time, calculation of fitness of an individual query is time-consuming, which involves mapping of nearly 35 000 conformations in the training set; therefore, an alternative criterion should be adopted to terminate a GA search, i.e., a GA will be stopped if no better solution appears in a certain number of generations. Strictly speaking, this criterion has nothing to do with convergence, thus it is interesting to investigate how convergent or diverse the final generation is. The final generation contained the 2 best pharmacophore queries of a 37/0 hit rate, the 22 second best queries of a 36/0 hit rate, and the 53 queries of a 35/0 hit rate (Figure 6). Three randomly selected queries of a 36/0 hit rate, together with the 2 best pharmacophore queries, were compared in a pairwise manner—it turned out that they were very similar to each other—chemical features were superimposed very well except for two nonessential chemical features. When these 5 queries were applied to search the testing set, the 2 best queries achieved a 33/8 hit rate and the 3 second best queries gave a 29/6 hit rate. These results indicated that the "concentration" of good solutions was reasonably high when the search was terminated.

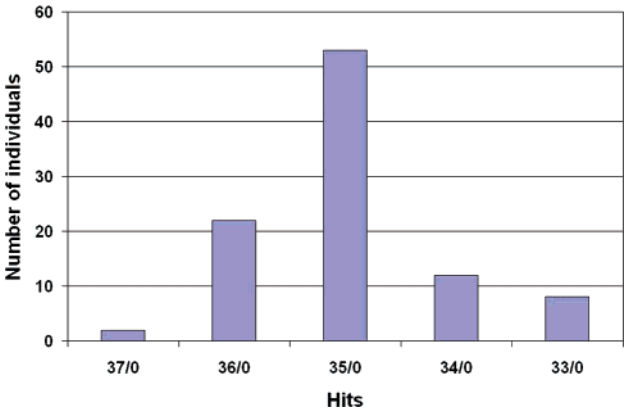

Figure 6. Histogram of individual distribution query in the last generation. The first five best groups were shown.

SUMMARY

The process of deriving a pharmacophore query from a structure will inevitably introduce uncertainty. To minimize this uncertainty, we developed an application to automatically optimize a pharmacophore query with GA. The case study on hMC4R ligands demonstrated that the optimized pharmacophore query had improved significantly, and it was able to discriminate among structurally similar hMC4R agonists and nonagonists. By automatically adjusting the position and tolerance radius of each pharmacopheric feature in a query, the program reached a final model capable of identifying 37 positive hMC4R agonists with no false positives from the training set. This represented a significant improvement from the initial query which exhibited a 37/32 hit rate. The final query was challenged with a testing set and achieved a hit rate that improved to 33/8 from 40/31.

To obtain acceptable performance of a GA, many controlling parameters must be adjusted. Taking the cost of both computing resources and searching performance into consideration, a general guideline based on our results is that higher mutation and crossover rates allow a larger space to be sampled, but the search becomes harder to converge as evaluated by the rmsd of the genes in the population. Nevertheless, an optimal combination can be achieved. Population size can also affect the performance of a GA. A reasonable population size of 200 yielded satisfactory searching efficiency in this study. Tuning GA controlling parameters could also be done by utilizing another GA—a self-adaptive algorithm called meta-GAs.[42]

ACKNOWLEDGMENT

The authors would like to thank Dr. David Fry, for critical reading of the manuscript. L.J. appreciates financial support for his summer internship at Hoffmann-La Roche Inc. in Nutley.

Supporting Information Available: Active and inactive training and test sets. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Irwin, J. J.; Shoichet, B. K. ZINC−a free database of commercially available compounds for virtual screening. *J. Chem. Inf. Model* **2005**, *45*, 177−182.
(2) PubChem Project. http://pubchem.ncbi.nlm.nih.gov/ (accessed March 2007).
(3) Sun, H.; Greeley, D. N.; Chu, X. J.; Cheung, A.; Danho, W. et al. A predictive pharmacophore model of human melanocortin-4 receptor as derived from the solution structures of cyclic peptides. *Bioorg. Med. Chem.* **2004**, *12*, 2671−2677.
(4) Dove, A. Screening for content−the evolution of high throughput. *Nat. Biotechnol.* **2003**, *21*, 859−864.
(5) Malo, N.; Hanley, J. A.; Cerquozzi, S.; Pelletier, J.; Nadon, R. Statistical practice in high-throughput screening data analysis. *Nat. Biotechnol.* **2006**, *24*, 167−175.
(6) Pandit, D.; So, S. S.; Sun, H. Enhancing specificity and sensitivity of pharmacophore-based virtual screening by incorporating chemical and shape features−a case study of HIV protease inhibitors. *J. Chem. Inf. Model* **2006**, *46*, 1236−1244.
(7) Bostrom, J.; Greenwood, J. R.; Gottfries, J. Assessing the performance of OMEGA with respect to retrieving bioactive conformations. *J. Mol. Graphics Modell.* **2003**, *21*, 449−462.
(8) Holland, J. H. Genetic Algorithms. *Sci. Am.* **1992**, *267*, 66−72.
(9) Forrest, S. Genetic Algorithms - Principles of Natural-Selection Applied to Computation. *Science* **1993**, *261*, 872−878.
(10) Sumida, B. H.; Houston, A. I.; Mcnamara, J. M.; Hamilton, W. D. Genetic Algorithms and Evolution. *J. Theor. Biol.* **1990**, *147*, 59−84.
(11) Cho, S. J.; Sun, Y. X. FLAME: A program to flexibly align molecules. *J. Chem. Inf. Model.* **2006**, *46*, 298−306.
(12) Gillet, V. J.; Willett, P.; Bradshaw, J. Identification of biological activity profiles using substructural analysis and genetic algorithms. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 165−179.
(13) Sheridan, R. P.; SanFeliciano, S. G.; Kearsley, S. K. Designing targeted libraries with genetic algorithms. *J. Mol. Graphics Modell.* **2000**, *18*, 320−334.
(14) Willett, P. Genetic algorithms in molecular recognition and design. *Trends Biotechnol.* **1995**, *13*, 516−521.
(15) Weber, L.; Wallbaum, S.; Broger, C.; Gubernator, K. Optimization of the Biological-Activity of Combinatorial Compound Libraries by a Genetic Algorithm. *Angew. Chem., Int. Ed. Engl.* **1995**, *34*, 2280−2282.
(16) Parrill, A. L. Evolutionary and genetic methods in drug design. *Drug Discovery Today* **1996**, *1*, 514−521.
(17) Douguet, D.; Thoreau, E.; Grassy, G. A genetic algorithm for the automated generation of small organic molecules: Drug design using an evolutionary algorithm. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 449−466.
(18) Pegg, S. C. H.; Haresco, J. J.; Kuntz, I. D. A genetic algorithm for structure-based de novo design. *J. Comput.-Aided Mol. Des.* **2001**, *15*, 911−933.
(19) Schneider, G.; Lee, M. L.; Stahl, M.; Schneider, P. De novo design of molecular architectures by evolutionary assembly of drug-derived building blocks. *J. Comput.-Aided Mol. Des.* **2000**, *14*, 487−494.
(20) Taha, M. O.; Qandil, A. M.; Zaki, D. D.; AlDamen, M. A. Ligand-based assessment of factor Xa binding site flexibility via elaborate pharmacophore exploration and genetic algorithm-based QSAR modeling. *Eur. J. Med. Chem.* **2005**, *40*, 701−727.
(21) Vainio, M. J.; Johnson, M. S. McQSAR: A multiconformational quantitative structure-activity relationship engine driven by genetic algorithms. *J. Chem. Inf. Model.* **2005**, *45*, 1953−1961.
(22) Gantz, I.; Konda, Y.; Tashiro, T.; Shimoto, Y.; Miwa, H. et al. Molecular cloning of a novel melanocortin receptor. *J. Biol. Chem.* **1993**, *268*, 8246−8250.
(23) Mountjoy, K. G.; Robbins, L. S.; Mortrud, M. T.; Cone, R. D. The cloning of a family of genes that encode the melanocortin receptors. *Science* **1992**, *257*, 1248−1251.
(24) Holder, J. R.; Haskell-Luevano, C. Melanocortin ligands: 30 years of structure-activity relationship (SAR) studies. *Med. Res. Rev.* **2004**, *24*, 325−356.
(25) Alvaro, J. D.; Tatro, J. B.; Quillan, J. M.; Fogliano, M.; Eisenhard, M. et al. Morphine down-regulates melanocortin-4 receptor expression in brain regions that mediate opiate addiction. *Mol. Pharmacol.* **1996**, *50*, 583−591.
(26) Mountjoy, K. G.; Mortrud, M. T.; Low, M. J.; Simerly, R. B.; Cone, R. D. Localization of the melanocortin-4 receptor (MC4-R) in neuroendocrine and autonomic control circuits in the brain. *Mol. Endocrinol.* **1994**, *8*, 1298−1308.
(27) Huszar, D.; Lynch, C. A.; Fairchild-Huntress, V.; Dunmore, J. H.; Fang, Q. et al. Targeted disruption of the melanocortin-4 receptor results in obesity in mice. *Cell* **1997**, *88*, 131−141.
(28) Ho, G.; MacKenzie, R. G. Functional characterization of mutations in melanocortin-4 receptor associated with human obesity. *J. Biol. Chem.* **1999**, *274*, 35816−35822.
(29) Forbes, S.; Bui, S.; Robinson, B. R.; Hochgeschwender, U.; Brennan, M. B. Integrated control of appetite and fat metabolism by the leptin-proopiomelanocortin pathway. *Proc. Natl. Acad. Sci. U.S.A.* **2001**, *98*, 4233−4237.
(30) Lubrano-Berthelier, C.; Cavazos, M.; Dubern, B.; Shapiro, A.; Stunff, C. L. et al. Molecular genetics of human obesity-associated MC4R mutations. *Ann. N. Y. Acad. Sci.* **2003**, *994*, 49−57.
(31) Van der Ploeg, L. H.; Martin, W. J.; Howard, A. D.; Nargund, R. P.; Austin, C. P. et al. A role for the melanocortin 4 receptor in sexual function. *Proc. Natl. Acad. Sci. U.S.A.* **2002**, *99*, 11381−11386.
(32) Wikberg, J. E. S. Melanocortin receptors: new opportunities in drug discovery. *Expert Opin. Ther. Pat.* **2001**, *11*, 61−76.
(33) Goodfellow, V. S.; Saunders, J. The melanocortin system and its role in obesity and cachexia. *Curr. Top. Med. Chem.* **2003**, *3*, 855−883.
(34) Oosterom, J.; Garner, K. M.; den Dekker, W. K.; Nijenhuis, W. A.; Gispen, W. H. et al. Common requirements for melanocortin-4 receptor selectivity of structurally unrelated melanocortin agonist and endogenous antagonist, Agouti protein. *J. Biol. Chem.* **2001**, *276*, 931−936.
(35) Ollmann, M. M.; Wilson, B. D.; Yang, Y. K.; Kerns, J. A.; Chen, Y. et al. Antagonism of central melanocortin receptors in vitro and in vivo by agouti-related protein. *Science* **1997**, *278*, 135−138.
(36) Schioth, H. B.; Haitina, T.; Ling, M. K.; Ringholm, A.; Fredriksson, R. et al. Evolutionary conservation of the structural, pharmacological, and genomic characteristics of the melanocortin receptor subtypes. *Peptides* **2005**, *26*, 1886−1900.
(37) Voisey, J.; Carroll, L.; van Daal, A. Melanocortins and their receptors and antagonists. *Curr. Drug Targets* **2003**, *4*, 586−597.
(38) Irani, B. G.; Holder, J. R.; Todorovic, A.; Wilczynski, A. M.; Joseph, C. G. et al. Progress in the development of melanocortin receptor selective ligands. *Curr. Pharm. Des.* **2004**, *10*, 3443−3479.
(39) MOE. *Molecular Operating Environment* (*MOE*); Chemical Computing Group, Inc.: Montreal, Quebec, Canada.
(40) Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **1996**, *17*, 490−519.
(41) Omega. *Omega*; OpenEye Scientific Software: Santa Fe, NM.
(42) Clune, J.; Goings, S.; Punch, B.; Goodman, E. Investigations in meta-GAs: Panaceas or pipe dreams? *Proceedings of the 2005 workshops ongenetic and evolutionary computation*; ACM Press: Washington, DC, 2005; pp 235−241.

CI700089W