ARTICLES

FOG: Fragment Optimized Growth Algorithm for the *de Novo* Generation of Molecules Occupying Druglike Chemical Space

Peter S. Kutchukian, David Lou, and Eugene I. Shakhnovich*

Department of Chemistry and Chemical Biology, Harvard University, 12 Oxford Street,
Cambridge, Massachusetts 02138

Received February 5, 2009

An essential feature of all practical *de novo* molecule generating programs is the ability to focus the potential combinatorial explosion of grown molecules on a desired chemical space. It is a daunting task to balance the generation of new molecules with limitations on growth that produce desired features such as stability in water, synthetic accessibility, or druglikeness. We have developed an algorithm, Fragment Optimized Growth (FOG), which statistically biases the growth of molecules with desired features. At the heart of the algorithm is a Markov Chain which adds fragments to the nascent molecule in a biased manner, depending on the frequency of specific fragment-fragment connections in the database of chemicals it was trained on. We show that in addition to generating synthetically feasible molecules, it can be trained to grow new molecules that resemble desired classes of molecules such as drugs, natural products, and diversity-oriented synthetic products. In order to classify our grown molecules, we developed the Topology Classifier (TopClass) algorithm that is capable of classifying compounds, for example as drugs or nondrugs. The classification accuracies obtained with TopClass compare favorably with the literature. Furthermore, in contrast to "black-box" approaches such as Neural Networks, TopClass brings to light characteristics of drugs that distinguish them from nondrugs.

INTRODUCTION

All the components necessary for the success of computer-assisted drug design seem to be in place compared to its early days, such as a vast wealth of macromolecular crystal structures and seemingly unlimited computational resources, positioning it to take an ever increasing place of prominence in the drug discovery process. While a number of successful algorithms have been reported, there is still a great need to develop algorithms that are relevant to practical experimental work. Perhaps the development of *de novo* growth algorithms of small molecules is especially timely, as it has been suggested that there have been no major breakthroughs in the past two years.[1] *De novo* algorithms can be used to generate new molecules that can then be subsequently docked to target molecules, or they can be used to grow potential ligands into the active site of a protein. More generally, they can be used to generate new molecules with desired biological or physical properties such as a large dipole moment.[2] Our goal here was to develop a *de novo* algorithm that would produce molecules occupying a desired chemical space, such as druglike or natural productlike.

De novo methods have been the subject of a number of reviews,[1,3,4] so only features especially relevant to the current work will be highlighted. In all *de novo* growth applications, the key issue is how to focus the generation of new molecules that lie in useful chemical space. The combinatorial space when generating new molecules is vast,[5−9] and when the goal is to develop new therapeutically relevant molecules, it is essential to focus that space on compounds that will bind their target adequately, are synthetically feasible, and possess druglike properties. While most early *de novo* methods focused on shape and energetic complementarity to binding pockets (MCSS,[10] SPROUT,[11−13] LUDI,[14] "linked-fragment approach,[15] GenStar,[16] CONCEPTS,[17] PRO_LIGAND,[18] SMoG[19]), some methods crudely addressed synthetic tractability by penalizing connections between heteroatoms (MCDNLG[20]), only allowing new bonds to form between carbons when linking functional groups together (DLD[21]), only allowing functional groups to be connected to sp^3 carbons (BUILDER v.2[22]), disallowing certain connections between atoms (LEGEND,[23,24] GrowMol,[25] RASSE[26]), or by disallowing certain connections as well as sequences of connections between fragments to avoid generating unstable moieties such as acetals (GroupBuild[27]). *De novo* algorithms which grew specific classes of molecules such as peptides (GROW,[28] LUDI[29]) avoided the problem of developing rules to connect organic fragments by using amino acids as building blocks. Although stand-alone programs exist to offer retrosynthetic routes to grown molecules (LHASA,[30−33] SYNGEN[34−39]) and to additionally score the synthetic difficulty of grown ligands (CAESA,[40] SYLVIA[41]), later methods began to address the synthetic feasibility more carefully when generating molecules (DREAM++,[42] TOPAS,[43] and SYNOPSIS[2]). Covering

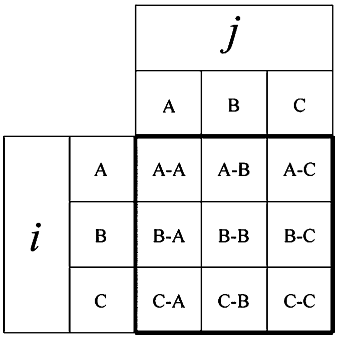
* Corresponding author e-mail: eugene@belok.harvard.edu.

10.1021/ci9000458 CCC: $40.75
Published on Web 06/15/2009

diverse chemical space while adhering to synthetic rules, however, remains problematic.[1] Druglikeness, on the other hand, has only been addressed in a very crude manner by *de novo* methods, for example by only using scaffolds and appendages commonly found in drugs (BOMB[44]) or by applying penalties when the cutoffs implied by the Lipinski "Rule of Five"[45] are violated by grown molecules (LigBuilder[46]). Although often used as a filter to rule out nondruglike compounds from chemical libraries, Lipinski's "Rule of Five" has been shown to classify drugs versus nondrugs with extremely poor—nearly random—accuracy[47] and is inferior to the methods described below.

Sophisticated classification algorithms have been developed in an attempt to recognize druglike molecules in an automated fashion. These can be used to screen virtual libraries for druglike molecules or for molecules that might interact with a specific biologic target. Although classification methods such as support vector machines (SVM),[48,49] binary kernel discrimination,[50,51] and learning trees[49] have been explored, the most common approach has been the use of artificial neural networks (ANNs). The feed-forward with back-propagation of error method has been employed most often and has been used to separate drugs from nondrugs,[48,52−54] serine protease actives from inactives,[55] GPCR actives from inactives,[56,57] easily synthesized compounds from synthetically difficult compounds,[48] and one database source of compounds from another database source.[58] Other ANN strategies have been applied as well, such as Bayesian networks to separate drugs from nondrugs[59] or CNS actives from inactives,[60] as well as probabilistic networks to classify drugs in respect to their biological targets.[61] ANNs have proven extremely accurate in separating classes of compounds with accuracies usually on the order of ∼80−90%. Unfortunately, ANNs are often viewed as a "black box" approach, and although important molecular descriptors can be identified in the separation of compounds,[53,55,60] specific features that add to or detract from a molecule's druglike character are not easily extracted from these methods.

The key advantage of ANNs over simpler linear approaches is that they are often more accurate,[59] while the main advantage of linear methods is that structural features responsible for the classification of a compound can be ascertained. For example, when identifying antibacterial compounds, although ANN outperformed linear discriminant analysis (LDA) in accuracy, key features of the antibacterial compounds were identified from the LDA approach.[62] Engkvist et al., on the other hand, used statistical biases in substructures (SUBSTRUCT) to accurately separate CNS active from nonactive drugs and reported similar accuracies to an ANN method.[63] Although their linear SUBSTRUCT approach was slower than an ANN, they were able to elucidate features such as aromatic rings, tertiary nitrogens, and halogens such as fluorine and chlorine that were more frequently found in CNS drugs versus other drugs, which they were not able to extract from an ANN.[63] Hutter recently reported a linear method based on the statistical biases of atom pair distributions in drugs versus nondrugs.[64] He was able to elucidate atom biases in drugs versus nondrugs, such as tetrahedral carbons being over-represented in drugs and aromatic carbons being over-represented in nondrugs.[64]



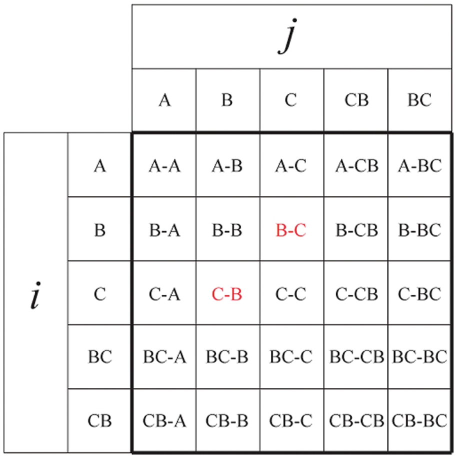
		j		
		A	B	C
i	A	A-A	A-B	A-C
	B	B-A	B-B	B-C
	C	C-A	C-B	C-C

Figure 1. Simple growth scheme where fragments A through C can be joined to fragments A through C.

We have developed an algorithm, FOG (Fragment Optimized Growth), which grows molecules by adding fragments to a nascent molecule in a statistically biased manner. FOG generates synthetically tractable molecules, as deemed by synthetic chemists and synthetic accessibility prediction software.[41] In addition, it is capable of being trained to grow new molecules that are similar in their chemical and topological features to a desired class of chemicals, such as natural products (NP), diversity-oriented synthesis (DOS) products, or drugs. For example, if trained on a NP database, our algorithm would be able to generate new natural productlike compounds with features such as polyphenol moieties that are characteristic of many of the chemicals in the authentic database, while being devoid of moieties like triazole rings which might be found in DOS compounds. In order to validate that our algorithm produced compounds occupying a desired chemical space, we developed an algorithm capable of classifying compounds, for example as drugs or nondrugs. Our classification algorithm, TopClass (Topology Classifier), exploits the statistical bias of fragments and fragment connections (2D metrics) as well as coupled 1D metrics (such as number of atoms and rotatable bonds). The accuracy of TopClass compares favorably with methods reported in the literature.[64] The algorithm is also transparent in the features that it classifies compounds by, which helps bring to light salient features of druglike compounds in contrast with the "black box" approach of ANNs. We also used more conventional approaches to evaluate our grown molecules such as the Tanimoto dissimilarity between grown molecules and training databases of chemicals.

METHOD

1. De Novo Growth Algorithm. *1.1. Calculating Transition Probabilities.* The first step in developing our new algorithm was to collect connectivity statistics for fragments of interest from a database of small molecules. The statistics were then converted into transition probabilities ($P_{i\to j}$), the probability that a fragment i will transition (be connected to) fragment j during small molecule growth. In the simplest scenario, where each fragment is a single atom (represented by a letter in Figure 1), A-B is the same as B-A, and only linear growth is allowed. We see that in order to convert counts (N_{ij}) into transition probabilities that will reproduce those counts, we must divide the counts by the total connection fragment i makes to all other fragments ($\sum_s(N_{is}^D)$). In the following equations we use superscripts to denote the library that we are searching—in this case we use D to denote that we are counting linkages in our training database. This training database D can be made up of rather specific classes



		j				
		A	B	C	CB	BC
i	A	A-A	A-B	A-C	A-CB	A-BC
	B	B-A	B-B	B-C	B-CB	B-BC
	C	C-A	C-B	C-C	C-CB	C-BC
	BC	BC-A	BC-B	BC-C	BC-CB	BC-BC
	CB	CB-A	CB-B	CB-C	CB-CB	CB-BC

Figure 2. Simple growth scheme where fragments A through CB can be joined to fragments A through CB. The red linked fragments are not allowed to grow.

of compounds such as drugs, nondrugs, NP, or DOS, or it can be a more general database such as the ChemBank Bioactives[65] or the NCI open database.[66] We must also take into account that the on-diagonal elements (such as A-A) are 1/2 as likely to grow as the off-diagonal elements (such as A-B). This can be done by dividing the counts for each linkage by the number of times the linkage occurs in an exhaustive 2mer database of fragments (N_{is}^{2mer}), giving us

$$P_{i\rightarrow j} = \left(\frac{N_{ij}^{D}}{\sum_{s}(N_{is}^{D})}\right)\left(\frac{1}{N_{ij}^{2mer}}\right) \qquad (1)$$

The next step in the development of our algorithm was to account for larger fragments that include smaller fragments as part of their substructure (Figure 2). In this case CB is not symmetric, and thus CB-C is not the same as C−CB. For example, an amide fragment (CB) could be divided into a carbonyl (C) and an amine (B) fragment. We found that by accounting for how often linkages occur in an exhaustive 3mer library $N_{ij}^{3,\,mer}$ (every way three fragments can be placed in positions i, $i + 1$, and $i + 2$) we were able to produce connectivity statistics (i.e., the probability that a given fragment is connected to another fragment, see eq 4) that were in agreement with the training database, giving us

$$P_{i\rightarrow j} = \left(\frac{N_{ij}^{D}}{\sum_{s}(N_{is}^{D})}\right)\left(\frac{1}{N_{ij}^{3mer}}\right) \qquad (2)$$

Also note that fragments cannot be joined in such a way that they grow fragments that already exist (red linked fragments in Figure 2). We do this by setting the corresponding transition probabilities to zero. We chose to invoke this rule in our algorithm because by growing fragments already in the database, one would effectively lose information, since our transition probabilities solely depend on the growth fragment and not on what it is connected to. Consider a fragment pool that included amide, amine, and carbonyl fragments. If we allowed the amine to combine with the carbonyl and then grew from the nitrogen, our growth algorithm would be unaware that the nitrogen is now better represented by an amide and would attach moieties to it that might be more likely bonded to an amine nitrogen rather than an amide nitrogen.

The next endeavor was to actually use real organic fragments. In all database searches that were performed, SMARTS[67] strings were used in order to query the desired fragment or substructure. The jcsearch program from ChemAxon was used in all searches.[68] Keeping with the sequential growth of fragments joined by single bonds employed by SMoG[19] as well as a number of other *de novo* algorithms,[25−28,46] fragments are attached to each other by removing a hydrogen from each fragment and subsequently connecting the atoms that were attached to those hydrogens to each other. Fragments are now added, however, in a statistically biased way, depending on the growth fragment. We soon found that certain corrections (*C*) had to be made for double counting (details in the Supporting Information), due to search strings being able to match a database molecule in more than one orientation, although our algorithm would only be able to grow one of these linkages if it were to reproduce the molecule. Taking double counting errors into account when necessary, we now have

$$P_{i\rightarrow j} = \left(\frac{N_{ij}^{D} - C_{ij}^{D}}{\sum_{s}(N_{is}^{D} - C_{is}^{D})}\right)\left(\frac{1}{N_{ij}^{3mer} - C_{ij}^{3mer}}\right) \qquad (3)$$

Using the above equation to obtain transition probabilities, we attempted to reproduce connectivity statistics with a small pool of fragments (methyl, ethyl, amine, hydroxyl, thiol, carbonyl, carboxyl, and amide). Some fragments must be represented more than once using our method since each unique growth site on a fragment has its own transition probabilities associated with it. In a preliminary study, we trained our algorithm on the ChemBank Bioactives Database (2004_05_01)[65] and grew a library of 10,000 linear molecules, each 5 fragments long. During growth, a random fragment is initially selected. Then, based on the growth fragment's transition probabilities, a second fragment is added to it. The newly added fragment subsequently becomes the growth fragment, and the process repeats itself until 4 fragments have been added to the initial fragment. This is in a sense a Markov Chain, where the selection of the next state (fragment *j*) depends on the current state (fragment *i*). Fragments that are more likely to be connected to the growth fragment in the database are also more likely to be selected during growth. To evaluate how well our algorithm reproduced the probabilities that specific fragments would be bonded to each other in grown databases, we compared these probabilities (for example, how likely an amide nitrogen is bonded to a methyl fragment) with the probabilities obtained from the training database. We defined our connectivity propensities as

$$P_{ij} = \left(\frac{N_{ij}^{D}}{\sum_{s}(N_{is}^{D})}\right) \qquad (4)$$

The connectivity statistics we obtained from our grown library were in excellent agreement with the statistics obtained from the training database (R^2=0.99 for all points, R^2=0.98 when values <0.1 were removed, Figure 3). As expected, libraries grown with no statistical biases as a

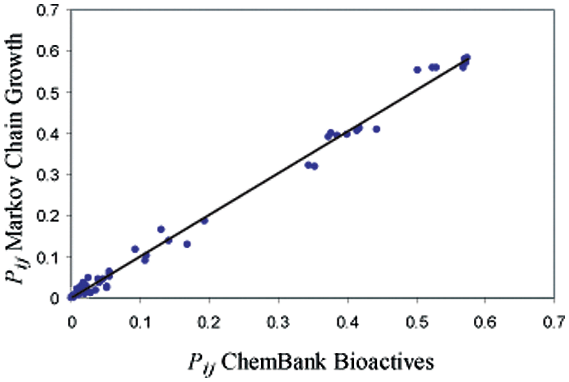


Figure 3. The probability that fragment *i* is connected to fragment *j* (P_{ij}, eq 4) in a database of linear molecules grown with a Markov Chain versus P_{ij} of ChemBank Bioactives used in training the Markov Chain. Only eight fragments were used for growth.

control correlated poorly with the connectivity statistics obtained from the Chembank Bioactives Database (R^2=0.25).

As a next step, we added more fragments to our algorithm such as rings (Table S1). In the current method, rings are only generated by adding ring fragments to the growing molecule, and two nonring fragments already connected to a growing molecule cannot connect to form a ring. We also allowed our molecules to branch. The branching probability $P(B)$—defined as the likelihood that a fragment will grow off of a fragment that already has at least two fragments connected to it—is controlled by the user, allowing one to bias the growth of small molecules ranging from linear to highly branched. With the branching mode present, it was not as straightforward to correct our counts by dividing them by the counts from an exhaustive library (as in the previous two cases where we searched a 2mer or 3mer library, respectively). Thus, in order to construct a normalization library we grew a 5mer library (10,000 molecules) employing the "rules" of growth (for example, fragments could not be connected if they grew a fragment already present in the pool of fragments) but without any statistical biases and with a branching probability $P(B)$ of 0.5. We then searched this library for the likelihood that linkages will occur when no statistical biases are present and corrected for double counting ($N_{ij}^{5\ mer} - C_{ij}^{5\ mer}$). Then, to obtain transition probabilities we used

$$P_{i\to j} = \left(\frac{N_{ij}^{D} - C_{ij}^{D}}{\sum_{s} (N_{is}^{D} - C_{is}^{D})}\right)\left(\frac{1}{N_{ij}^{5mer} - C_{ij}^{5mer}}\right) \qquad (5)$$

It is important to note that if a similar algorithm is implemented in another *de novo* method, the 5mer library used for normalization should be grown with all the biases of that method (i.e., grown into a sample active site, using the fragments available in that method, etc.) but without a statistical bias in place.

1.2. Growth. During growth an initial fragment is chosen, either randomly or based on how often it is observed in the training database (eq 6). All subsequent fragments are added in the following manner. Throughout growth fragments present in the growing compound are assigned to one of three lists based on what type of growth is available from that fragment: linear (only one existing connection to another fragment), branch (at least two connections to other fragments), or none (all growth sites have been filled). The population of these three lists determines the possible growth modes that are available (linear, branch, or none). If both the linear and branching growth modes are available, one of the modes is selected based on a user defined branching probability. If only one of the modes is detected, that mode is automatically selected. If all growth sites are saturated, then growth is terminated and the molecule is discarded. Unless stated otherwise, the branching probability $P(B)$ was set to 0.5 for our experiments. This was to access moderately branched structures, while avoiding highly branched structures that might be synthetically inaccessible.[69] Once a growth mode has been selected, a growth fragment is then chosen from the appropriate list, and a growth site on that fragment is randomly selected.

The selection of the fragment that will be connected to the current growth fragment is then made. This can either be done by using the transition probability of the growth fragment to select the subsequent fragment or by deciding to select a ring or nonring fragment prior to using transition probabilities to select the next fragment. When the latter method is used, a ring nonring decision is made based on how often the growth point of the fragment is connected to a ring or nonring in the training database. This value can also be set by the user to be a given value for all fragments. Once a decision to grow to a ring or nonring is selected, the correct type of fragment is then selected based on the growth fragment's transition probabilities. It should be noted that the transition probability matrix in this case is split into two matrices, one for transitions to rings and one for transitions to nonrings, and normalized accordingly.

This process then repeats itself until all growth sites are saturated, a user defined maximum number of fragments have been added, or a maximum molar mass has been obtained. The molecule is written to file as a SMILES string.[70] This process is illustrated in Figure S3.

1.3. 3mer Screen. Since our growth algorithm only employs information about the current growth fragment when adding a new fragment and is "unaware" of any other fragments that might be already connected to it, it is capable of stringing together a sequence of 3 fragments that might be synthetically unfeasible or chemically unstable. For example, it might produce a geminal diol (Figure S2, top) which in most cases would convert to a ketone in aqueous conditions. To remedy this situation, one could employ a second order Markov Chain, where transition probabilities are based on the current growth fragment and all fragments already connected to it. We decided to use a simpler approach that entails screening a grown library of compounds for 3mers that are "disallowed" and removing all compounds that contain them. FOG implements two sources for disallowed 3mers. First we search our training database for all 3mer sequences that can be composed of our fragments. In order to avoid being too stringent, rings are treated very generally. For example, a SMARTS string representing a ring carbon would match all sp^3 carbons that are in a ring, but it would not be sensitive to the type of ring that the sp^3 carbon belongs to. Any 3mer sequence that is not observed in the training database is considered disallowed. In addition, user defined disallowed 3mers are added to avoid chemically unstable moieties such as acetals, ketals, aminals, and iminals (Figure S2). The user defined disallowed 3mers are similar to the "disallowed angles" in the chemical rules employed by

GroupBuild.[27] One could imagine using higher order screens (4mer, 5mer, etc.), but we chose not to do this as we felt it would hinder the algorithm's ability to generate novel compounds while not significantly increasing the likelihood of generating synthetically feasible compounds.

2. Classification Algorithm: TopClass. In order to evaluate the output of our growth algorithm, we developed the classification algorithm TopClass. TopClass scores molecules based on a number of individual components that score different features of a molecule. Each measure is based on the difference in probabilities or log odds score of observing some feature in a given database *A* versus *B*. They return a positive or negative value depending on whether the scored molecule is deemed more representative of one class of molecules or another. The total score is a linear summation of the individual scores, and a molecule is classified based on the sign of the final score. The probability of finding a fragment in a database (Methods 2.1) as well as the log odds score of fragment connections (Methods 2.2) was computed in a similar fashion to Hutter,[64] except we ascertained the probability of fragments and fragment connections, rather than atoms and atom connections. We also did not look at fragment distributions that were separated by more than one bond, as Hutter did for atoms.[64] As such, these scores are only briefly discussed, and the coupled 1D topological metrics (Methods 2.3) are described in more detail.

2.1. Fragment Probability. The probability of finding fragment i in a given database of compounds was defined as the number of occurrences of that fragment N_i^D divided by the sum of the counts of all fragments

$$p_i = \left(\frac{N_i^D}{\sum_s (N_s^D)} \right) \tag{6}$$

The difference in probabilities of observing fragment i in database A compared to database B is thus

$$D_i = p_i^A - p_i^B \tag{7}$$

The full fragment frequency score is then

$$L_1' = \frac{1}{M_1} \sum_i^n \left\{ \begin{matrix} D_i \text{ if } i \text{ in molecule} \\ 0 \text{ otherwise} \end{matrix} \right\} \tag{8}$$

where M_1 is the total number of fragments in the molecule. We initially used the same fragments that were used in growth to generate this score. It was quickly observed however, that by adding fragments that were more representative of given databases (DOS, NP, etc.), more accurate classification of compounds could be obtained.

2.2. Fragment Connections. The probability that fragment i is connected to fragment j in a database was computed as the total times i was found connected to j (N_{ij}^D) divided by the sum of the counts of all pairing combinations of the fragments

$$q_{ij} = \left(\frac{N_{ij}^D}{\sum_s \sum_k (N_{sk}^D)} \right) \tag{9}$$

We can also define the probability p_i' that a fragment i is found connected to any other fragment in our pool we have as all connections i makes to other fragments $\sum_s N_{is}^D$ divided by all possible pairing combinations of the fragments

$$p_i' = \left(\frac{\sum_s N_{is}^D}{\sum_s \sum_k (N_{sk}^D)} \right) \tag{10}$$

Note, this is not the probability of finding i but rather the frequency that we observe it bonded to other fragments belonging to our fragments of interest. We then computed the relative probability that i is connected to j by dividing the frequency of finding that pairing q_{ij} by the product of the sums of the individual frequencies of finding i or j connected to other fragments

$$S_{ij}' = \left(\frac{q_{ij}}{p_i' p_j'} \right) \tag{11}$$

Rather than using the relative probability directly, it is more convenient to obtain the log odds score by taking the logarithm of the relative probability

$$S_{ij} = \ln\left(\frac{q_{ij}}{p_i' p_j'} \right) \tag{12}$$

Since it is possible that a certain fragment pairing is not observed, or that a specific fragment is never observed, the relative probability is assigned a minimum value of 0.0001. The log odds matrix is then renormalized as described by Hutter.[64] The difference in log odds scores between database A and database B for a specific fragment pairing is then

$$D_{ij} = S_{ij}^A - S_{ij}^B \tag{13}$$

The full fragment connection score is then

$$L_2' = \frac{1}{M_2} \sum_i^n \sum_j^n \left\{ \begin{matrix} D_{ij} \text{ if } ij \text{ in molecule} \\ 0 \text{ otherwise} \end{matrix} \right\} \tag{14}$$

where M_2 is the total number of fragment pairs in molecule.

2.3. Coupled 1D Topology Metrics. For each 1D topology metric, we computed the joint probability of two variables $P(a,b)$ that we reasoned would yield more information when looked at jointly, rather than as two single probabilities $P(a)$ and $P(b)$. For example, one of the joint probabilities we computed was H-bond acceptors and H-bond donors per molecule $P(don,acc)$. Hydrogen bond interactions allow drugs to specifically interact with a macromolecule, and it would be highly unlikely that a druglike molecule would lack both H-bond donor and H-bond acceptor sites. This is not a requirement for nondrugs, however, so we suspected molecules lacking both H-bond donors and H-bond acceptors would be scored as nondruglike.

A second topology metric we employed was the joint probability that a molecule has a certain number of atoms as well as a certain number of rings $P(atoms,rings)$. We reasoned that conformationally rigid rings would be favored in drugs and that a molecule with a high atom count but low ring count would probably be scored as a nondrug. In a similar vein, we computed the joint probability that a molecule has a certain number of atoms as well as a certain number of rotatable bonds *P*(*atoms*,*rbonds*). Each of the joint probabilites is computed as the total observances of both variables $N^D_{a,b}$ divided by the total number of molecules in the training database N^D_{total}

$$P(a,b) = \left(\frac{N^D_{a,b}}{N^D_{total}}\right) \qquad (15)$$

The difference of the joint probabilities between two database A and B was then computed for each possible joint value

$$D_{P(a,b)} = P(a,b)^A - P(a,b)^B \qquad (16)$$

Each of the metrics (donor, acceptor, rings, rotatable bonds, and atoms) were computed for each molecule using cxcalc available from ChemAxon.[68] The atom counts were binned with increments of 5. Thus we have

$$D_{P(don,acc)} = P(don,acc)^A - P(don,acc)^B \qquad (17)$$

$$D_{P(atoms,rbonds)} = P(atoms,rbonds)^A - P(atoms,rbonds)^B \qquad (18)$$

$$D_{P(atoms,rings)} = P(atoms,rings)^A - P(atoms,rings)^B \qquad (19)$$

It is possible to have very rare occurrences of some pairs of variables, but we do not correct for these in any way, because by the virtue of being rare they will not significantly contribute to the classification of a molecule.

2.4. Tuning the Separation Algorithm. A linear summation of each individual score yields the total score. Thus, if we consider the differences in fragment frequencies and fragment connections (2D metrics) we have

$$L = \alpha_1 L'_1 + \alpha_2 L'_2 \qquad (20)$$

This is what we used to separate DOS compounds from natural products (Table 1, discussed in results). Taking into account the coupled 1D topology metrics as well we have

$$L = \alpha_1 L'_1 + \alpha_2 L'_2 + \alpha_3 D_{P(don,acc)} + \alpha_4 D_{P(atoms,rbonds)} + \alpha_5 D_{P(atoms,rings)} \qquad (21)$$

The coefficients (α_1-α_5) are chosen in order to yield the best separation, without overfitting to the training set. This is by optimizing their values on a validation set, assembled by randomly selecting ~10% of the training set, and then applying these optimized values to the test set. These values were varied in increments of 5 with the constraint that $\sum_{i=1}^{5}\alpha_i = 100$. The weights for various separations are provided in the Supporting Information.

3. Separation Algorithm: D(min) or D(ave). Chemical fingerprints for compounds were generated using GenerateMD (Chemaxon).[68] The minimum *D*(min) and average *D*(ave) Tanimoto dissimilarity as computed by Chemaxon's Compr[68] when the chemical fingerprints of test compounds were compared against those of training database compounds was used to assign test compounds to one database or another. If $D(\mathrm{min})^A < D(\mathrm{min})^B$ the test compound was assigned to database A, while if $D(\mathrm{min})^A > D(\mathrm{min})^B$ the compound was assigned to database B. The average dissimilarity *D*(ave) between a test compound and training database compounds was also used in a similar manner to assign test molecules to database A or B. Details of chemical fingerprint generation, the *D*(min) and *D*(ave) calculations, as well as how the *D*(min) score was combined with the three coupled 1D metrics described in eqs 17−19 are provided in the Supporting Information.

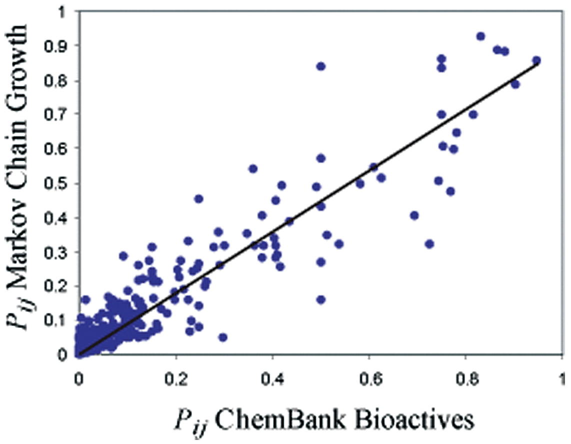


Figure 4. The probability that fragment *i* is connected to fragment *j* (P_{ij}, eq 4) in a database of molecules grown with a Markov Chain versus P_{ij} of ChemBank Bioactives used in training the Markov Chain. The probability of branching *P*(*B*)=0.5.

4. Authentic Compound Libraries. The drug test (218) and training sets (2495) as well as the nondrug test (110) and training sets (1263) have previously been described.[64] The DOS and Natural Product (NP) libraries are from the Forma Collection compiled at the Broad Institute.[65] About 10% of the DOS and NP compounds were randomly selected for use as the test sets (673 for DOS, 230 for NP), and the remaining compounds were used as the training sets (5950 for DOS, 2247 for NP).

RESULTS

In an attempt to develop an algorithm that generates novel small molecules that are similar but not identical to known compounds, we used a Markov Chain approach with branching, treating each growth fragment as the current state, and selecting subsequent fragments based on transition probabilities. These probabilities for a diverse set of fragments (Table S1) were initially trained on the ChemBank Bioactives (4669 compounds).[65] We chose this database for our initial studies because it is relatively small and contains chemically reasonable molecules capable of perturbing biological systems. Using such an approach we grew 10,000 molecules and compared their connectivity statistics (the probability that a given fragment *i* is connected to fragment *j* in the database, eq 4) to those of the ChemBank Bioactives Database. The excellent agreement we observed (R^2=0.90 for all points, R^2=0.76 when values <0.1 were removed, Figure 4) suggested that we were growing molecules that were to some degree similar to the database that we had trained our algorithm on. This correlation was not observed for unbiased growth (R^2=0.01). The same agreement was observed when a larger training database (NCI Open Database Aug00,[66] 250,251 compounds, R^2=0.92 for all points, R^2=0.81 when values <0.1 were removed) was employed.

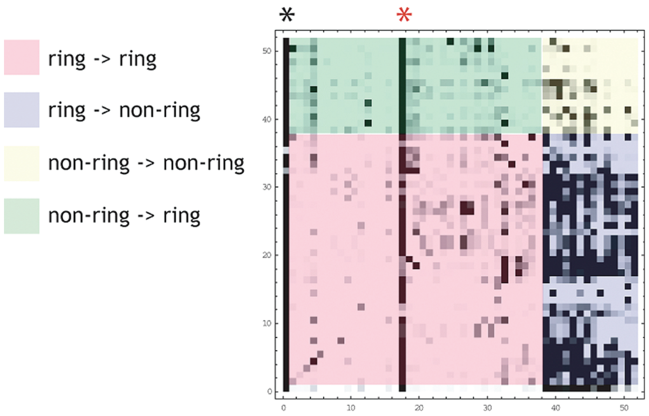


Figure 5. Transition probability matrix used in Markov Chain growth. High probability transitions are depicted as black, while low probability transitions are clear. Certain transitions are highlighted with color. Transitions from methyl to other fragments (bottom row) and from other fragments to methyl (first column) are not highlighted. Asterisks are used to denote the columns representing transitions to methyl (black) and benzene (red).

The transition probability matrix (Figure 5) that we obtained after training on the ChemBank Bioactives was in good agreement with chemical intuition. Transitions from ring fragments to other ring fragments were low, as might be expected since these connections are often difficult to synthesize. The matrix is also quite sparse, revealing that many fragments are never connected in the training database. This undoubtedly helps focus combinatorial growth. It is also apparent that transitions to specific fragments are especially high—most notably the methyl and benzene fragments. Since sp^3 carbons often serve as part of the framework of organic compounds, it is not surprising that the methyl group is so prominent. Benzene chemistry is very mature, and facile substitutions and transformations of appendages allow for diverse groups being connected to benzene.[71]

To further investigate the behavior of our algorithm, we studied how the number of fragments added and the branching probability influence how grown molecules compare to the training database. We used the corresponding SMARTS[67] of molecules of various sizes (1−11 fragments) and grown with different branching probabilities ($P(B)$ = 0.0−1.0) as substructure search strings on the original training database (Figure 6). The branching probability did not have a significant effect on the percentage of substructure hits (Figure S4). As the number of fragments increased, on the other hand, the number of hits fell quite rapidly (Figure 6, FOG (MC)). It was reassuring that the number of substructure hits was much higher in the molecules that were grown with a statistical bias compared to the unbiased control (Figure 6, No Bias). We interpret this data in the following way: when a few fragments are added with our algorithm, it is likely that they yield a substructure of a molecule in the original database. As more fragments are added, an entirely new molecule is accessed, but it is likely that it is composed of one or more substructures that can be found in the database.

Statistically biasing the addition of fragments with a Markov Chain approach was not enough to produce synthetically feasible compounds. Indeed, when we surveyed organic chemists (Supporting Information) and asked them to assess the synthetic feasibility of compounds that were grown with and without a statistical bias, molecules generated with a statistical bias were just as likely to be scored as unsynthesizable or unstable as compounds grown with no bias (Figure S5). Upon visually inspecting what molecules were deemed unstable, the following improvements were implemented in FOG.

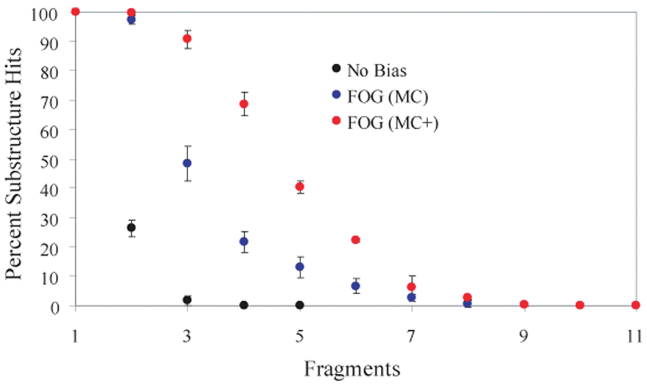


Figure 6. The probability of a grown molecule to be a substructure hit of a compound in the training database. Molecules are either grown with no statistical bias (No Bias) or with FOG using a Markov Chain (MC) or a Markov Chain employing a ring/nonring transition probability as well as a disallowed 3mer screen (MC+). Errors bars reflect the standard deviation of 3 sets of 100 grown molecules.

First, we noticed that the probability that a fragment is connected to a ring in the grown molecules (11%) was less than in the training set (24%). We realized that this is due to our fragment pool under-representing ring fragments in the training database. This would lead to fragments transitioning to nonring fragments more often than they would if more ring fragments were included in our growth fragments. As such, we added a ring/nonring transition probability. Whenever a fragment is about to be added to a growing molecule, the algorithm first decides whether the next fragment should be a ring or a nonring, based on how often the growth fragment is connected to rings in the training database. It then selects a specific fragment from the pool of rings or nonrings based on renormalized transition probabilities. The second modification we made was to include a disallowed 3mer screen after growth. Since our algorithm adds fragments based on the current growth fragment and not on fragments that might be connected to the current growth fragment, it is capable of forming 3mers that are chemically unstable or synthetically demanding. To remedy this FOG employs a 3mer screen that removes all 3mers that are undesired by the user (such as acetals), in addition to any 3mer substructures that were not detected in the training database.

After generating molecules with our modified algorithm, we observed that ring propensities in the grown molecules (21%) were similar to those observed in the training set (24%). Furthermore, the likelihood of growing substructures present in our training database was much higher than the previous version (Figure 6, FOG (MC+)). Surveys of organic chemists demonstrated that FOG was significantly improved: it did not grow a single molecule that was deemed unsynthesizable or unstable (Figure S5). The difficulty of synthesis, however, remained similar to molecules grown without any bias (Figure S6). Synthetic evaluation of the grown compounds with SYLVIA[41] suggests that their synthetic acces-

Table 1. Evaluation of Separation Algorithms[a]

		classification method (% correct)				
compound set	comp.	2D	2D + c1D	*D* (min)	*D* (ave)	*D*(min) + c1D
DOS test	673	79.3		99.7	96.3	
NP test	230	90.0		97.8	56.1	
DOS grown	100	100.0		63.0	100.0	
NP grown	100	88.0		78.0	17.0	
drugs test	218	80.3	80.7	94.5	98.6	92.7
nondrugs test	110	58.2	62.7	68.2	9.1	73.6

[a] We used fragment and fragment connection biases (2D) as well as coupled 1D metrics such as H-bond donor/H-bond acceptors in addition to the 2D descriptors(2D + c1D). In addition *D*(min) and *D*(ave) of chemical fingerprints compared to training sets were used for classification. We also used a combination of *D*(min) and the coupled 1D metrics (*D*(min)+c1D). Test sets were evaluated (DOS vs NP, drug vs nondrug) as well as molecules grown with the FOG algorithm (DOS grown, NP grown).

sibility is similar to those of the chemicals they were trained on and that they are slightly more accessible than compounds grown with no bias (Figure S7).

We then sought to grow classes of molecules with FOG. In order to prove our algorithm capable of such a task, we first needed to develop a classification algorithm capable of accurately classifying molecules. We initially used an algorithm that classified compounds based on statistical biases in the fragments that they were composed of and how they were connected (eq 20). Using such an algorithm, we could accurately separate authentic DOS products from natural products (2D, Table 1). We then trained our growth algorithm on either DOS or NP compounds and generated libraries of putatively DOS and NP-like molecules, respectively. Our classification algorithm then demonstrated that the grown DOS compounds (100%) and natural product compounds (88%) were indeed classified as the molecules they were trained on. Using an alternative separation algorithm based on the minimum chemical fingerprint Tanimoto dissimilarity when a test compound is compared to training set compounds, we were also able to separate authentic DOS compounds for NP compounds with high accuracy (*D*(min), Table 1). We also observed that using the average dissimilarity in chemical fingerprints as a metric to classify compounds gave poor separation of classes (D(ave), Table 1). Using *D*(min) to assess our grown molecules suggested that while our molecules were more often scored as the database they were trained on (63.0% DOS, 78.0% NP), the enrichment was more moderate than our earlier assessment would suggest (2D, Table 1).

We then attempted to separate authentic drugs from nondrugs. Our initial results for identifying drugs (80.3%) and nondrugs (58.2%) were superior to accuracies reported in literature for the same databases (71.1% for drugs, 40.9% for nondrugs).[64] Our method allowed us to inspect the over-representation of certain fragments in drugs versus nondrugs (Figure 7). We see for example that the top three fragments over-represented in drugs are methyl, amide, and nonring trisubstituted sp^3 carbon, while the alkene, nonring sp^2 oxygens and fused benzene rings (as in naphthalene) are over-represented in nondrugs.

Our next aim was to improve separation accuracy by adding three 1D coupled topology metrics to our classification algorithm. These metrics score the differences in joint probabilities for two variables between two databases. They are depicted as 2D matrices in Figure 8, and investigation of the plots yields valuable information concerning druglike features. Interestingly the druglike region (red) of the donor−acceptor plot resides above the nondruglike (blue) region. From this plot we see that molecules with ∼3−7 more acceptors than donors are scored druglike. We also observe that the bulk of both the drug and nondruglike regions lie within the Lipinski cutoffs (<10 acceptors, <5 donors), while some of the druglike region lies outside of these cutoffs. From the atoms-rings plot it is clear that highly fused structures (high ring:atom ratio) as well as large molecules without any rings (low ring:atom ratio) lie in the nondruglike region. Drug and nondruglike regions are also separated for rotatable bonds versus atoms. Molecules with ∼30−40 atoms that are either highly flexible or extremely rigid tend to be nondrugs, but those with intermediate flexibility tend to be scored as drugs at that size. Molecules with >40 atoms are predominantly scored as drugs, while those with <25 are predominantly scored as nondrugs. Using the modified algorithm, TopClass (Topology Classifier), the accuracy in classifying drugs remained the same (80.7% compared to Hutter's accuracy of 71.1%),[64] while there was an improvement in the classification of nondrugs (62.7%), which is a significant improvement over the accuracy reported by Hutter (40.9%).[64]

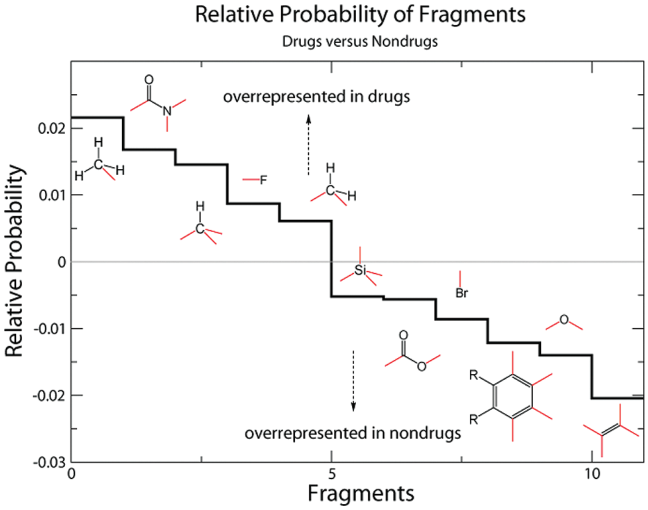


Figure 7. Relative probability of fragments in drugs versus nondrugs. Red bonds indicate connections to any atom including hydrogen, except for the sp^3 carbons where the number of attached hydrogens in explicitly defined. The benzene ring with two R substituents searches for fused benzene rings as in napthalene. Only fragments with large positive or negative values are depicted for clarity.

We then assessed why the TopClass algorithm fails to score some compounds correctly. For all the compounds that were scored incorrectly, we calculated what percent failed primarily due to one of the components that are summed to obtain the final score (Table 2). If a component had the incorrect sign (ex., "-" for a drug) and was larger in magnitude than all other components with an incorrect sign, it was deemed primarily responsible for the incorrect score. In no cases was the H-bond donor:acceptor portion of the score primarily responsible for an incorrect score using this method. The fragment pair score was most often responsible (47.6% incorrect drugs, 34.1% incorrect nondrugs). This may

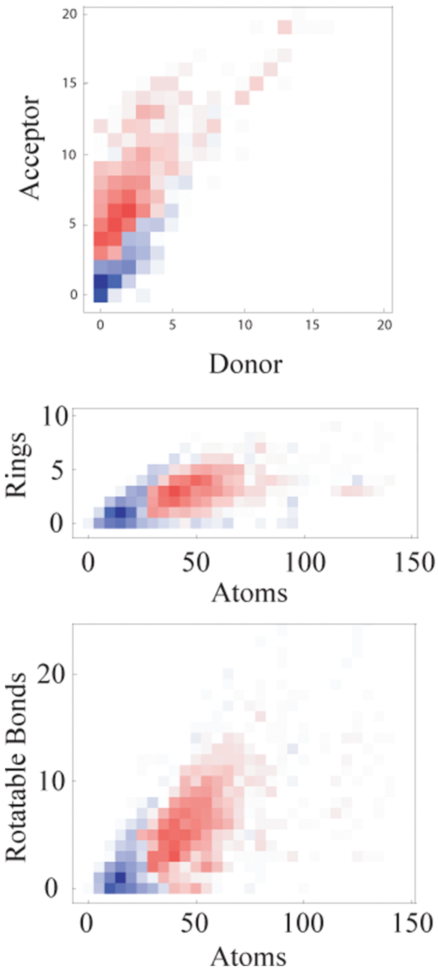


Figure 8. The differences in joint probabilities of 1D topology descriptors between drugs and nondrugs reveal druglike regions (red) and nondruglike regions (blue). Atom counts are binned with increments of 5.

be because this score often has outlying values compared to the other scores, and in the future this might be minimized by assigning a cutoff value to fragment pair scores with outlying magnitudes.

We also selected falsely scored compounds representative of specific component failures (Figure 9) and report their entire score (Table 2). In the case of oxapadol and ampyrone, our fragment pool does not cover the entire framework of the compound. For oxapadol, the fragment component only matches the benzene appendage and the fused benzene portion of the polycyclic scaffold, and for ampyrone, this score is only aware of the substitutions but not the central ring. In the future these types of failures may be diminished by including more fragments in our substructure searches. For flusilazole, the nondruglike connections between silica and its neighbors is primarily responsible for the nondruglike score. The rest of the representative incorrectly scored drugs (mesalamine, amphetamine) are rather small, and thus the scores based on ratios of rings:atoms and rotatable bonds: atoms score them incorrectly. With regards to molecules that deviate significantly in some metric (in this case size) from the majority of compounds in a database, it may be beneficial in the future to train TopClass on subclasses of a database after database clustering, since TopClass currently compares test compounds to a database as a whole, rather than making molecule−molecule comparisons as with *D*(min) (results below). With Sudan-III and irigenin, although the overall fragment scores are nondruglike, the molecules lie in the druglike region of the rings:atoms and rotatable bonds:atoms scores, which dominate the final score.

We also applied a separation strategy based on the minimum Tanimoto dissimilarities *D*(min) of chemical fingerprints of a test compound compared to the training set compounds. We obtained high accuracies separating the drugs (94.5%) and nondrugs (68.2%) test sets. The overall accuracy was slightly improved by combining the *D*(min) score with our three coupled 1D metrics (92.7% drugs, 73.6% nondrugs).

In order to ascertain how enriched in druglikeness our grown molecules were compared to those grown with no bias, we devised the following two step screen. First we classified molecules as “random” or drugs. Compounds previously grown with no bias (10,000) were used at the training set to represent “random” compounds. Compounds that passed the first screen were then classified as drugs or nondrugs. We performed these screens using either TopClass, *D*(min), or *D*(min) as well as the coupled 1D metrics from TopClass (*D*(min)+c1D) as classification algorithms (Table 3). When we subjected 200 compounds grown with no bias to the first screen using TopClass, not a single molecule was classified as druglike (Table 3). When we classified molecules grown with FOG (previously trained on the drug database), however, 83.0% remained after the first screen, and 81.5% of the initial 200 remained after both screens. Using other separation algorithms yielded slightly different results. Using *D*(min) or *D*(min)+c1D, 2% of the compounds grown with no bias are identified as drugs, while 39.5% or 46.5%, respectively, of compounds grown with FOG are identified as drugs.

We also used two popular oral bioavailability screens to assess the druglikeness of our grown molecules. To assess the accuracy of the Lipinski[45] and Veber[72] screens (details in the Supporting Information), we first applied them to the authentic drugs and nondrugs (Table 3). The majority of drugs pass these screens but unfortunately so do the majority of nondrugs. It has been demonstrated that the Lipinski rule of 5 is a poor discriminator of drugs versus nondrugs,[47,49] and this further supports this notion. Even so, to demonstrate the weaknesses inherent in relying on these screens during *de novo* design, we applied them to our grown molecules. A good amount of compounds grown with FOG pass the Lipinski[45] (55%) and Veber[72] (80%) screen. Surprisingly the majority of the compounds grown with no bias pass the Lipinski screen (79.5%) or the Veber screen (80.0%) even though only 0−2% passed our two step screens.

While focusing the combinatorial explosion inherent in *de novo* approaches on druglike chemical space, we also wanted to make sure that we were balancing the ability to grow “druglike” molecules with the ability to access synthetically accessible *new* molecules. According to SYLVIA,[41] the synthetic accessibility of our grown drugs was similar to that of authentic drugs of similar molecular weight, and it was slightly more accessible than compounds grown with no bias (Figure 10A). To ensure that we were accessing new molecules, we calculated the minimum Tanimoto dissimilarity (see the Supporting Information) when comparing each of our grown drugs’ chemical fingerprints with the entire training database of authentic drugs

Table 2. Drugs and Nondrugs That Were Scored Incorrectly by TopClass

				components of TopClass score for example				
	% inc.[a]	example[b]	score	F	FP	DA	ARB	ARI
F	7.1	oxapadol	−0.16	**−0.30**	0	0.07	0.11	−0.05
FP	47.6	flusilazole	−31.79	0.29	**−33.08**	0.14	0.38	0.48
ARB	26.2	mesalamine	−5.40	−0.24	−0.95	−0.04	**−2.40**	−1.77
ARI	19.0	amphetamine	−1.22	0.45	0.13	−0.27	−0.62	**−0.91**
F	19.5	ampyrone	0.04	**0.58**	0	−0.14	−0.15	−0.25
FP	34.1	chavicine	26.23	−0.39	**25.15**	0.07	0.65	0.75
ARB	17.1	Sudan-III	0.87	−0.35	0.25	0.15	**0.50**	0.32
ARI	29.3	irigenin	0.84	−0.14	−0.30	0.02	0.50	**0.75**

[a] The percentage of compounds scored incorrectly (% inc.) primarily due the frequency score (F), fragment pair score (FP), atoms:rotatable bonds score (ARB), or atoms:rings score(ARI) are reported. No compounds were scored incorrectly primarily due to the donor:acceptor score (DA). [b] A representative example of an incorrectly scored compound due primarily to a component (F, FP, ARB, or ARI) is reported as well as the individual weighted scores that were summed to yield the overall score used in classification. The score of the component most responsible for the overall false score is bold.

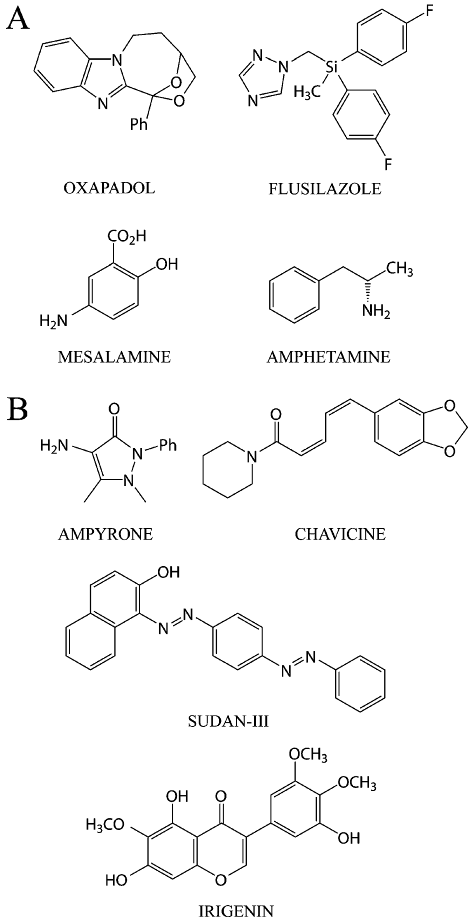


Figure 9. Representative drugs incorrectly scored as nondrugs (A) and nondrugs incorrectly scored as drugs (B). Explicit scores for these compounds are in Table 2.

(Figure 10B). We also calculated the minimum dissimilarity of the drugs test, nondrugs test, and no bias grown libraries for comparison. A minimum dissimilarity of ∼0.4−0.6 was obtained for most of our grown compounds, ensuring that we were indeed generating novel compounds.

DISCUSSION

We have developed an algorithm, FOG, which generates new compounds in a chemical space that is similar to the compounds that it was trained on, whether they are drugs, natural products, or DOS compounds. At the heart of our algorithm is the sequential growth of small molecules constrained by the transition probabilities of the growth fragment. This is in contrast to programs that sequentially grow small molecules by selecting new fragments randomly or by selecting them based on a user defined frequency[26,73] or their frequency in a database,[74] rather than based on the frequency of their *connections* to the growth fragment in a database. Importantly, libraries grown with our transition probabilities reproduce the frequencies in connections between fragments (Figure 4). Our algorithm can easily be incorporated into *de novo* design programs that employ sequential growth of fragments, or it can be used as a standalone program to generate a virtual library of compounds of specific classes.

Other methods have been employed to generate 2D virtual libraries that are then converted into 3D molecules. This has been done by randomly combining SMILES strings (BOOMSLANG[75]) or substituting side chains onto a given framework.[76] DBMAKER grows side chains and assigns atom types to a user defined framework, employing user specified constraints during the selection of atoms attempting to focus the potential combinatorial explosion on meaningful molecules.[73] While this is more sophisticated than randomly splicing together SMILES strings or combinatorially substituting side chains, it requires extensive user input for each application. For example, when potential ligands were grown with DBMAKER, a different user defined scaffold was used for each target, and different parameters files dictated the growth of various parts of each ligand.[73] We sought an alternate strategy that did not require the guidance of extensive user defined constraints. MOLMAKER is strongly rooted in graph theory and uses vertex degree sets to generate all possible molecular graphs for a given set of constraints such as number of atoms and maximum ring size.[77] It then adds atom types to the graphs in a probabilistic manner and finally screens the compounds for user defined disallowed substructures. Another strategy that has been used to generate 2D libraries that are subsequently converted to 3D is using synthetic rules to guide the assembly of building blocks.[43,78,79] Some methods also employ synthetic rules when generating molecules in the active site of a protein.[2,42] We sought a different method that does not require synthetic knowledge for two primary reasons. First, the enumeration of synthetic rules would be extremely time-consuming and difficult and could not be readily implemented by developers of various

Table 3. Percentage of Compounds Scored As Drugs Using the Two Step Screens As Well As Popular Oral Bioavailability Screens Such as Lipinski (L) and Veber (V)

		Drug Screen (%)[a]						
compound set	compounds	2 step TopClass	2 step *D*(min)	2 step *D*(min) + c1D	L(2)	L(1)	L(0)	V
drugs test	218				100.0	93.1	85.3	84.8
nondrugs test	110				99.1	99.1	88.2	93.6
drugs grown	200	81.5	39.5	46.5	100.0	99.5	54.5	79.5
no bias grown	200	0	2.0	2.0	100.0	99.0	79.5	80.0

[a] Two step screens were based on TopClass, *D*(min), or *D*(min) and coupled 1D descriptors (*D*(min)+c1D). Oral bioavailability screens such as Lipinski (L) with 2, 1, or 0 violations are allowed and Veber (V) are also reported.

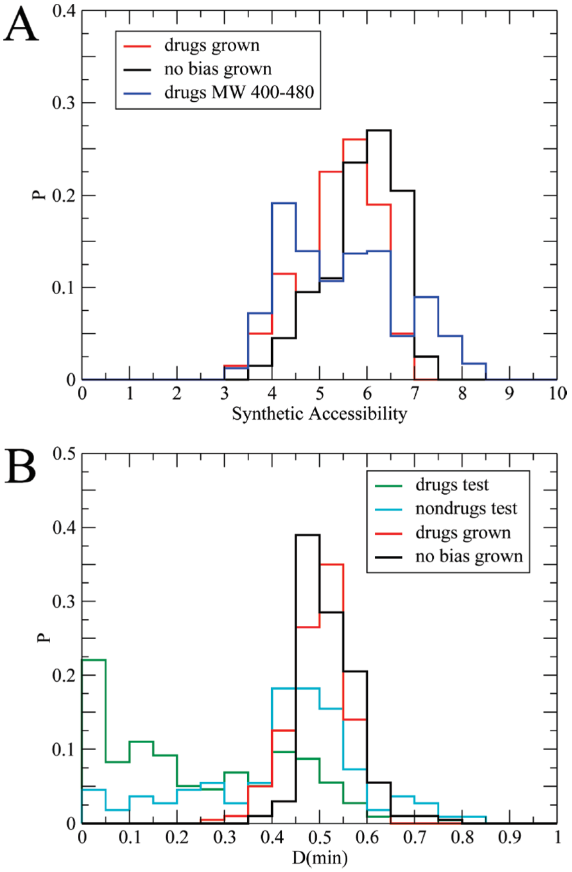


Figure 10. (A) Histogram of synthetic accessibility (1=easy, 10=difficult) of drugs grown ($N = 200$), no bias grown ($N = 200$), and authentic drugs with MW of 400−480 ($N = 410$) as assessed by SYLVIA. (B) Histogram of minimum dissimilarity *D*(min) of chemical fingerprints of the drugs test ($N = 218$), nondrugs test ($N = 110$), drugs grown ($N = 200$), and no bias grown ($N = 200$) libraries compared to the authentic drugs training set. Chemical fingerprints were generated with GenerateMD (Chemaxon), and *D*(min) was calculated with Compr (Chemaxon).

de novo tools. Second, when a molecule is growing in the binding site of a macromolecule, if synthetic transformations are taking place, then intermediates may lack steric or electrostatic complementarity to the binding site, and a potential ligand might not be found due to these unfavorable intermediates. This would not be a problem if the ultimate goal is to generate a virtual library that will subsequently be docked or screened.

We have also developed a linear scoring algorithm, TopClass, which classifies compounds, such as drugs and nondrugs. Our algorithm is transparent and has allowed us to investigate interesting features that distinguish drugs from nondrugs. For example, the H-bond donor and acceptor plot (Figure 8) revealed that drugs tend to have ∼3−7 more acceptors than donors. This observation leads one to question whether binding sites of proteins have a higher ratio of donors than acceptors, or whether it is because of some other physical or biological reason. Nondrugs on the other hand tended to have the same number of donors and acceptors, so it does not seem to be a synthetic bias. We also used a complementary separation method based on Tanimoto chemical fingerprint dissimilarities *D*(min). This approach was extremely accurate in classifying test sets of drugs and nondrugs and helped inform how druglike our grown molecules were. It is important to note the differences between these two strategies. TopClass compares a molecule to a database as a whole, while *D*(min) compares a molecule to single molecules in a database. It may be due to this difference that the accuracies obtained with the *D*(min) method were superior to TopClass. For example, if a small subclass of compounds “A-1” resides within a larger database “A” of chemicals, and this subclass is more topologically similar to compounds in database “B” than other compounds in database “A,” TopClass will likely assign test compounds that resemble “A-1” compounds to database “B”, as the characteristics of subclass “A-1” may be washed out by the other molecules in database “A”. *D*(min) would not suffer from this as it would directly compare test compounds to the members of the “A-1” subclass, and this score would in no way depend on how small this subclass is in respect to the rest of library “A.” In the future it may be advantageous to train TopClass on clusters of similar compounds and compare test compounds to these profiles rather than on an entire database of compounds.

Using our TopClass separation algorithm, we were able to show that our generated molecules did indeed occupy the chemical space that was intended (81.5% deemed druglike). Alternate separation strategies (*D*(min) or *D*(min)+c1D) also suggested that we were biasing the growth of druglike chemicals. Generating druglike libraries is not a trivial task. For example, when molecules were generated by randomly combining fragments, <0.1% were selected when they were screened for similarity to known drugs and predicted biological activity (using trend vector analysis).[74] When 30 compounds were generated by randomly combining common scaffolds and appendages found in drugs, only 33% were scored as druglike.[59] Likewise, when 10^6 compounds were generated by a similar method, only 7% were considered CNS-active with a high degree of confidence.[60] When 26.4 × 10^6 million compounds containing 11 atoms or less and composed of chemically stable combinations of C, N, O, or F were virtually generated and screened with a Bayesian

ANN, only ~0.16% were deemed as having GPCR, kinase, or ion channel blocking activity.[9] We found that when we build molecules by the sequential addition of fragments without any bias it is nearly impossible to find a molecule that is scored as a drug (0% with TopClass, 2% with *D*(min) or *D*(min)+c1D), although the majority of these compounds pass popular oral bioavailability filters such as Lipinski (80%) or Verber (80%). When we employ our growth algorithm, a much greater amount of generated compounds are scored as drugs with various separation algorithms in our two step method (81.5% Topclass, 39.5% *D*(min), or 46.5% *D*(min)+c1D). This signifies a huge enrichment in druglike character when our algorithm is employed. It also strongly supports the notion that methods that generate compounds without any connectivity bias, followed by an oral bioavailability filter (such as LigBuilder[46]), generate molecules that lie outside of druglike space, although a user of these methods may have a false sense of focusing the combinatorial space with the oral bioavailability filters.

Our findings strongly suggest that implementing our growth algorithm in *de novo* methods could help focus potential combinatorial explosion on compounds that occupy relevant chemical space, thus greatly improving the chances of identifying interesting lead compounds. We are currently implementing it in SMoG[19] as well as using it to generate interesting virtual libraries.

ACKNOWLEDGMENT

We thank Michael C. Hutter for providing us with the drug and nondrug training and test sets that he assembled, Paul Clemons of the Broad Institute for providing us with the Natural Product and Diversity Oriented Synthesis libraries, Winston Wang for technical programming assistance, and each of the expert chemists that completed our surveys (see the Supporting Information).

Supporting Information Available: Details of correction count *C*, Figures S1−S7, Tables S1-S3, details of surveys, a full list of survey takers, and details of Tanimoto dissimilarity calculations. This material is available free of charge via the Internet at http://pubs.acs.org.

REFERENCES AND NOTES

(1) Mauser, H.; Guba, W. Recent developments in de novo design and scaffold hopping. *Curr. Opin. Drug Discovery Dev.* **2008**, *11* (3), 365–74.

(2) Vinkers, H. M.; de Jonge, M. R.; Daeyaert, F. F.; Heeres, J.; Koymans, L. M.; van Lenthe, J. H.; Lewi, P. J.; Timmerman, H.; Van Aken, K.; Janssen, P. A. SYNOPSIS: SYNthesize and OPtimize System in Silico. *J. Med. Chem.* **2003**, *46* (13), 2765–73.

(3) Schneider, G.; Fechner, U. Computer-based de novo design of druglike molecules. *Nat. Rev. Drug Discovery* **2005**, *4* (8), 649–63.

(4) Todorov, N. P.; Alberts, I. L.; Dean, P. M. De Novo Design. In *Comprehensive Medicinal Chemistry, II*; Triggle, D. J., Taylor, J. B., Eds.; Elsevier Science: Amsterdam, The Netherlands, 2006; Vol. 4, pp 283−305.

(5) Cayley, A. On the Mathematical Theory of Isomers. *Philos. Mag.* **1874**, *47*, 444–446.

(6) Trinajstic, N.; Nikolic, S.; Knop, J. V.; Muller, W. R.; Szymansky, K., *Computational Graph Theory: Characterization, Enumeration and Generation of Chemical Structures by Computer Methods*; Ellis Horwood: New York, 1991.

(7) Bohacek, R. S.; McMartin, C.; Guida, W. C. The art and practice of structure-based drug design: a molecular modeling perspective. *Med. Res. Rev.* **1996**, *16* (1), 3–50.

(8) Fink, T.; Bruggesser, H.; Reymond, J. L. Virtual exploration of the small-molecule chemical universe below 160 Da. *Angew. Chem., Int. Ed. Engl.* **2005**, *44* (10), 1504–8.

(9) Fink, T.; Reymond, J. L. Virtual exploration of the chemical universe up to 11 atoms of C, N, O, F: assembly of 26.4 million structures (110.9 million stereoisomers) and analysis for new ring systems, stereochemistry, physicochemical properties, compound classes, and drug discovery. *J. Chem. Inf. Model.* **2007**, *47* (2), 342–53.

(10) Miranker, A.; Karplus, M. Functionality Maps of Binding-Sites - a Multiple Copy Simultaneous Search Method. *Proteins: Struct., Funct., Genet.* **1991**, *11* (1), 29–34.

(11) Gillet, V. J.; Johnson, A. P.; Mata, P.; Sike, S. Automated structure design in 3D. *Tetrahedron Comput. Methodol.* **1990**, *3*, 681–696.

(12) Gillet, V.; Johnson, A. P.; Mata, P.; Sike, S.; Williams, P. SPROUT: a program for structure generation. *J. Comput.-Aided Mol. Des.* **1993**, *7* (2), 127–53.

(13) Gillet, V. J.; Newell, W.; Mata, P.; Myatt, G.; Sike, S.; Zsoldos, Z.; Johnson, A. P. SPROUT: recent developments in the de novo design of molecules. *J. Chem. Inf. Comput. Sci.* **1994**, *34* (1), 207–17.

(14) Bohm, H. J. The computer program LUDI: a new method for the de novo design of enzyme inhibitors. *J. Comput.-Aided Mol. Des.* **1992**, *6* (1), 61–78.

(15) Verlinde, C. L.; Rudenko, G.; Hol, W. G. In search of new lead compounds for trypanosomiasis drug design: a protein structure-based linked-fragment approach. *J. Comput.-Aided Mol. Des.* **1992**, *6* (2), 131–47.

(16) Rotstein, S. H.; Murcko, M. A. GenStar: a method for de novo drug design. *J. Comput.-Aided Mol. Des.* **1993**, *7* (1), 23–43.

(17) Pearlman, D. A.; Murcko, M. A. Concepts - New Dynamic Algorithm for De-Novo Drug Suggestion. *J. Comput. Chem.* **1993**, *14* (10), 1184–1193.

(18) Clark, D. E.; Frenkel, D.; Levy, S. A.; Li, J.; Murray, C. W.; Robson, B.; Waszkowycz, B.; Westhead, D. R. PRO-LIGAND: an approach to de novo molecular design. 1. Application to the design of organic molecules. *J. Comput.-Aided Mol. Des.* **1995**, *9* (1), 13–32.

(19) DeWitte, R. S.; Shakhnovich, E. SMoG: de novo design method based on simple, fast and accurate free energy estimates. 1. Methodology and supporting evidence. *J. Am. Chem. Soc.* **1996**, *118*, 11733–11744.

(20) Gehlhaar, D. K.; Moerder, K. E.; Zichi, D.; Sherman, C. J.; Ogden, R. C.; Freer, S. T. De novo design of enzyme inhibitors by Monte Carlo ligand generation. *J. Med. Chem.* **1995**, *38* (3), 466–72.

(21) Miranker, A.; Karplus, M. An automated method for dynamic ligand design. *Proteins* **1995**, *23* (4), 472–90.

(22) Roe, D. C.; Kuntz, I. D. BUILDER v.2: improving the chemistry of a de novo design strategy. *J. Comput.-Aided Mol. Des.* **1995**, *9* (3), 269–82.

(23) Nishibata, Y.; Itai, A. Automatic Creation of Drug Candidate Structures Based On Receptor Structure - Starting Point For Artificial Lead Generation. *Tetrahedron* **1991**, *47* (43), 8985–8990.

(24) Nishibata, Y.; Itai, A. Confirmation of Usefulness of a Structure Construction Program Based on 3-Dimensional Receptor Structure for Rational Lead Generation. *J. Med. Chem.* **1993**, *36* (20), 2921–2928.

(25) Bohacek, R. S.; Mcmartin, C. Multiple Highly Diverse Structures Complementary to Enzyme Binding-Sites - Results of Extensive Application of a De-Novo Design Method Incorporating Combinatorial Growth. *J. Am. Chem. Soc.* **1994**, *116* (13), 5560–5571.

(26) Luo, Z. W.; Wang, R. X.; Lai, L. H. RASSE: A new method for structure-based drug design. *J. Chem. Inf. Comput. Sci.* **1996**, *36* (6), 1187–1194.

(27) Rotstein, S. H.; Murcko, M. A. GroupBuild: a fragment-based method for de novo drug design. *J. Med. Chem.* **1993**, *36* (12), 1700–10.

(28) Moon, J. B.; Howe, W. J. Computer design of bioactive molecules: a method for receptor-based de novo ligand design. *Proteins* **1991**, *11* (4), 314–28.

(29) Bohm, H. J. Towards the automatic design of synthetically accessible protein ligands: peptides, amides and peptidomimetics. *J. Comput.-Aided Mol. Des.* **1996**, *10* (4), 265–72.

(30) Corey, E. J.; Howe, W. J.; Orf, H. W.; Pensak, D. A.; Petersson, G. General methods of synthetic analysis. Strategic bond disconnections for bridged polycyclic structures. *J. Am. Chem. Soc.* **1975**, *97* (21), 6116–6124.

(31) Corey, E. J.; Jorgensen, W. L. Computer-assisted synthetic analysis. Generation of synthetic sequences involving sequential functional group interchanges. *J. Am. Chem. Soc.* **1975**, *98* (1), 203–209.

(32) Corey, E. J.; Jorgensen, W. L. Computer-assisted synthetic analysis. Synthetic strategies based on appendages and the use of reconnective transforms. *J. Am. Chem. Soc.* **1976**, *98* (1), 189–203.

(33) Corey, E. J.; Long, A. K.; Greene, T. W.; Miller, J. W. Computer-assisted synthetic analysis. Selection of protective groups for multistep organic synthesis. *J. Org. Chem.* **1985**, *50* (11), 1920–1927.

(34) Hendrickson, J. B. A general protocol for systematic synthesis design. *Top. Curr. Chem.* **1976**, *62*, 49–172.

(35) Hendrickson, J. B. Approaching the Logic of Synthesis Design. *Acc. Chem. Res.* **1986**, *19*, 274–281.

(36) Hendrickson, J. B.; Miller, T. M. Reaction Classification and Retrieval. A Linkage between Synthesis Generation and Reaction Databases. *J. Am. Chem. Soc.* **1991**, *113*, 902–910.

(37) Hendrickson, J. B.; Parks, C. A. A Program for the Forward Generation of Synthetic Routes. *J. Chem. Inf. Comput. Sci.* **1992**, *32*, 209–215.

(38) Hendrickson, J. B.; Walker, M. A. A two-component pericyclic reaction for synthesis of substituted benzofurans and aryl-quaternary carbon bonds. *Org. Lett.* **2000**, *2* (18), 2729–31.

(39) Hendrickson, J. B.; Wang, J. A new synthesis of lysergic acid. *Org. Lett.* **2004**, *6* (1), 3–5.

(40) Gillet, V. J.; Myatt, G.; Zsoldos, Z.; Johnson, A. P. SPROUT, HIPPO and CAESA: tools for de novo structure generation and estimation of synthetic accessibility. *Perspect. Drug Discovery Des.* **1995**, *3*, 34–50.

(41) Boda, K.; Seidel, T.; Gasteiger, J. Structure and reaction based evaluation of synthetic accessibility. *J. Comput.-Aided Mol. Des.* **2007**, *21* (6), 311–25.

(42) Makino, S.; Ewing, T. J.; Kuntz, I. D. DREAM++: flexible docking program for virtual combinatorial libraries. *J. Comput.-Aided Mol. Des.* **1999**, *13* (5), 513–32.

(43) 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* (5), 487–94.

(44) Jorgensen, W. L.; Ruiz-Caro, J.; Tirado-Rives, J.; Basavapathruni, A.; Anderson, K. S.; Hamilton, A. D. Computer-aided design of non-nucleoside inhibitors of HIV-1 reverse transcriptase. *Bioorg. Med. Chem. Lett.* **2006**, *16* (3), 663–7.

(45) Lipinski, C. A.; Lombardo, F.; Dominy, B. W.; Feeney, P. J. Experimental and computational approaches to estimate solubility and permeability in drug discovery and development settings. *Adv. Drug Delivery Rev.* **1997**, *23*, 3–25.

(46) Wang, R.; Gao, Y.; Lai, L. LigBuilder: A Multi-Purpose Program for Structure-Based Drug Design. *J. Mol. Model.* **2000**, *6*, 498–516.

(47) Frimurer, T. M.; Bywater, R.; Naerum, L.; Lauritsen, L. N.; Brunak, S. Improving the odds in discriminating "drug-like" from "non drug-like" compounds. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (6), 1315–24.

(48) Takaoka, Y.; Endo, Y.; Yamanobe, S.; Kakinuma, H.; Okubo, T.; Shimazaki, Y.; Ota, T.; Sumiya, S.; Yoshikawa, K. Development of a method for evaluating drug-likeness and ease of synthesis using a data set in which compounds are assigned scores based on chemists' intuition. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (4), 1269–75.

(49) Schneider, N.; Jackels, C.; Andres, C.; Hutter, M. C. Gradual in silico filtering for druglike substances. *J. Chem. Inf. Model.* **2008**, *48* (3), 613–28.

(50) Harper, G.; Bradshaw, J.; Gittins, J. C.; Green, D. V.; Leach, A. R. Prediction of biological activity for high-throughput screening using binary kernel discrimination. *J. Chem. Inf. Comput. Sci.* **2001**, *41* (5), 1295–300.

(51) Chen, B.; Harrison, R. F.; Pasupa, K.; Willett, P.; Wilton, D. J.; Wood, D. J.; Lewell, X. Q. Virtual screening using binary kernel discrimination: effect of noisy training data and the optimization of performance. *J. Chem. Inf. Model.* **2006**, *46* (2), 478–86.

(52) Sadowski, J.; Kubinyi, H. A scoring scheme for discriminating between drugs and nondrugs. *J. Med. Chem.* **1998**, *41* (18), 3325–9.

(53) Frimurer, T. M.; Bywater, R.; Naerum, L.; Lauritsen, L. N.; Brunak, S. Improving the odds in discriminating ”drug-like” from ”non drug-like” compounds. *J. Chem. Inf. Comput. Sci.* **2000**, *40* (6), 1315–24.

(54) Murcia-Soler, M.; Perez-Gimenez, F.; Garcia-March, F. J.; Salabert-Salvador, M. T.; Diaz-Villanueva, W.; Castro-Bleda, M. J. Drugs and nondrugs: an effective discrimination with topological methods and artificial neural networks. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (5), 1688–702.

(55) Lang, S. A.; Kozyukov, A. V.; Balakin, K. V.; Skorenko, A. V.; Ivashchenko, A. A.; Savchuk, N. P. Classification scheme for the design of serine protease targeted compound libraries. *J. Comput.-Aided. Mol. Des.* **2002**, *16* (11), 803–7.

(56) Balakin, K. V.; Tkachenko, S. E.; Lang, S. A.; Okun, I.; Ivashchenko, A. A.; Savchuk, N. P. Property-based design of GPCR-targeted library. *J. Chem. Inf. Comput. Sci.* **2002**, *42* (6), 1332–42.

(57) Balakin, K. V.; Lang, S. A.; Skorenko, A. V.; Tkachenko, S. E.; Ivashchenko, A. A.; Savchuk, N. P. Structure-based versus property-based approaches in the design of G-protein-coupled receptor-targeted libraries. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (5), 1553–62.

(58) Muresan, S.; Sadowski, J. "In-house likeness": comparison of large compound collections using artificial neural networks. *J. Chem. Inf. Model.* **2005**, *45* (4), 888–93.

(59) Ajay, A.; Walters, W. P.; Murcko, M. A. Can we learn to distinguish between "drug-like" and "nondrug-like" molecules. *J. Med. Chem.* **1998**, *41* (18), 3314–24.

(60) Ajay, A.; Bemis, G. W.; Murcko, M. A. Designing libraries with CNS activity. *J. Med. Chem.* **1999**, *42* (24), 4942–51.

(61) Niwa, T. Prediction of biological targets using probabilistic neural networks and atom-type descriptors. *J. Med. Chem.* **2004**, *47* (10), 2645–50.

(62) Murcia-Soler, M.; Perez-Gimenez, F.; Garcia-March, F. J.; Salabert-Salvador, M. T.; Diaz-Villanueva, W.; Castro-Bleda, M. J.; Villanueva-Pareja, A. Artificial neural networks and linear discriminant analysis: a valuable combination in the selection of new antibacterial compounds. *J. Chem. Inf. Comput. Sci.* **2004**, *44* (3), 1031–41.

(63) Engkvist, O.; Wrede, P.; Rester, U. Prediction of CNS activity of compound libraries using substructure analysis. *J. Chem. Inf. Comput. Sci.* **2003**, *43* (1), 155–60.

(64) Hutter, M. C. Separating drugs from nondrugs: a statistical approach using atom pair distributions. *J. Chem. Inf. Model.* **2007**, *47* (1), 186–94.

(65) ChemBank. http://chembank.broad.harvard.edu/welcome.htm (accessed July 6, 2006).

(66) NCI Open Database. http://cactus.nci.nih.gov/ncidb2/download.html (accessed July 6, 2006).

(67) Daylight Theory Manual. In Daylight Chemical Information Systems, Inc.: Aliso Viejo, CA, 2008.

(68) ChemAxon. http://www.chemaxon.com (accessed July 6, 2006).

(69) de Silva, K. M.; Goodman, J. M. What is the smallest saturated acyclic alkane that cannot be made. *J. Chem. Inf. Model.* **2005**, *45* (1), 81–7.

(70) Weininger, D. Smiles, a Chemical Language and Information-System 0.1. Introduction to Methodology and Encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28* (1), 31–36.

(71) Badger, G. M. *The Structures and Reactions of the Aromatic Compounds*; Cambridge University Press: Cambridge, 1954.

(72) Veber, D. F.; Johnson, S. R.; Cheng, H. Y.; Smith, B. R.; Ward, K. W.; Kopple, K. D. Molecular properties that influence the oral bioavailability of drug candidates. *J. Med. Chem.* **2002**, *45* (12), 2615–23.

(73) Ho, C. M.; Marshall, G. R. DBMAKER: a set of programs to generate three-dimensional databases based upon user-specified criteria. *J. Comput.-Aided Mol. Des.* **1995**, *9* (1), 65–86.

(74) Nilakantan, R.; Bauman, N.; Venkataraghavan, R. A Method for Automatic-Generation of Novel Chemical Structures and Its Potential Applications to Drug Discovery. *J. Chem. Inf. Comput. Sci.* **1991**, *31* (4), 527–530.

(75) Cosgrove, D. A.; Kenny, P. W. BOOMSLANG: a program for combinatorial structure generation. *J. Mol. Graph.* **1996**, *14* (1), 1–5, 23.

(76) Zheng, W.; Cho, S. J.; Tropsha, A. Rational combinatorial library design. 1. Focus-2D: a new approach to the design of targeted combinatorial chemical libraries. *J. Chem. Inf. Comput. Sci.* **1998**, *38* (2), 251–8.

(77) Clark, D. E.; Firth, M. A.; Murray, C. W. MOLMAKER: De novo generation of 3D databases for use in drug design. *J. Chem. Inf. Comput. Sci.* **1996**, *36* (1), 137–145.

(78) Pirok, G.; Mate, N.; Varga, J.; Szegezdi, J.; Vargyas, M.; Dorant, S.; Csizmadia, F. Making ”real” molecules in virtual space. *J. Chem. Inf. Model.* **2006**, *46* (2), 563–8.

(79) Cramer, R. D.; Soltanshahi, F.; Jilek, R.; Campbell, B. AllChem: generating and searching 10(20) synthetically accessible structures. *J. Comput.-Aided Mol. Des.* **2007**, *21* (6), 341–50.

CI9000458