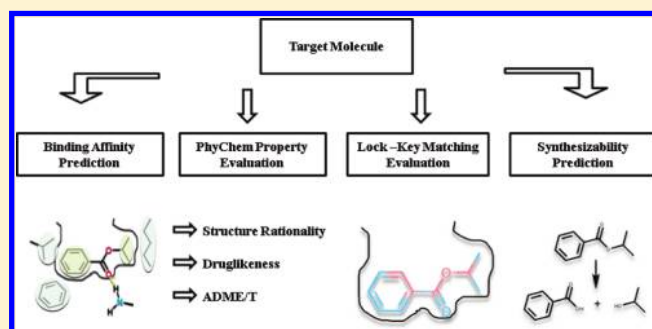


LigBuilder 2: A Practical *de Novo* Drug Design Approach

Yaxia Yuan,^{†,‡} Jianfeng Pei,^{*,‡} and Luhua Lai^{*,†,‡}[†]BNLMS, State Key Laboratory for Structural Chemistry of Unstable and Stable Species, College of Chemistry and Molecular Engineering, Peking University, Beijing 100871, China[‡]Center for Theoretical Biology, Academy for Advanced Interdisciplinary Studies, Peking University, Beijing 100871, China

S Supporting Information

ABSTRACT: We have developed a new version (2.0) of the *de novo* drug design program LigBuilder. With LigBuilder 2.0, the synthesis accessibility of designed compounds can be analyzed, and a cavity detection procedure is implemented to detect the positions and shapes of the binding sites on the surface of a given protein structure and to quantitatively estimate drugability. Ligands are designed to best fit the detected cavities using a set of rules for evaluation. Drug-like and privileged fragments are used to construct the ligands with the aid of internal and external absorption, distribution, metabolism, excretion, and toxicity (ADME/T) and drug-like filters.



INTRODUCTION

High throughput screening (HTS) is commonly used for novel drug lead discovery. As an *in silico* counterpart to HTS, virtual screening (VS), in particular protein–ligand docking techniques, has become increasingly used in the past decade. However, one weakness of the VS technique is that identified hits are limited to the compound libraries used. In addition, it is often difficult to further modify or optimize hits from their original scaffolds because they are “picked” from molecules with high scores, which usually arise from exhaustive exploration of a prosaic scaffold rather than a distinctive scaffold.

Compared to VS, computational *de novo* drug design can produce novel molecular entities without structural limitations and produce highly efficient scaffolds with the required pharmacological profiles. Considering the immensity of the drug-like chemical space, *de novo* design can be a more viable method for identifying drug candidates. Several *de novo* drug design programs, such as LEGEND,¹ LUDI,² SPROUT,³ HOOK,⁴ GrowMol,⁵ PRO-LIGAND,⁶ CONCERTS,⁷ RASSE,⁸ LeapFrog,⁹ LEA3D,¹⁰ and LigBuilder,¹¹ have been developed in the last two decades. These tools can be classified according to different criteria such as molecular building block (e.g., atom- or fragment-based), chemical space searching strategy (e.g., evolutionary algorithm or Monte Carlo simulation), and scoring method used for evaluation of the designed molecules (e.g., force field, empirical scoring function, or molecular similarity).^{12–14} Some of these programs have been applied to the lead generation practice in drug discovery projects. For example, evolutionary *de novo* design by TOPAS led to the identification of a new molecular scaffold as a lead structure candidate for a novel Kv1.5 channel blocking agent.¹⁵ A novel hepatitis C virus helicase inhibitor was discovered using LigBuilder.¹⁶ *De novo* drug design

programs also serve as molecular optimization tools. For example, LUDI was used to generate ideas for *de novo* design of new ligands (e.g., FKBP-12) based on the core structure of FK506 protein.¹⁷ LigBuilder 1.2 was used by Goldberg et al. to optimize a p38 MAP kinase inhibitor with the link function of the program.¹⁸ To accomplish this, LigBuilder 1.2 was first used to suggest 800 structures by bridging leader structure and fragments from a fragment library. Compounds were then manually designed based on these structures and considerations for synthetic flexibility, compound rigidity, and optimal positioning of the pyridine ring for interaction with Met109, which partake in an H-bond with the N-7 of the adenine ring of ATP. Ultimately, a new class of inhibitors was successfully obtained that could bind to p38 MAP kinase in a Phe-out conformation.

Despite these successes, computational *de novo* drug design methods have not been widely adopted by medicinal chemists in routine drug discovery. Compared to the success of docking-based VS programs, only a few hits and leads have been discovered using *de novo* drug design programs. One major problem with current *de novo* drug design programs is that most molecules produced by these algorithms are difficult to synthesize. Moreover, *de novo* design is not a high throughput approach like HTS or VS. Synthesis of compounds designed *de novo* is often labor-intensive and time-consuming due to the involvement of numerous scaffolds, as such, a practical *de novo* drug design program that can develop drug candidates with a high success rate (i.e., fewer false positives). In other words, a highly accurate scoring function is needed, in particular one that is more demanding than that of VS as compounds selected using this

Received: September 5, 2010

Published: April 22, 2011

approach can normally be purchased for testing. For example, a common phenomenon in *de novo* drug design is the generation of large ligands as greater molecular size tends to achieve higher binding affinity. However, this facet often poses a high risk of false positives since key interactions and shape complementarity between protein and ligand are neglected.

We developed a multipurpose program called LigBuilder 1.0 (released version 1.2)¹¹ based on our previously developed program RASSE⁸ for structure-based *de novo* drug design. Within the three-dimensional structural constraints of the target protein, LigBuilder 1.0 uses a genetic algorithm to construct ligands iteratively using a library of organic fragments. Various operations, such as growing, linking, and mutation, are implemented to manipulate the molecular structure, and the user can choose either the growing or linking strategy for ligand construction. The protein–ligand binding affinity is evaluated by an empirical scoring function called SCORE 2.0,¹⁹ and bioavailability is evaluated using a set of chemical rules. In our report, we demonstrated the ability of LigBuilder 1.0 to generate chemical structures similar to thrombin and dihydrofolate reductase.¹¹ Since its release, LigBuilder 1.2 has been widely used and applied successfully in a number of drug design projects.^{16,18,20,21}

To increase the applicability of computational *de novo* drug design, we have developed a new generation of this program, LigBuilder 2.0. With this version, the synthesis accessibility of designed compounds can be analyzed with the aid of an embedded chemical reaction database and a retro-synthesis analyzer. A cavity detection procedure is also implemented to detect the positions and shapes of binding sites on the surface of a given protein structure and to quantitatively estimate their drugability. Moreover, using a set of evaluation rules, ligands can be designed to best fit the identified cavities. Drug-like and privileged fragments can be used to construct ligands with the aid of internal and external absorption, distribution, metabolism, excretion, and toxicity (ADME/T) and drug-like filters. In addition to these novel functions, LigBuilder 2.0 inherited all the existing algorithms, features, and libraries used in LigBuilder 1.2. Here we introduce the new features and improvements of LigBuilder 2.0.

■ CAVITY DETECTION AND RECEPTOR-BASED PHARMACOPHORE MODEL

A well-defined binding pocket is critical for reliable ligand design. In fact, identification of protein cavities is of fundamental importance for a range of applications, including molecular docking, *de novo* drug design, allosteric site discovery, and comparison of functional sites. We developed a novel cavity detection procedure called Cavity 1.0 to examine the entire surface or a given area of a target protein (a more detailed description of Cavity 1.0 will be reported elsewhere). This software uses a probe sphere and navigates the protein surface to simulate the microcosmic kinetic processes used by small molecules during their attempt to bind the protein. This step can be imagined as a fictitious ball rolling around the surface of a protein to detect the inaccessible volume. Since detected cavities typically become connected, we define the depth of a binding cavity as the distance from the surface to the bottom and disjoin the linkage area of cavities within a depth threshold. Using a test set consisting of 1300 protein–ligand complexes, Cavity 1.0 demonstrated 86% accuracy in predicting correct binding sites as the top-ranked pocket with a ranking score evaluated by the

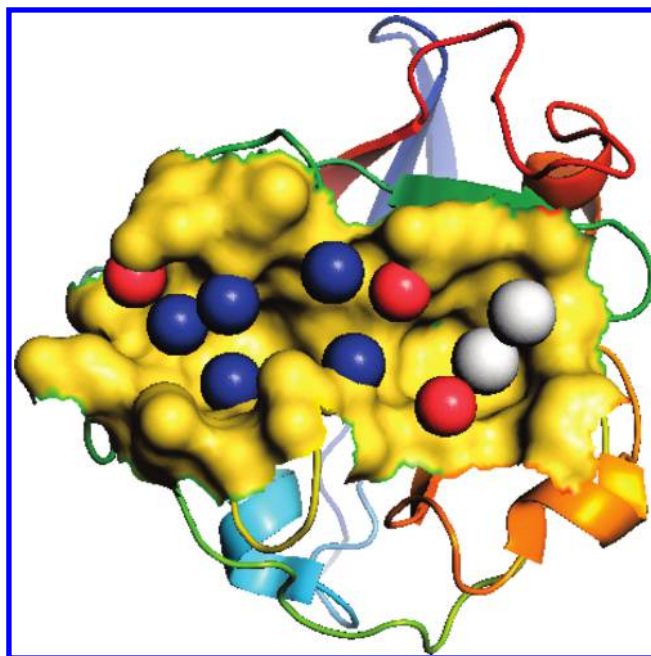


Figure 1. Binding pocket and pharmacophore model of Cyclophilin A produced by Cavity 1.0. Green indicates the surface of the protein–ligand binding site, and the colored spheres depict the pharmacophore features of the binding site. Red indicates the hydrogen-bond acceptor area, dark blue defines the hydrogen-bond donor area, and white indicates the hydrophobic area.

volume, surface area, H-bond, and hydrophobic properties of the pockets. When testing a set of 35 unbound proteins, 83% of the predicted top-ranked pockets were actual ligand-binding sites, thereby demonstrating a much higher success rate than other cavity detection programs.^{22–24}

The cavities detected by this software can provide clear, accurate information about the shapes and boundaries of ligand binding sites. As a result, ligands are designed to fit within the binding cavities and overgrowth is eliminated. Cavity 1.0 also has a unique feature that can calculate a score for each cavity as a function of geometric shape, hydrogen bonding, and hydrophobic effect. We have shown a linear relationship between this score and the binding affinity range of existing ligands that can bind the cavity. That is, given a known binding pocket, the binding constant range of designed ligands can be estimated according to its Cavity 1.0 score, which is a useful index for drug design and optimization. We previously demonstrated the use of Cavity 1.0 to detect a new binding pocket of cyclin-dependent kinase 2 and the design of potent inhibitors capable of binding this enzyme.²⁵ Supplementary Figure S2 demonstrates the improvements made to this module for this study. The Cavity 1.0 program can be downloaded at <http://mdl.ipc.pku.edu.cn/>.

We had also developed a standalone program, Pocket v.2,²⁶ based on the original Pocket module in LigBuilder 1.0. Pocket v.2 can derive pharmacophore models (i.e., key interaction sites) directly from given protein receptor structures or protein–ligand structures without human intervention. Cavity 1.0 used the same strategy as Pocket v.2 to define the pharmacophore features within cavities. The pharmacophore features could provide a visual representation of the properties of the binding site. Furthermore, the pharmacophore features were used to guide the mutation operation inherited from LigBuilder 1.0. The

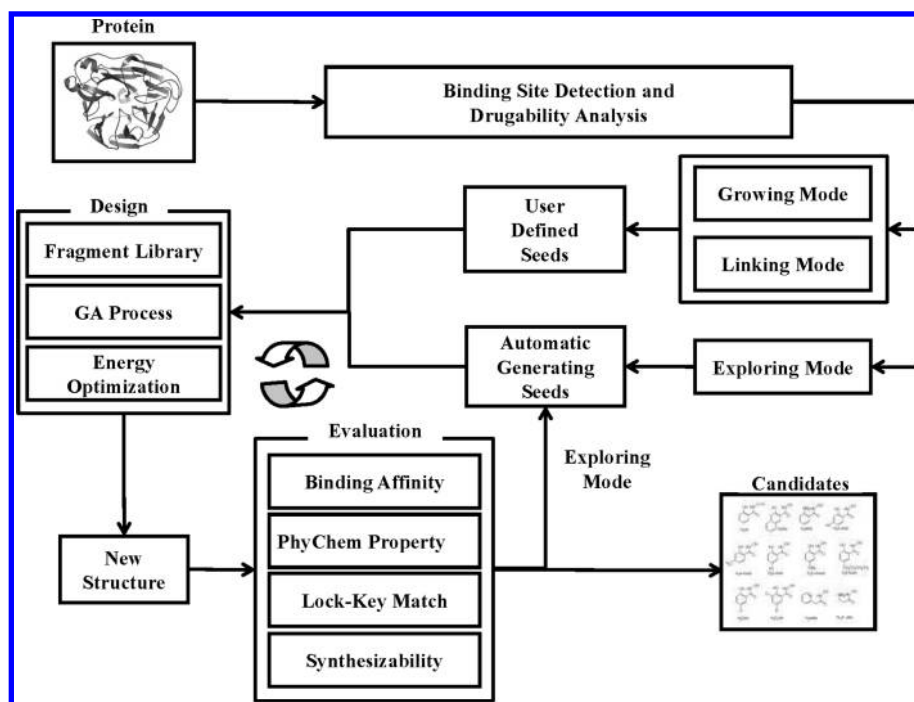


Figure 2. The design strategy of LigBuilder 2.0.

binding pocket surface and key features of the pharmacophore model produced by Cavity 1.0 for the structure of Cyclophilin A (PDB code 1nmk) are shown in Figure 1.

■ IMPROVED DESIGN STRATEGY AND FRAGMENT DATABASE

LigBuilder 2.0 provides the user with three strategies for constructing ligands, namely the growing and linking strategies inherited from LigBuilder 1.0 and a new exploring strategy developed specifically for *de novo* molecule design. The genetic algorithm from LigBuilder 1.0 was used to control the process of building up the ligand in growing, linking, and exploring. With the growing strategy, this process starts from a seed structure that has been preplaced into the binding pocket. The user can assign certain “growing sites” on the seed structure, and the program will then attempt to replace each site with a candidate fragment. This newly formed structure serves as the seed for the next growing cycle, and the process continues until the ligand is fully designed. With the linking strategy, the build-up process also starts from a preplaced seed structure. However, in this case, the seed consists of several distinct pieces that are positioned to maximize the interactions with the target protein. Every fragment grows simultaneously on each piece, and the program continuously tries to link these pieces by attempting to form rational bonds. This process continues until all the pieces have been integrated into one molecule. In addition to these two strategies that use a starting structure, the “Drug Space Exploring Algorithm” was introduced into the design process. LigBuilder 2.0 can analyze newly formed molecules and extract high contribution fragments to a seed structure pool, which will supply seeds at the beginning of each design cycle. With this algorithm, LigBuilder 2.0 can avoid the limitations associated with preassigned seed structures and explore a broader space, thus greatly improving design efficiency. This new design strategy of LigBuilder 2.0 is diagrammed in Figure 2.

Additional improvements in design were also implemented in LigBuilder 2.0. For example, local energy minimization is applied at each step in growing or linking to adjust the positions of the designed ligands (or fragments) and thus avoid accumulating minor errors. LigBuilder 2.0 employs a stochastic optimization method to optimize the intramolecular energy of a ligand by minimizing torsion and intermolecular binding energies through ligand pose perturbation. Another improvement enables pharmacophore features (or key interaction sites) within the binding pocket produced by Cavity 1.0 to be used as a guide for subsequent ligand construction processes. Finally, seed structure is no longer a requirement for the user in exploring mode. If no preset seed structures are available, LigBuilder 2.0 can automatically use an sp^3 carbon as the starting point to construct ligands and extract valid seed structures from these ligands as subsequent starting points.

LigBuilder 2.0 uses a fragment-based algorithm to construct molecules (Figure 3). The term “fragment” here describes the building blocks used in the construction process. The rationale of this algorithm lies in the fact that organic structures can be deconstructed into basic chemical fragments. Although the diversity of organic structures is infinite, the number of basic fragments is rather limited. Fifty-seven elementary and “complex” fragments were used in LigBuilder 1.0 for ligand construction. However, in LigBuilder 2.0, more drug-like and privileged fragments (listed in Supplementary Figure S1, extracted from the World Drug Index²⁷) were added into the fragment database to accelerate the ligand construction process and the development of more “drug-like” ligands. As an open library, users can add their own fragments of interest (e.g., the focused fragment library) or remove unwanted ones. LigBuilder 2.0 also allows the user to supply an external fragment library called the “forbidden substructure library” to further filter the molecules generated. Therefore, the user can deposit into this library any undesired chemical structure from among the resultant

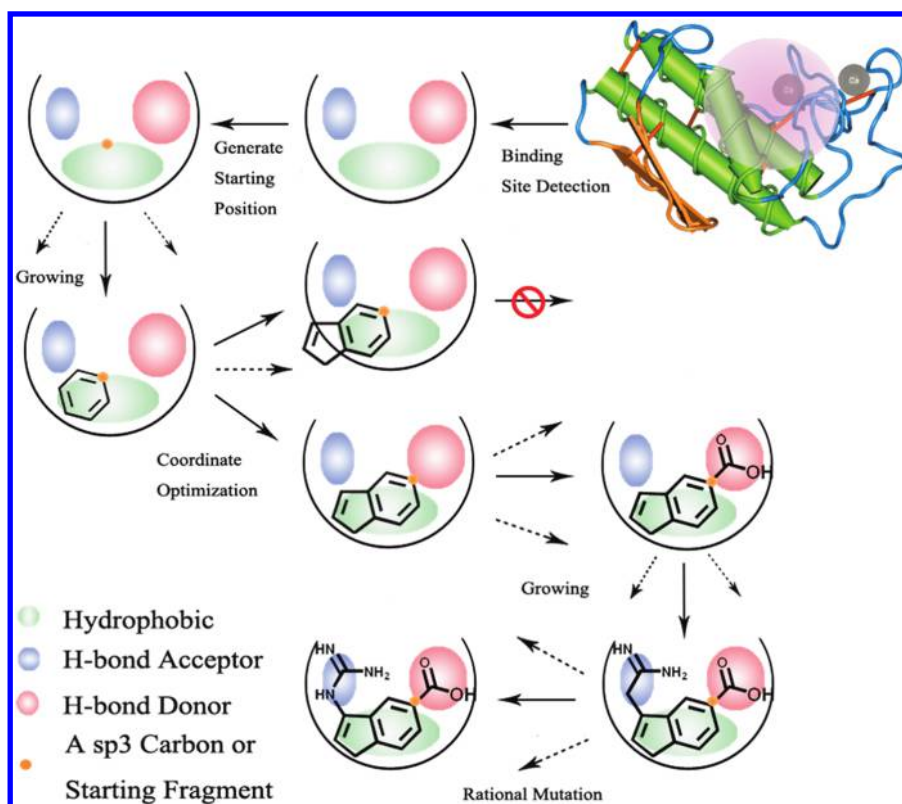


Figure 3. Scheme of fragment growing algorithm of LigBuilder 2.0.

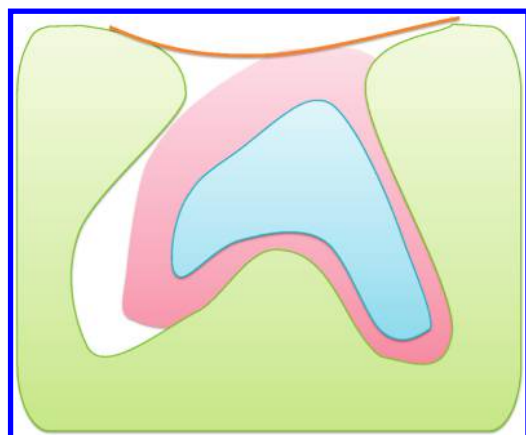


Figure 4. A sketch map of the lock-key model match algorithm. The binding site, ligand, and expandable area are denoted by green, blue, and red, respectively. The orange line delineates the boundary between the binding site and solvent.

molecules. To accomplish this, the program checks each molecule using a substructure mapping algorithm, and a molecule will be rejected if it contains any of the forbidden fragments.

■ SHAPE COMPLIMENTARITY MATCHING AND LI-GAND EVALUATION

The lock-key model is applied to evaluate conformational matching between the protein and ligand. We hypothesize that an effective drug molecule binds its receptor with good shape complementarity. Thus, ligands are designed to fit the shape of

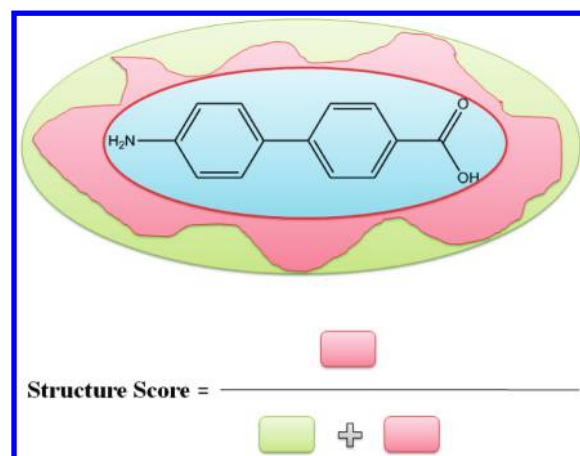


Figure 5. Definition of the structure score. The collision, cushion, and ligand-occupied space are denoted by green, red, and blue, respectively. The structure score is defined as the percentage of the motion accessible space around the ligand.

the binding pocket generated by Cavity 1.0 as closely as possible. Any potential collision or cushion space between protein and ligand is disfavored and a penalty score is assigned. We define a protein–ligand noncomplimentarity index using the following steps: 1) Make a three-dimensional grid within the binding cavities of the protein with a cushion distance of 0.5 Å. 2) Calculate the van der Waals surface of the ligand and protein-binding site. 3) Expand the van der Waals surface (maximal expanding length equals 2.5 Å) of the ligand in 0.5 Å increments until it collides with the protein van der Waals surface or the

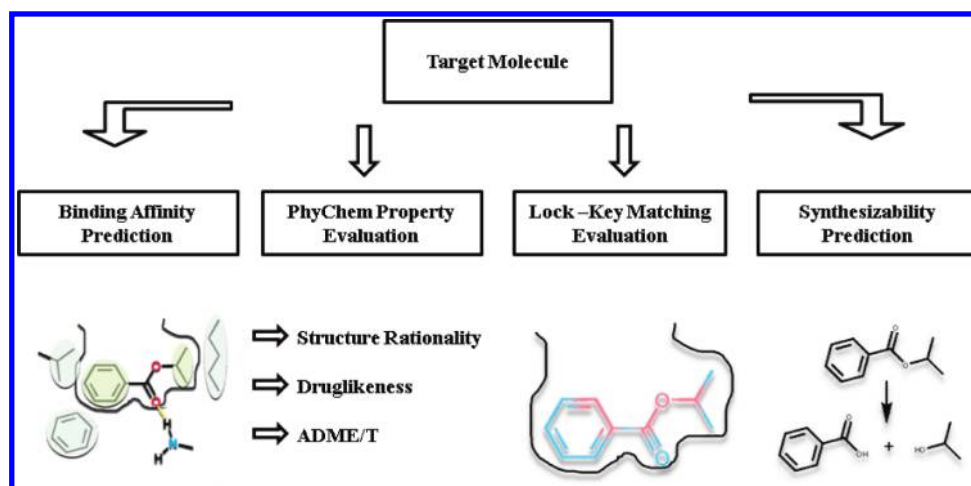


Figure 6. Schema of the scoring and evaluation functions applied in LigBuilder 2.0.

boundary of the cavity as detected by Cavity 1.0. The number of covered grids in these steps represents an index for protein–ligand noncomplementarity, and reflects the motion ability of the ligand within the receptor binding site (shown in Figure 4). We used a structure score as defined in Figure 5 to describe the motion ability.

To evaluate protein–ligand binding, we developed an improved version of SCORE. This program uses an empirical scoring function to calculate protein–ligand binding affinity. Based on previous versions, namely SCORE 2.0¹⁹ and SCORE 3.0,²⁸ we applied an algorithm that more accurately evaluates hydrogen bonding by assessing the direction of lone pairs of hydrogen-bond acceptor atoms. The new scoring function was retrained using the PDBBIND²⁹ refined set containing 800 protein–ligand complexes with known experimental binding affinities. The linear correlation coefficient (*R*) and standard deviation of the predictions with experimental binding affinities are 0.68 and 1.8 kcal/mol, respectively. Details of the improved version of the scoring function will be described elsewhere.

Other evaluations also implemented into LigBuilder 2.0 were as follows: ligand properties such as structure stability; simple ADME/T properties such as Log *P*; druglikeness filter to limit the Log*P* of ligands in the range of −0.4 to 5.6; a molecular weight range of 160 to 480 Da; and a range of total atom number between 20 and 70.³⁰ In addition, external drug-like and ADME/T filters can be used to further filter results generated by LigBuilder 2.0. Users can also apply the ligand efficiency option to promote the design of smaller ligands with similar binding affinities. An overview of the ligand evaluation processes is shown in Figure 6.

SYNTHESIS ACCESSIBILITY

Synthesis accessibility analysis is used as an aid in selecting compounds that are easy to synthesize at low cost. We based our synthesis accessibility algorithm on retro-synthetic analysis, a technique for solving problems in the planning of organic syntheses with the goal of structural simplification. A target molecule is transformed into simpler precursor structures without assumptions regarding the starting materials. Each precursor material is examined by the same method, and this procedure is repeated until simple or commercially available structures are obtained. Oftentimes more than one possible route can be used

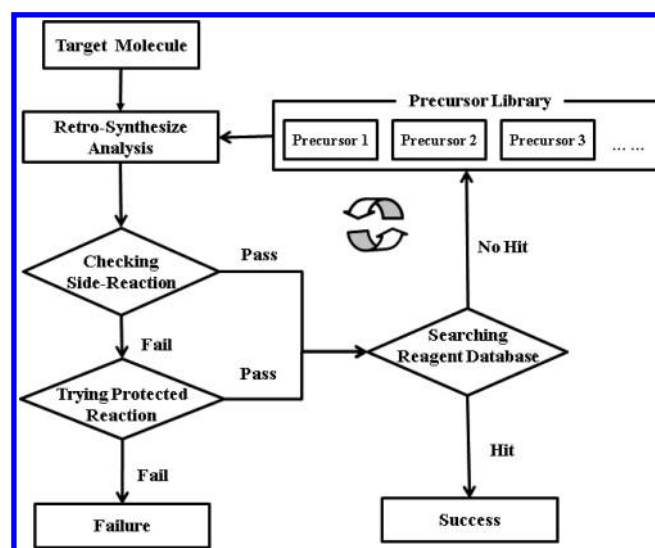


Figure 7. Flowchart of the retro-synthesis analysis process.

to synthesize a target molecule. Retro-synthesis analysis is well-suited for discovering different synthetic routes and comparing them in a logical and straightforward fashion.³¹

In our synthesis accessibility analysis algorithm, a database was constructed that contains not only combination reactions used for reaction searches but also chemical group introduction, bond-cutting, and elimination information, all of which are used for group-based analysis. A protection and positioning database was also constructed for analysis of protected reactions and aromatic ring positioning. Because side reactions are a key index for assessing the availability of a reaction, we generated a side-reaction database. Finally, a reagent database was built as the fragment terminals of retro-synthesis. A flowchart of the retro-synthesis analysis process is depicted in Figure 7.

LigBuilder 2.0 using an object structure-based strategy to process the retro-synthesize analysis. The algorithm finds a potential reaction route of a target molecule by identifying structural features of the reaction. Thus, four types of information for this particular reaction were obtained from the reaction database, namely its structural features, bond-forming features, conditions, and production yield. For example, Figure 8

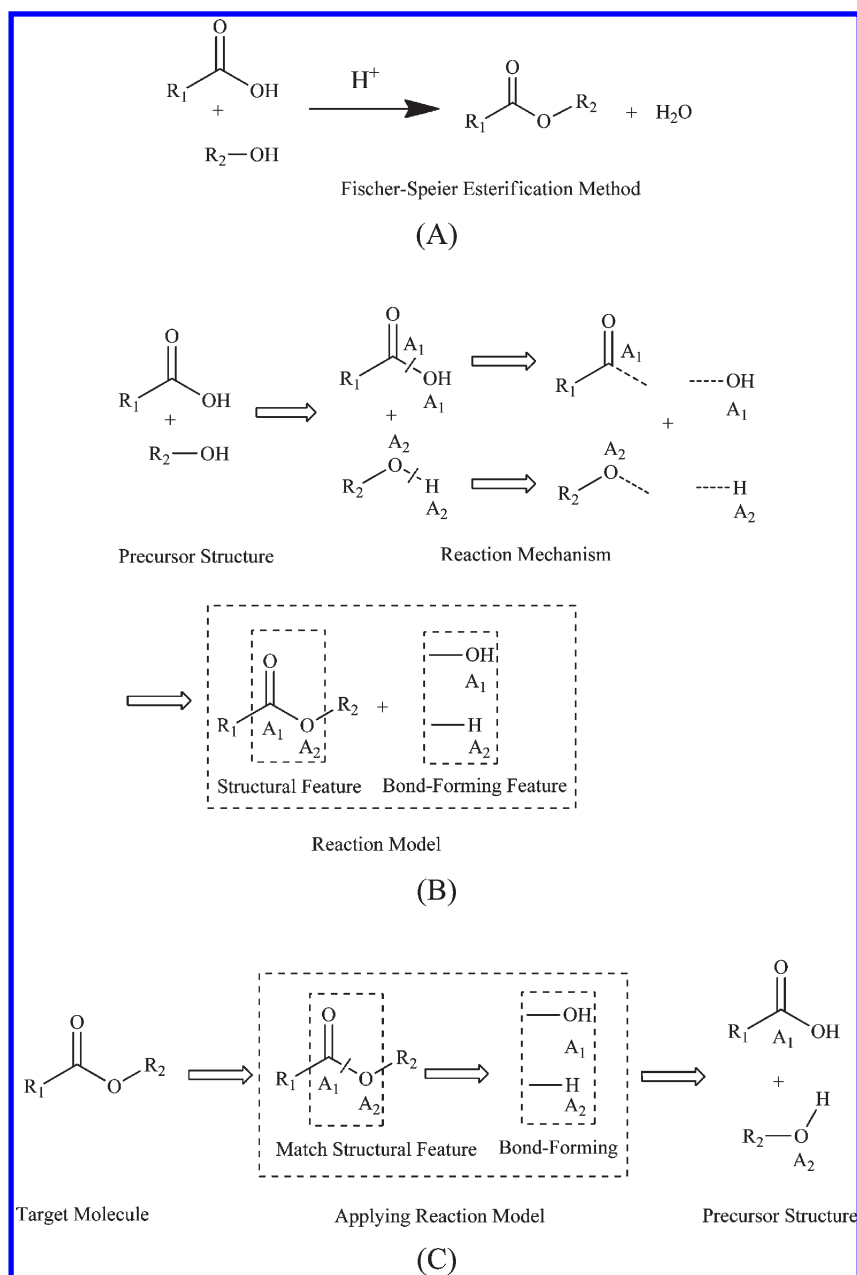


Figure 8. Esterification reaction as an example for retro-synthesis analysis.³² (A) The Fischer–Speier Esterification reaction.³² (B) The nonessential structural features of the reactants are neglected, and the key reactive atoms are marked according to the reaction mechanism. In this case, the carbon atom and the hydroxyl oxygen atom of the acid are marked as ‘A1’ and ‘A2’, respectively. The reaction model is composed of structural and bond-forming features. (C) The ‘Match Structural Feature’ box on the left shows the matching process of ester structural feature. The ‘Bond-Forming Feature’ box on the right shows the structural transformation by bond-forming. In this case, the bond between ‘A1’ and ‘A2’ is cut, and the atoms form new bonds with a hydroxyl and a hydrogen atom, respectively.

illustrates the mechanism of an esterification reaction with all of the atoms and bonds depicted. The bond-forming and structural features of this reaction are also defined. For synthesis accessibility analysis, our algorithm will recognize whether a target molecule contains an ester component and then mark the bond-cutting positions prior to cleavage between the marked atoms. Corresponding fragments are attached to these atoms to form the reaction precursor structures according to the bond-formation features of the reaction in the reaction database.

For cases in which a reaction is carried out by several consecutive steps, we simplified the reactions into a single step

and ignored the intermediate ones to accelerate the retro-synthesis analysis process. For example, the synthesis routine of ar-turmerone was simplified to a one-step reaction using our synthesis module (Figure 9). Side reactions as well as protection and position reactions were simplified into a single step by omitting the inducing and deducing processes of assistant chemical groups.

Each reaction has been assigned an initial yield of 90% as the default value. The algorithm thus calculates the adjusted yield of a reaction by taking into account several characteristics that can affect the yield, such as the atom type, bulky bump, and electron

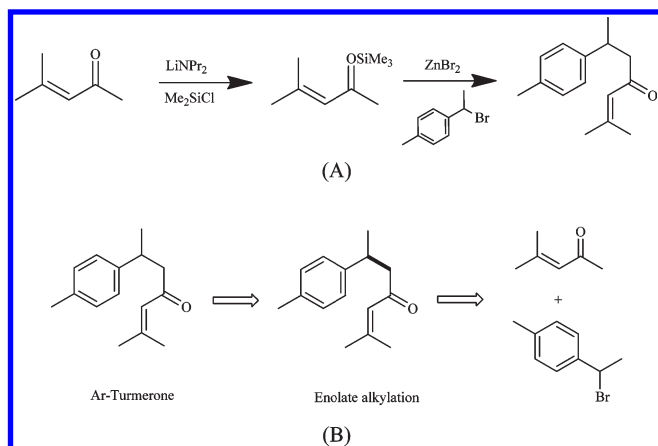


Figure 9. Multistep reactions can be simplified into a single-step reaction. (A) The process of synthesizing ar-turmerone experimentally (Reprinted with permission of Elsevier from ref. 33, copyright Elsevier 1979). (B) The retro-synthesis routine performed by LigBuilder 2.0 assembles all steps of the reaction into a whole.

effect. Although external factors, including temperature, pH, catalyst, and oxidation–reduction conditions, may also affect the reaction, these are not considered because they can be resolved experimentally. If the calculated yield is above a cutoff value, then a side-reaction check will be performed using a retro-process of the analysis procedure. Precursor structures are evaluated again by the reaction database for other products besides the target structure that could be produced under the same reaction conditions. If side products could be formed, then the yield of the target structure is recalculated to take the side-reaction into consideration. If this examination shows that the production rate falls below 30%, then the protection and position reactions are loaded to search for an alternative reaction route. As mentioned above, the protection and positioning groups are virtually added and removed by an integrated single-step reaction. Since the side reactions omitted in the simplification step may raise the risk of inaccurate analysis, protection and positioning groups are only examined when the target structure is difficult to synthesize using a more direct route.

The retro-synthesis procedure is repeated until commercially available structures are obtained. Total computation time may be quite long, especially when the population size in the genetic algorithm is large. Since individuals in a population are somewhat similar in the genetic design process, the algorithm uses a molecular database to store frequently occurring precursors to avoid the need for repetitious computations.

Since the chemical space is much larger than the synthesis-accessible compound space, with the molecular weight increasing as the procedure progresses, synthesis-accessible molecules become more vulnerable in the process of genetic algorithm, the proportion of which declines sharply. Therefore, molecules that are difficult to synthesize at each generation of the genetic algorithm must be eliminated to maintain the superiority of synthesis accessible compounds. The analysis algorithm returns the highest yields of all possible routes to synthesize the target structure, which is used by the genetic algorithm to evaluate synthesis accessibility. When the scaffold of the target molecule is considered accessible for synthesis, its derivatives are most likely also synthetically accessible. Moreover, fragments under a certain size are considered easy to obtain. In these cases, scaffold priority

strategy and empirical fragments terminal rules were used to accelerate the evaluation process in each intermediate generation. Thus, the retro-synthesis procedure inclined to apply reactions involving scaffold and only separate the structure into several small fragments to meet the rough empirical terminal rules in three to five steps. However, in the last generation of the genetic algorithm, the entire retro-synthesis analysis is performed until a commercially available reagent is obtained.

The reactions stored in the organic reaction databases were mainly referenced from <http://www.chempensoftware.com/organicreactions.htm> and classic organic chemistry textbooks. Reactions with critical conditions or low production rates were not included in the database. The current database contains 493 elementary organic reactions without redundancy. The permitted reagents for constructing a compound were collected from a commercially available small molecules database and structures known to be easily synthesized were included. The reagent database, which contains 6,961,470 reagents, was obtained from the ZINC 8 database³⁴ following removal of incorrect and duplicate reagents. Due to its size, this database is difficult to search by direct match. Therefore, we used the ELF hash algorithm to construct a simplified hash database to accelerate the search. The hash algorithm sometimes converts different reagents into the same hash string, so we used an enormous hash table to greatly reduce this possibility for a rapid searching.

DISCUSSION

We have developed a new generation of a *de novo* structure-based drug design program, LigBuilder 2.0, which has greater applicability to actual drug design. This program implements several state-of-the-art techniques that may be of interest to drug designers. Based on recent publications on *de novo* drug design and our own experience, the bottleneck for *de novo* design programs, including LigBuilder 1.0, is the difficulty in actually synthesizing the designed molecules. This issue has been resolved in LigBuilder 2.0 by introducing an embedded synthesis accessibility analyzer that limits design results to a synthesis accessible chemical space. This feature greatly improves design efficiency. To demonstrate this, the SYLVIA³⁵ online Web service was used to compare the synthetic accessibility of molecules designed for the Cyclophilin A inhibitor with the growing mode of LigBuilder 1.2 and 2.0 as well as the exploring mode of LigBuilder 2.0. SYLVIA uses a scale of 1 to 10 to represent the synthesis accessibility of a molecule, where 1 indicates straightforward synthesis and 10 implies a structure that could be difficult to synthesize. The average synthetic accessibility indexes confirmed the increase in synthetic accessibility with using LigBuilder 2.0 compared to previous versions. Further details regarding these molecules are included in Supplementary Figure S3.

For the purpose of designing true positive ligands, several major changes were made in LigBuilder 2.0. One was an improvement in the scoring function. We developed a new version of SCORE that examines hydrogen-bond interactions and trained it using a large training set to improve the scoring accuracy. The other alteration was the development of an accurate cavity detection program, Cavity 1.0. Cavities detected by the program are not only used to limit the shape of designed ligands to avoid overgrowth into unwanted regions but also to calculate a structure score for the bound ligand. This score has

proven to be a good index for discriminating true versus false binding. Furthermore, the “Drug Space Exploring Algorithm” was incorporated into LigBuilder 2.0. With this strategy, we could eliminate the dependence of guidance from users. In addition, prior knowledge of the molecular template, fragments, leaders, and starting seed structures is no longer required. Thus LigBuilder 2.0 could be expected to achieve a much higher diversity compared with other *de novo* design methods. We have tested LigBuilder 2.0 in designing and optimizing Cyclophilin A inhibitors. By using only a one-round design, a novel inhibitor with greater potency than the positive control, cyclosporin A, was discovered.³⁶ This successful *de novo* drug design example verified the effectiveness of LigBuilder 2.0.

Besides being a good tool to *ab initio* generate leads without given initial structures, LigBuilder 2.0 can also serve as an effective tool for lead optimization. Users can select critical fragments of a lead, remove unwanted ones, and then use the “grow” and “link” strategies of LigBuilder 2.0 for optimization. Scaffold changes to a lead can also be achieved easily with this program. LigBuilder 2.0 compliments the “fragment-based drug design” methods available currently. Moreover, this program can be used to computationally link and optimize the experimentally validated “fragments” with slight conformation adjustments. Further improvements to LigBuilder 2.0, such as developing more accurate scoring functions and implementing multitarget *de novo* design function, are under development.

The Cavity 1.0 program can be downloaded at <http://mdl.ipc.pku.edu.cn/>. Academic users can acquire the LigBuilder 2.0 package by contacting the authors.

■ ASSOCIATED CONTENT

S Supporting Information. The building block fragments of LigBuilder 2.0 are listed in the Supporting Information section. A demonstration of improvements in binding-site definition and synthetic accessibility of LigBuilder 2.0 is also included. This material is available free of charge via the Internet at <http://pubs.acs.org>.

■ AUTHOR INFORMATION

Corresponding Author

*E-mail: jfpei@pku.edu.cn (J.F.P.); lhilai@pku.edu.cn (L.H.L.).

■ ACKNOWLEDGMENT

This project was supported in part by the Ministry of Science and Technology of China and the National Natural Science Foundation of China (20703002, 10721403, and 90913021).

■ REFERENCES

- (1) Nishibata, Y.; Itai, A. Automatic creation of drug candidate structures based on receptor structure. Starting point for artificial lead generation. *Tetrahedron* **1991**, *47*, 8885–8990.
- (2) 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*, 61–78.
- (3) Gillet, V.; Johnson, A. P.; Mata, P.; Sike, S.; Williams, P. SPROUT: A program for structure generation. *J. Comput.-Aided Mol. Des.* **1993**, *7*, 127–153.
- (4) Eisen, M. B.; Wiley, D. C.; Karplus, M.; Hubbard, R. E. HOOK: A program for finding novel molecular architectures that satisfy the

chemical and steric requirements of a macromolecule binding site. *Proteins* **1994**, *19*, 199–221.

- (5) Bohacek, R. S.; McMartin, C. Multiple highly diverse structures complementary to enzyme binding sites: Results of extensive application of *de novo* design method incorporating combinatorial growth. *J. Am. Chem. Soc.* **1994**, *116*, 5560–5571.

- (6) 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*, 13–32.

- (7) Pearlman, D. A.; Murcko, M. A. CONCERTS: Dynamic connection of fragments as an approach to *de novo* ligand design. *J. Med. Chem.* **1996**, *39*, 1651–1663.

- (8) 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*, 1187–1194.

- (9) LeapFrog; Tripos, Inc.: St Louis, MO, USA.

- (10) LEA3D: A Computer-Aided Ligand Design for Structure-Based Drug Design. *J. Med. Chem.* **2005**, *48*, 2457–2468

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

- (12) Schneider, G.; Fechner, U. Computer-based *de novo* design of drug-like molecules. *Nat. Rev. Drug Discovery* **2005**, *4*, 649–663.

- (13) Schneider, G.; Hartenfeller, M.; Reutlinger, M.; Tanrikulu, Y.; Proschak, E.; Schneider, P. Voyages to the (un)known: adaptive design of bioactive compounds. *Trends Biotechnol.* **2009**, *27* (1), 18–26.

- (14) Mauser, H.; Guba, W. Recent developments in *de novo* design and scaffold hopping. *Curr. Opin. Drug Discovery Dev.* **2008**, *11*, 365–374.

- (15) 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.

- (16) Kandil, S.; Biondaro, S.; Vlachakis, D.; Cummins, A. C.; Coluccia, A.; Berry, C.; Leyssen, P.; Neyts, J.; Brancale, A. Discovery of a novel HCV helicase inhibitor by a *de novo* drug design approach. *Bioorg. Med. Chem. Lett.* **2009**, *19* (11), 2935–2937.

- (17) Babine, R. E.; Bleckman, T. M.; Kissinger, C. R.; Showalter, R.; Pelletier, L. A.; Lewis, C.; Tucker, K.; Moomaw, E.; Parge, H. E.; Villafranca, J. E. Design, Synthesis and X-Ray Crystallographic Studies of Novel Fkbp-12 Ligands. *Bioorg. Med. Chem. Lett.* **1995**, *3*, 1719–1724.

- (18) Goldberg, D. R.; Hao, M. H.; Qian, K. C.; Swinamer, A. D.; Gao, D. H. A.; Xiong, Z.; Sarko, C.; Berry, A.; Lord, J.; Magolda, R. L.; Fadra, T.; Kroe, R. R.; Kukulka, A.; Madwed, J. B.; Martin, L.; Pargellis, C.; Skow, D.; Song, J. J.; Tan, Z. L.; Torcellini, C. A.; Zimmitti, C. S.; Yee, N. K.; Moss, N. Discovery and optimization of p38 Inhibitors via computer-assisted drug design. *J. Med. Chem.* **2007**, *50*, 4016–4026.

- (19) Wang, R. X.; Liu, L.; Lai, L. H.; Tang, Y. Q. SCORE: A new empirical method for estimating the binding affinity of a protein-ligand complex. *J. Mol. Model.* **1998**, *4*, 379–394.

- (20) Yates, A. S.; Doughty, S. W.; Kendall, D. A.; Kellam, B. Chemical modification of the naphthoyl 3-position of JWH-015: In search of a fluorescent probe to the cannabinoid CB2 receptor. *Bioorg. Med. Chem. Lett.* **2005**, *15*, 3758–3762.

- (21) Tondel, K.; Drablos, F. Design of selective inhibitors of tyrosine kinase 2. *Letts. Drug Des. Discovery* **2005**, *2*, 507–515.

- (22) Huang, B. D.; Schroeder, M. LIGSITE(csc): predicting ligand binding sites using the Connolly surface and degree of conservation. *BMC Struct. Biol.* **2006**, *6*, ARTN 19.

- (23) Laurie, A. T. R.; Jackson, R. M. Q-SiteFinder: an energy-based method for the prediction of protein-ligand binding sites. *Bioinformatics* **2005**, *21* (9), 1908–1916.

- (24) Laskowski, R. A. Surfnets - a Program for Visualizing Molecular Surfaces, Cavities, and Intermolecular Interactions. *J. Mol. Graphics* **1995**, *13* (5), 323.

- (25) Chen, H.; Van Duyne, R.; Zhang, N.; Kashanchi, F.; Zeng, C. A novel binding pocket of cyclin-dependent kinase 2. *Proteins* **2009**, *74*, 122–132.

(26) Chen, J.; Lai, L. H. pocket v.2: Further developments on receptor-based pharmacophore modeling. *J. Chem. Inf. Model.* **2006**, *46*, 2684–2691.

(27) Derwent Information Ltd., Derwent House, 14 Great Queen Street, London, WC2B 5DF, UK. Web address: <http://www.derwent.co.uk/> (accessed Feb. 2002).

(28) Pei, J. F.; Wang, Q.; Zhou, J. J.; Lai, L. H. Estimating protein-ligand binding free energy: Atomic solvation parameters for partition coefficient and solvation free energy calculation. *Proteins* **2004**, *57*, 651–664.

(29) Wang, R. X.; Fang, X. L.; Lu, Y. P.; Yang, C. Y.; Wang, S. M. The PDBbind database: Methodologies and updates. *J. Med. Chem.* **2005**, *48*, 4111–4119.

(30) Ghose, A. K.; Viswanadhan, V. N.; John, J.; Wendoloski, J. J. A Knowledge-Based Approach in Designing Combinatorial or Medicinal Chemistry Libraries for Drug Discovery. 1. A Qualitative and Quantitative Characterization of Known Drug Databases. *J. Comb. Chem.* **1999**, *1* (1), 55–68.

(31) Law, J.; Zsoldos, Z.; Simon, A.; Reid, D.; Liu, Y.; Khew, S. Y.; Johnson, A. P.; Major, S.; Wade, R. A.; Ando, H. Y. Route Designer: A Retrosynthetic Analysis Tool Utilizing Automated Retrosynthetic Rule Generation. *J. Chem. Inf. Model.* **2009**, *49*, 593–602.

(32) Fischer, E.; Speier, A. Darstellung der Ester. *Chem. Ber.* **1895**, *28*, 3252–3258.

(33) Paterson, I. α -Alkylation of ketones, esters, and lactones: zinc bromide-catalysed alkylation of *o*-silylated enolates, a synthesis of (\pm)-ar-turmerone. *Tetrahedron Lett.* **1979**, *20*, 1519–1520.

(34) 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.

(35) Boda, K.; Seidel, T.; Gasteiger, J. Structure and reaction based evaluation of synthetic accessibility. *J. Comput.-Aided Mol. Des.* **2007**, *21*, 311–325. Online web service: http://www.molecular-networks.com/online_demos/sylvia/ (accessed Jan. 16, 2011).

(36) Ni, S. S.; Yuan, Y. X.; Huang, J.; Mao, X. N.; Lv, M. S.; Zhu, J.; Shen, X.; Pei, J. F.; Lai, L. H.; Jiang, H. L.; Li, J. Discovering Potent Small Molecule Inhibitors of Cyclophilin A Using *de novo* Drug Design Approach. *J. Med. Chem.* **2009**, *52*, 5295–5298.