OpenGrowth: An Automated and Rational Algorithm for Finding New Protein Ligands
Nicolas Chéron, Naveen Jasty, Eugene I Shakhnovich

To cite this version:

HAL Id: hal-02973969
https://hal.science/hal-02973969
Submitted on 21 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
OpenGrowth: an automated and rational algorithm for finding new protein ligands

1Nicolas Chéron, 1Naveen Jasty, and 1Eugene I. Shakhnovich*

1Department of Chemistry and Chemical Biology, Harvard University, Cambridge MA 02138, USA
Abstract

We present a new open-source software, called OpenGrowth, which aims to create de novo ligands by connecting small organic fragments in the active site of proteins. Molecule growth is biased to produce structures that statistically resemble drugs in an input training database. Consequently, the produced molecules have superior synthetic accessibility and pharmacokinetic properties than randomly grown molecules. The growth process can take into account the flexibility of the target protein and can be started from a seed to mimic R-group strategy or fragment-based drug discovery. Primary applications of the software on the HIV-1 protease allowed us to quickly identify new inhibitors with a predicted $K_d$ as low as 18nM. We also present a graphical user interface that allows a user to select easily the fragments to include in the growth process. OpenGrowth is released under the GNU GPL licence and is available free of charge on the author’s website and at http://opengrowth.sourceforge.net/.

Keywords

de novo design, drug discovery, protein inhibitors, ligand growth, combinatorial chemistry, HIV-1 protease.
Introduction

Computational tools are now fully integrated in all steps of structure-based drug discovery, from the early stage of homology modelling¹ to free energy perturbations used during lead optimization² or predictions of pharmacokinetic properties³. High-throughput virtual screenings (and especially molecular docking) have proven to be reliable tools to identify new hits⁴ and are widely used in drug discovery. Once a target protein has been identified, this kind of approach requires a predefined library of compounds to screen. The most straightforward libraries consist of existing molecules, either commercially available ones (such as the ZINC library⁵), a set of natural products, in-house compounds or current drugs/drug candidates. This methodology leads either directly to new candidates or to hits that will later be optimized. The former, a “drug repurposing” approach, has proven to be both efficient and effective⁶. A prime example is raloxifene, which is now used to treat both osteoporosis and breast cancer⁷. However, drug repurposing is limited in scope, especially when we aim to find original scaffolds. Increases in computing power and in automation of biochemical assays forecast that large libraries (if not the full ZINC library) will in the near future have been docked or tested in almost all target proteins with known structures. Since the needs for new ways to cure diseases will never stop –because of either drug-resistance⁸ or the emergence of new diseases–, we cannot rely only on already existing molecules or scaffolds in drug discovery. Thereby, there is a need to discover new chemical materials.

So-called computational de novo drug design approaches have emerged to overcome the limitation of existing libraries. In such strategies, a computer generates novel ligands for a target protein. One must note that in addition to protein ligands, de novo design has also been applied to solar cells⁹ and polymers¹⁰. Due to the importance of the field, de novo drug design methods have already been reviewed on several occasions¹¹ and a list of available programs and successes of these methods are available in these reviews. We want to present here a brief overview on some of the possible methods to construct the new molecules. First, a common approach is to link two fragments that have been
previously docked in the active site (LUDI\textsuperscript{12} and PhDD\textsuperscript{13} are programs that do this). The BREED program \textit{recombines} molecules by exchanging fragments after the identification of overlapping bonds\textsuperscript{14}. Another approach, used by FLUX\textsuperscript{15}, \textit{mutates} fragments from bound ligands using retrosynthetic rules such as RECAP\textsuperscript{16}. A fourth strategy –inspired by biology– uses \textit{evolutionary algorithms}: each moiety of a known ligand is considered a gene that can evolve\textsuperscript{17}. Many recently published programs use this method, such as AutoGrow\textsuperscript{18}. Starting from a seed, a \textit{R-group} strategy can also be used: fragments are added around the scaffold to decorate it\textsuperscript{19}. This strategy is not unlike the strategy used by medicinal chemists after the identification of a hit. Finally, \textit{growing} methods have been successfully implemented, where new fragments (or atoms) are added at a time until a given threshold is reached. This approach shares several similarities to the \textit{R-group} strategy, as the latter is a growing method with an evolving seed. Recombination, mutation, evolution and \textit{R-group} approaches need first to have information about existing ligands, whereas only linking and growing approaches can create completely new ligands.

Two sorts of programs using a growing strategy exist. First, it is possible to grow libraries without considering a protein: the produced molecules are then docked by regular means in the target protein. Thus, there is no need to use three-dimensional representation of molecules: for instance, we described in 2009 an algorithm (called FOG) that analyses a training drug database in terms of probabilities that two fragments are connected\textsuperscript{20}. New libraries of molecules (that occupy a desired chemical space) are then grown using these probabilities and written as SMILES strings. The second family of program, by far the largest, uses three-dimensional description of the ligands in the active site of a protein. The early approaches were mainly atom-based, i.e. new structures were created atom by atom (e.g. LEGEND\textsuperscript{21} (1991), GenStar\textsuperscript{22} (1993) or RASSE\textsuperscript{23} (1996)). However, the last notable purely atom-based program was published in 1996\textsuperscript{11a, 23} and most of the recent algorithms are now fragment-based. In 1993, one of the first fragment-based growth algorithms, GroupBuild\textsuperscript{24}, was published. Using a starting fragment (or seed), GroupBuild generates a set of new ligands and randomly keeps one of the top-scored ligands.
before proceeding to the next step. GrowMol\textsuperscript{25} (1994) is both atom and fragment-based and was the first program to use a Metropolis criterion to accept or reject the new fragment. Our group developed the SMoG algorithm\textsuperscript{26} in 1996 whose effectiveness was demonstrated in 2002\textsuperscript{27}. However, the ligands grown by SMoG often lacked “chemical realism” and SMoG was resultantly implemented less than expected. In 2006, BOMB\textsuperscript{28} and FlexNovo\textsuperscript{29} were released. The details of the BOMB algorithms have never been published and it is thus difficult to compare it with other approaches; FlexNovo was commercial and is no more available for download. LigBuilder was first published in 2000, with a second version presented in 2011\textsuperscript{30}; it is described below. In 2012, Vitae Pharmaceuticals described Contour, an internal project that is based on SMoG\textsuperscript{31}; since it is proprietary software, it is not available to the public.

Thereby, to the best of our knowledge, only one program which uses a growing approach and which is non-proprietary or non-commercial is currently available to the public: LigBuilder (which is free for academic users, but not open-source). This program grows molecules in "growing sites" that must have been pre-defined. Thus, only fragments that fit the shape of the cavity (according to some predefined rules) can be added. In order to increase synthetic accessibility, LigBuilder uses retrosynthetic analysis in combination with connection rules (e.g., O-O bonds cannot be created). We present here a new program that uses other strategies as the ones in LigBuilder. First, we do not define in advance sites of growth: should the placement of a new fragment in a site be not favorable, our approach will reject the fragment (because of energetic criterion or detection of steric clashes). The reason of the existence of the cavities in LigBuilder is to avoid overgrowth: since our program is very fast, we can relax this constraint and our approach does not need a prior preparation of the target protein. Concerning the synthetic accessibility, the use of retrosynthetic analysis or a set of chemical rules to create new molecules is efficient, but can also be too restrictive and may not represent the diversity offered by organic synthesis. For example, the flexibility offered by protection groups is difficult to implement in computational retrosynthetic analysis. Moreover, when a set of chemical reactions is used (such as in SYNOPSIS\textsuperscript{32} or DOGS\textsuperscript{33}), the set must be
updated when new synthesis pathways are discovered. In contrast, we have preferred to use statistics extracted from training drug databases. This allows us to generate realistic sets of ligands whose properties will match the ones from the training database and which can be more easily synthesized. Moreover, while the current version of LigBuilder can only work with one target, the program presented here has the possibility to take into account protein flexibility or to create multitarget compounds by growing ligands in different target proteins at the same time. Finally, since our program is open-source external contributors can provide new ideas or implement other scoring functions whereas the sources of LigBuilder are not accessible.

The program that is described in this manuscript can create completely de novo ligands using a growing approach (even when no actual ligands are known since only the coordinates of the active site are needed). It is also possible to start from a seed when prior experimental data are known. Depending on the size of the seed, this workflow can mimic a R-group strategy where the aim is to decorate a first hit, or a fragment-based drug discovery strategy where a small binding fragment is evolved to a small molecule. The approach used by the program resembles the one in SMoG in that it grows molecules in the active site of a protein by connecting small organic fragments. However, SMoG ligands often lacked “chemical realism” and we aim to bring this “chemical realism” to our growth method. Since SMoG was released almost two decades ago, we decided to start the development of the new tool from scratch. Our program (OpenGrowth) is free, open-source, flexible and modular, easy to download, easy to install and easy to use. Amongst the important specifications we wanted, one can cite: taking into account protein flexibility from the beginning; having a flexible list of fragments; producing new ligands that are easy to synthesize and have good pharmacokinetic properties (ADME). To achieve the latter and bring the missing “chemical realism”, it is crucial to choose new fragments in a reasonable way. We rely in the use of statistics of connection between fragments as defined in FOG; if the training drug database is big enough, significant information can be extracted and will produce compounds occupying a desired
chemical space (i.e. drug-like molecules). Thus, our program is the first one to grow three-dimensional ligands using statistical analysis of training databases and having the possibility to take into account protein flexibility or to discover multi-target drugs, while being free, open-source and easy to use. The possibility to support fragment-based drug discovery with a seed approach makes OpenGrowth a very versatile tool for computational de novo drug design and open wide opportunities regarding its use.

Methods

Growing molecules

The growth begins by placing one of the fragments in the active site of the target protein; new fragments are then added to the current ligand until a given threshold is reached (“fragment” is used as a synonymous as “building block” throughout the text). Potential thresholds include molecular weight, number of atoms, or number of fragments (see Figure 1). The ligand is saved and then a new growth starts iteratively. The aim of the program is to grow hundreds of thousands of new ligands that can then be sorted by applying additional search criteria. The connection between the current state of the ligand and the next fragment is made by removing one hydrogen atom from each component and creating a new single bond between the two atoms to which the hydrogens were connected. The distance at which the new moiety is added depends on the type of the two atoms (for example, 1.54 Å between two sp³ carbon atoms and 1.46 Å between sp³ and sp carbon atoms). For each new fragment added, a rotameric search is performed about the new bond and the orientation with the lowest binding score and without steric clash is kept. A geometry optimization in the (fixed) active site of the protein can also be performed to relax the ligand structure. More details on the beginning of the growth, adding new fragments and optimizing the ligand geometry and its position are given in the Supporting Information. The new ligand score is then compared to the previous score (before fragment addition), and a Monte-Carlo approach is used to accept or reject the fragment34. If the new score is lower than the previous one, the new fragment is accepted; if the new score is higher than the previous one, the new fragment is
accepted with a Metropolis criterion probability $\exp(-\Delta S/RT)$ where $\Delta S = S_{\text{new}} - S_{\text{old}}$ is the difference of score and $RT$ is a parameter given by the user. Thus, the growth tends to decrease the binding energy of the small molecules to the protein. Occasionally an increase in binding energy is allowed, which is useful in cases like crossing a tight channel.

![Diagram](image)

**Figure 1.** Principle of the OpenGrowth algorithm. After docking a primary fragment, new fragments are selected. For each one, once the best orientation is found a Metropolis criterion is used to either accept or reject the fragment. When the ligand reaches a given threshold property, it is saved, it can be filtered, and a new growth starts.

**Drug library**

Several aspects of the program use a drug library for training (for example, the construction of the fragment libraries or the evaluation of statistics of connection between fragments, see below). Before going further, we want to describe quickly the library that was used. To prepare this set of molecules, we downloaded the full library of ChEMBL Drugstore (https://www.ebi.ac.uk/chembl/drugstore) current as of February 18th 2014 (with the reference
chembl_drugs-14_19-23-20$^{35}$. This set contains 10,342 molecules, among which we kept the 8,145 containing a SMILES entry. This subset library was then cleaned by removing redundant molecules, single atom cations (Na$^+$, K$^+$, Mg$^{2+}$, Ca$^{2+}$, Li$^+$, Zn$^{2+}$, Al$^{3+}$), molecules with Si, Al, As, Pt, Bi, Sb atoms, molecules labelled as “inorganic” or “polymer”, and small molecules such as bromoform, chloroform, carbon tetrachloride, etc... At the end, we obtained a library of 8,018 molecules. Not all these molecules have been approved by regulation agencies and some are still in clinical trials. However, we do not intend to capture by our algorithms the reasons why molecules may fail in clinical phase: our goal is first to create good protein binders. Since our approach uses statistical analysis of the drug library, we envisioned that statistics that are more accurate would be obtained with a larger library and we kept molecules that are still in clinical trials.

**Fragment library**

The choice of a proper fragment library is crucial for the success of a growth algorithm. As we will present later, we have developed a graphical interface for the user to choose which fragments he/she wants to use amongst a predefined list. To prepare this list, we sorted fragments in different classes. First, we present the list of non-ring fragments in Figure 2. As a general principle, we tried to avoid denaturalizing the fragments. For example, an amide could theoretically be made during the growth by joining aldehyde and amine. However, we considered the nitrogen of an amide too different from that of an amine; thus, we added an amide fragment to the non-ring set. This approach is similar to the one used in FOG$^{20}$. Some fragments presented in Figure 2 are trimers of other fragments (last line, for example urea could be made by joining amine, aldehyde and amine); the reason for the existence of the trimer fragments will be explained later. Some of the fragments in Figure 2 do not seem to be drug-like, such as carbonic acid (last line, first fragment); nonetheless, this fragment was actually found in the drug database. However it was found very rarely, thus it is very likely that the algorithm will almost never select it. A user can easily decide to remove this fragment from the list as explained in the user manual.
Some organic fragments are not in the list since they are rarely found in drugs: azide (H-N3) was found 7 times in our drug library, sulfoxide (O=SH₂) was found 5 times and thionitrosyl (HN=S) only 2 times. Moreover, we did not keep any fragments involving a phosphorus atom since most of these are unstable to hydrolysis; thus, a molecule containing a fragment with a phosphorus atom will probably not be considered drug-like at first (even if some drugs contain phosphorus atoms). One can manually enter new fragments in the list as well; we describe this process in the user manual. Nevertheless, the included set of non-ring fragments covers almost all the possibilities offered by organic chemistry.

![Non-ring fragments](image)

*Figure 2. List of non-ring fragments.*

The choice of ring fragments is more complicated since many possibilities exist (note that in our algorithms two non-ring fragments cannot join to form a ring). We first extracted all the rings from the drug library by cutting all single bonds between rings and side chains. We achieved that by using the *fragmenter* tool from JChem 6.1.4 (ChemAxon, 2013, http://www.chemaxon.com). We can then assume that the rings that were found in our drug library (made of 8,018 molecules from ChEMBL) represent all the rings that can be found in drugs. We then classified this set of rings according to the number of rings fused together per fragment. We found the following numbers of different rings: mono-ring: 280; 2-fused-rings: 444; 3-fused-rings: 361; 4-fused-rings: 171; 5-fused-rings: 77; 6-fused-rings: 28; 7-fused-rings: 8; 8-fused-rings: 6; 9-fused-rings: 5; 11-fused-rings: 3. We point out that in our definition, all double bonds connected to a ring are a part of the ring fragment (since we have cut single bonds to
prepare the set of rings and we create single bonds during the growth). Thus, cyclopentane and cyclopentanone are considered different rings and are treated as such. To achieve this difference of treatment, we describe rings as SMARTS patterns instead of SMILES strings; SMARTS patterns allow us to add more information in the description of each atom. We created all the SMARTS patterns for the mono-rings and 2-fused-rings and validated them (for example, checking that a search for cyclopentane yielded no results that were actually cyclopentanone rings). Larger fragments (with three or more fused-rings) are not currently included. During the development of the program, we used only 21 mono-rings and six of the 2-fused-rings (in addition to the 24 non-rings fragments). These fragments were selected by prevalence in the drug library, and we kept those found the most often. This corresponds to a total of 113 fragments in the working library: indeed, since we use connection statistics between fragments (vide infra), a fragment such as pyridine must be described three times in the final library (one for each type of hydrogen). The 21 mono-rings and six 2-fused-rings which were used are displayed in Figure 3 and Figure 4. Expectedly, the most frequently occurring mono-ring and 2-fused-rings fragments are the same as the one found by Aldehi et al. and Taylor et al. (although not in the same order). Amongst the 2-fused-rings, one can find purine, xanthine and a β-lactam ring constituting the core of cephalosporin antibiotics (closely related to penicillin). Should a user decide not to use the β-lactam ring (e.g.) because the target protein is not related to antibiotics (whereas this fragment is mainly used in antibiotics), it can easily be discarded by not selecting it with the GUI. In addition to the fragments presented in Figure 3 and Figure 4, we have prepared all mono-rings and 2-fused rings fragment found more than 10 times in the drug library. This corresponds to a total of 58 mono-rings and 41 2-fused rings (413 different fragments in total). The full list of currently available fragments is displayed in the user manual. Moreover, the fragment preparation method is explained in the manual, allowing users to add new fragments according to their needs (3-fused-rings for example). The three-dimensional structures of each fragment were prepared by optimizing them with the Gaussian09 software (Rev. A.02) at the M06-2X/aug-cc-pVTZ
level of theory\textsuperscript{38} with an implicit description of water (IEF-PCM as implemented in Gaussian09)\textsuperscript{39}. When it was relevant, we started from different conformations and kept the one with the lowest energy.

Figure 3. List of mono-ring fragments used. The number of instances of each ring in the drug library is shown below the fragment.

Figure 4. List of 2-fused-ring fragments used. The number of instances of each ring in the drug library is shown below the fragment.

**Connecting fragments**

One of the lessons learned from SMoG was that fragments should not be randomly connected. When no rules are used for selecting fragments, the output molecules can be unstable, toxic, non-soluble and are also very likely difficult to synthesize. In 2009, our lab proposed to use chemoinformatic tools to analyze databases and to create *de novo* molecules occupying a given chemical space\textsuperscript{20}, resulting
in the so-called FOG algorithm. This approach has been used and improved here to fit the OpenGrowth needs. Briefly, we counted the connection between each pair of fragments in the drug database from ChEMBL. The probability that fragment $i$ is connected to fragment $j$ is then defined as:

$$P_{i\rightarrow j} = \frac{N_{i,j}}{\sum_f N_{i,f}}$$

where $N_{i,j}$ is the number of connections between fragments $i$ and $j$ and the sum in the denominator is made on all the fragments (more details are given in Supporting Information). In the original FOG algorithm, corrections had to be applied to the counting (see SI of [20]). However, the use of complete and more precise SMARTS patterns allows us to overcome this constraint. For example, amides are now defined as $[C;\!R;X3;!S(C(=O))(N)(N)]=-[O;X1]-[N;\!R;X3;!S(N(C(=O))C(=O))];!S(N=C);!S(N(S(O)(=O))(=O))C(=O)]$ (instead of $[C&\!R&X3](=O)-[N&\!R&X3]$ in FOG). In this new pattern, it is explicitly mentioned that the carbon is not part of a urea moiety $(C(=O))(N)(N)$, and that the nitrogen is not part of a $N(C(=O))C(=O)$ moiety, an imine $(N=C)$, nor a $N(S(O)(=O))C(=O)$ moiety. Since this amide fragment cannot be part of a urea group, it is necessary to define urea as a new fragment, explaining why some fragments in Figure 2 (last line) are trimers of other fragments. We can then create an $N\times N$ matrix with all the connections probabilities (where $N$ is the number of strictly different fragments, $N=113$ for the selected fragments in this manuscript). Analysis of this matrix teaches us, for example, that a phenyl has a 29.3% chance of being connected to a methyl group, whereas a methyl has only a 6.4% chance of being connected to a phenyl ring. During the growth, new fragments are chosen according to these probabilities. Consequently, the produced molecules will statistically “look like” molecules from the initial drug database. We will show later that the synthetic accessibility (as predicted by Sylvia) and the “drug-like” properties (as predicted by QikProp) are subsequently superior to the ones from molecules randomly grown.
Searching for 3mers

To further increase the synthetic accessibility of new molecules, we implemented a 3mer-screen algorithm that was developed at the same time as FOG\textsuperscript{20}. As we show further, this 3mer-screen also increases the “drug-like” properties of the new ligands (as predicted by QikProp). The algorithm is implemented as a sorting stage that is applied to grown ligands. The FOG growth is a two-state Markov chain that uses no information about connections to other fragments. Therefore, it is possible to connect three fragments together that are unlikely to be observed in that connected form in drugs, either because of synthetic difficulties, stability, or other reasons. For example, a $-\text{CH}_2-\text{CH}_2-\text{Br}$ moiety is never found in our drug library. The 3mer-screen consists of first creating all the possible 3mers from the fragment library; the drug library is then screened for those 3mers, and we retain in a list the ones that are never found. The user can then manually add 3mers, such as acetals or ketals, which are known to be unstable. The result is a list of “forbidden 3mers”, and when a newly grown molecule contains one of these 3mers the program warns the user about it. The principle of this screen is presented in Figure 5. A significant improvement has been made to the original algorithm: initially, rings were treated in a very general way with no specifications (SMARTS pattern [*&R]). We envisioned that this could lead to saving molecules that should have been discarded, especially concerning the connections around aromatic fragments. For example, it may be possible to find two phenyl substituents A and B in ortho to each other in drugs, but never in meta. We have thus used our SMARTS patterns to more accurately define 3mers involving rings, and the two patterns defining the case where the substituents are in ortho or in meta will now be seen as different cases. As a direct consequence, the list of forbidden 3mers is much longer; however, analyzing one new molecule against ~420,000 forbidden 3mers in an AMD Opteron 6376 CPU (2.3GHz) takes only ~20 seconds. Thus, searching for forbidden 3mers in the longer list does not significantly decrease the speed of the program.
Choosing fragments (GUI)

To facilitate the selection of fragments and the preparation of files needed by OpenGrowth (connection probabilities matrix, 3mer list, 3D structure files of the fragments...), we have developed a graphical user interface (GUI, see Figure 6). This interface is a program independent from OpenGrowth, and works under any version of Windows™ or Linux, using the Qt framework. It works in different stages: first, the user selects the “resource” and “output” directories and the database against which the probabilities will be calculated. We provide in the downloadable package our drug database (from ChEMBL) and the resource for the fragments (see user manual). Then, the user “upgrades” the database to count the occurrences of each fragment that are then displayed, as in Figure 6. Each fragment is a clickable button, allowing the user to select which one to keep; the “Fragment Files” button will create the list of the retained fragments. By default, non-ring fragments are always included and thus are not displayed on the GUI selection panel. If one wants to remove or add some of these fragments, he/she can manually alter the fragment list produced by the GUI. The “Probabilities” button reads the fragment list and creates the probability matrix, the “3-mer List” button creates the list of all the possible 3mer
made by the fragment list and the “3-mers” button looks for the forbidden 3mers. All computational stages can be performed by the interface on a local machine or run on external servers.

![OpenGrowth GUI](image)

**Figure 6.** Graphical User Interface developed to select fragments, calculate connection probabilities, prepare the 3mer list and the 3D structure files.

**Taking into account the protein flexibility**

Using a single, rigid conformation of a protein is known to be a source of errors, leading to false-positive and false-negative results\(^4^0\). To overcome this in the realm of docking, three approaches are mainly used\(^4^0^\text{a}, \text{b}, 4^1\): (i) decreasing the van der Waals radii of atoms to give more space to the ligands (“soft-docking")\(^4^2\); (ii) sampling side chains of the protein residues, or removing them prior to adding them back after the ligand has been docked\(^4^3\); (iii) docking the ligands in different conformations\(^4^0^\text{a}, 4^4\). We used an approach similar to that of the latter: in OpenGrowth, ligands can be grown simultaneously in several conformations of the protein. Since the program only needs .pdb files, the different conformations of the target protein can come from crystallographic structures of the same protein, structures from NMR experiments, or snapshots from molecular dynamics simulations. Moreover, all the input structures do
not necessarily have to come from the same protein — the program can be used with different protein mutants, with the same protein from different bacteria or with strictly different targets (e.g.). At every stage of the growth, ligands orientations and geometries are optimized in each conformation independently (see Supporting Information). Therefore, to each protein conformation corresponds a specific ligand position, optimal for its own protein structure (see Figure 7). The score used for the Metropolis criterion is then either the lowest score in all structures, an arithmetic average or a Boltzmann average (according to the user’s choice):

$$S_{\text{average}} = \frac{\sum_i S_i \ast e^{-\beta S_i}}{\sum_i e^{-\beta S_i}}, \quad S_i = \text{Score in conformation } i$$

This latter approach shares similarities with previous work, such as those proposed by Fisher et al. where a Boltzmann average is used to take into account penalties from crystallographic occupancies or from Österberg et al. where Boltzmann averages are used to construct a grid.

A modern view on protein flexibility consists of assuming that the protein “pre-exists in a number of energetically similar conformations.” When a ligand binds to a protein, it can bind to all these conformations. The most populated complex is not necessarily the one with the lowest binding free energy or the one from which the apo protein has the lowest folding free energy. The relevant energetic criterion is the sum of these two terms (plus the energetic deformation of the ligand, which is currently neglected). If we consider that the set of pre-existing conformations have a similar energy, then the equilibrium between all the complexes relies only on the binding score, which justifies the use of a Boltzmann average. Hence, this is an initial way to take induced fit into account, which supposes that the protein conformation will adapt around the ligand to lower the total Gibbs free energy. In our description, when the ligand is in a favorable position a higher weight is given to its score. Therefore, this specific protein conformation is considered to be adapted around the ligand as in the induced fit.
Scoring the protein/ligand binding energy

The current version of the program uses the same scoring function for both the rotameric search and the evaluation of the binding score for the Metropolis criterion. Two options are currently available: the SMOG2001 scoring function\(^{49}\) and a more recent one called SMOG2015. More details on these functions are given in the Supporting Information. Since SMOG2015 is still in development, most of the results presented below were obtained with the SMOG2001 function (unless otherwise noted). Other scoring functions will be included in our program in future releases. However, since OpenGrowth is a modular platform and an open-source project, external contributors could easily implement their own scoring functions in the program (how to do so is explained in the user manual).

Computing accurate protein/ligand binding energy

Once OpenGrowth has generated in the order of 100,000 new ligands, it is necessary to screen these ligands with methods that are more accurate than scoring functions. Our test protein has been the HIV-1 protease, and we have used an approach proposed by Wright et al. to reproduce experimental binding Gibbs free energies\(^{50}\). We performed 50 independent molecular dynamics simulations of 5
nanoseconds for each protein-ligand complex, differing only by the initial velocities. Simulations were performed on GPU cards with Gromacs v5.0.2, the Amber03 force field and the TIP3P water model. Proteins were pre-processed by the Protein Preparation Wizard tool of Maestro (to add hydrogens and missing atoms and to minimize the energy). The correct protonation state of ligands was determined using Chimera 1.9; they were then optimized at the M06-2X/6-31+G** level of theory with an implicit description of water (IEF-PCM) using Gaussian09. Finally, RESP charges were computed at the HF/6-31G* level of theory with Gaussian09, and acpype (an interface to antechamber) was used to generate the ligand topology using the General Amber Force Field. The system was then solvated in a rhombic dodecahedron box with a distance of at least 11 Å between the complex and the box edges. A 10 Å cut-off was used for van der Waals interactions, whereas electrostatic ones were treated with PME with a similar “cut-off” of 10 Å. The energy of the solvated system was minimized until the maximum force reached 5.0 kJ.mol$^{-1}$.nm$^{-1}$, first with steepest descent and then with conjugate gradient. A primary NVT simulation of 200ps (dt=1fs, velocity-rescale thermostat, T=100K) was performed with the protein atoms being fixed (allowing the solvent to equilibrate). Then, another NVT simulation (without restraints) was done with simulated annealing, the temperature going from 100K to 300K linearly in 400ps (dt=1fs, velocity-rescale thermostat); the system then evolved freely 100ps at 300K. A 500ps NPT simulation was subsequently carried out to equilibrate the pressure (dt=1fs, Nosé-Hoover thermostat, Berendsen barostat). Finally, the production simulation was done (dt=2fs, Nosé-Hoover thermostat, Parinello-Rahman barostat). Bond lengths were constrained during the simulations using the LINCS algorithm.

In the last nanosecond of each production simulation, 100 snapshots were extracted (every 10ps). The Amber12 package was then used to compute the Molecular Mechanics/Poisson–Boltzmann Surface Area binding energy (MMPBSA) for the 5000 snapshots and the entropy contribution for 1000 snapshots (one every 50ps), using a single-trajectory approximation; the results are then averaged. The ionic strength was fixed at 0.0M, whereas the surface tension and offset of the cavity of the Poisson-
Boltzmann calculations were set to 0.0072 kcal/mol/Å² and 0.00 kcal/mol respectively\(^{65-66}\) (referred as Set1 below). We also performed calculations with surface tension and offset of the cavity set to 0.00542 kcal/mol/Å² and 0.92 kcal/mol respectively for comparisons (referred as Set2 below). The nonpolar optimization method of Amber12 (“inp” keyword) was set to 1; the use of the default value of 2 did not allow us to reproduce experimental results. For each complex, we verified that the distribution of the MMPBSA energies followed a normal distribution, ensuring that the 5000 snapshots represented a system at thermodynamic equilibrium. Each simulation took on average 6 hours to be completed (equilibration + production) whereas MMPBSA and entropy contributions took on average 8 hours on modern clusters. Note that since Gromacs and AmberTools are open-source softwares, all the simulations and analysis can be performed free of charge. The scripts and input files used for these simulations are available for download on the authors’ websites and on the OpenGrowth webpage.

To validate this method, we applied it to a set of complexes with known crystallographic structures. Similarly to the work of Wright et al., we selected the 9 FDA-approved HIV-1 protease inhibitors\(^{50}\). To choose the complexes, we targeted sequences of the protein as close as possible to the consensus wild-type HXB2\(^{67}\) (same sequence as the PDB ID 1HXB). The list of retained structures is shown in Table 1. For some complexes, two orientations of the ligand (with occupancy smaller than 1.0) are given in the pdb structure. In such cases, simulations were started from both orientations and we kept the one with the lowest energetic values. The active site of the HIV-1 protease is made of two aspartate residues (ASP25) and there is a debate surrounding the protonation state of these amino-acids since four states are possible: the two residues from each chain can be both non-protonated, both protonated, or one or the other can be protonated. Recent studies have shown that the latter possibility is the most likely (see \(^{68}\) and references cited therein). Thus, for each complex we have simulated the two possibilities (ASP25 of chain A protonated and not ASP25 of chain B, or vice versa). This set represents simulations of 15 complexes with two protonation states, corresponding to 1500 simulations of 5ns.
<table>
<thead>
<tr>
<th>Ligand</th>
<th>Abbreviation</th>
<th>Mutations</th>
<th>PDB ID</th>
<th>$K_d$ (nM)</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amprenavir</td>
<td>APV</td>
<td>V3I</td>
<td>1HPV</td>
<td>0.6</td>
<td>69</td>
</tr>
<tr>
<td>Atazanavir</td>
<td>AZV</td>
<td>V3I</td>
<td>2AQU</td>
<td>0.48</td>
<td>70</td>
</tr>
<tr>
<td>Darunavir</td>
<td>DRV</td>
<td>V3I</td>
<td>4HLA</td>
<td>0.016</td>
<td>71</td>
</tr>
<tr>
<td>Indinavir</td>
<td>IDV</td>
<td>V3I</td>
<td>1HSG</td>
<td>2.45</td>
<td>72</td>
</tr>
<tr>
<td>Lopinavir</td>
<td>LPV</td>
<td>V3I/I47A</td>
<td>2QHC</td>
<td>1.1</td>
<td>73</td>
</tr>
<tr>
<td>Nelfinavir</td>
<td>NFV</td>
<td>V3I</td>
<td>1OHR</td>
<td>2.0</td>
<td>74</td>
</tr>
<tr>
<td>Ritonavir</td>
<td>RTV</td>
<td>V3I/S37N</td>
<td>1HXW</td>
<td>0.015</td>
<td>75</td>
</tr>
<tr>
<td>Saquinavir</td>
<td>SQV</td>
<td>Wild-Type</td>
<td>1HXB</td>
<td>0.12</td>
<td>76</td>
</tr>
<tr>
<td>Tipranavir</td>
<td>TPV</td>
<td>V3I/Q7K</td>
<td>2O4P</td>
<td>0.019</td>
<td>77</td>
</tr>
</tbody>
</table>

Table 1. List of retained HIV-1 protease-inhibitor complexes to validate the computational estimation of the binding Gibbs free energy.

The correlation coefficients (R) between the computational and experimental binding Gibbs free energies for different methods are presented in Table 2. We observed that for tight binding ligands (with $K_d$ in the low picomolar (pM) range), the MMPBSA approach could fail to provide realistic estimations of the binding Gibbs free energy. This is the case when lopinavir is simulated with the complex from the PDB ID 1MUI (data not shown here) for which $K_d=0.0013\text{nM}^{78}$; consequently, lopinavir was simulated with the complex from the PDB ID 2QHC instead ($K_d=1.1\text{nM}^{73}$). Similarly, we observed that by removing darunavir (DRV) from the set of ligands used to benchmark our method ($K_d=0.016\text{nM}$ with PDB ID 4HLA), the correlation was significantly increased. When Set2 of parameters are used for the MMPBSA calculations, the correlations between computations and experiments are slightly lower than with Set1 (see Table 2). The best correlation is obtained with Set1 of parameters, when entropy estimations are used, and when the lowest binding Gibbs free energy between the two protonation states is used (last column of the third line of Table 2, $R=0.852$, $R^2=0.726$). For a majority of ligands (five of them), the lowest energy was coming from a structure where ASP25 from chain B is protonated (noted (3) in Table 2). Thus, it was the protonation state used for the protein when implemented in OpenGrowth in the future sections.
Table 2. Correlation coefficient (R) obtained with different methods. (2) means that ASP25 of chain A is protonated and not ASP25 from chain B. (3) means the opposite: ASP25 of chain B is protonated and not ASP25 of chain A. (Best) means that the lowest binding free energy is kept, coming either from (2) or from (3). Set1 refers to values of 0.0072 kcal/mol/Å² and 0.00 kcal/mol for the surface tension and offset of the cavity respectively. Set2 refers to values of 0.00542 kcal/mol/Å² and 0.92 kcal/mol.

The correlation between the computational (with Set1) and experimental binding Gibbs free energies for the eight kept FDA-approved HIV-1 protease inhibitors is presented in Figure 8. In the following, the “predicted dissociation constant” refers to the dissociation constant ($K_d = \exp(\Delta G_{bind}/RT)$) obtained using the correlation between computation and experiments, as shown in Figure 8.
Figure 8. Correlation between computation and experiments for the binding Gibbs free energy for eight FDA-approved HIV-1 protease inhibitors.

Technical details

OpenGrowth is written in C++ (~6500 lines of code), and uses the OpenBabel library extensively. It can typically grow ~100 ligands/day/core in an AMD Opteron 6376 CPU (2.3GHz) and ~170 ligands/day/core in an Intel Xeon E5-2640 v2 CPU (2.0GHz). The most computationally demanding part of the program is the geometry optimization of the ligands (see Supporting Information), and by tweaking the parameters (for example, performing less minimization steps) the speed can be increased. OpenGrowth has been prepared in a modular manner, and each function can be updated to fit the user needs. We release it as an open-source project under the GNU GPL license. It is freely downloadable from the author websites and from http://opengrowth.sourceforge.net. OpenGrowth is easy to use (“./OpenGrowth.exe Input.inp”) and uses a keyword-oriented input file. None of the parameters are hard-coded to allow the user to have more flexibility, and default values are proposed to simplify the process for new users. More details and explanations are provided in the Supporting Information of the current manuscript and on the program manual downloadable from the websites.

Results

Ability of the growth algorithm to reproduce transition probabilities

We wanted to verify that our program can grow new molecules with properties as desired, that is: connection probabilities between fragments, synthetic accessibility and pharmacokinetic properties, binding energies. First, in order to assess the ability of OpenGrowth to reproduce the transition probabilities from the training library, we developed a new program called FOG2.0. This program has the same goals as the original FOG, i.e. creating libraries occupying a given chemical space by using connection probabilities (without considering any protein). However, FOG2.0 uses the same structure
and input files as OpenGrowth; in particular, it creates 3D structures whereas FOG produced molecules as SMILES strings. FOG2.0 is released as an open-source project (under the GNU GPL license) jointly with OpenGrowth. We grew two sets of molecules with FOG2.0: one using the connection probabilities to select new fragments (as in FOG), one with randomly connecting fragments. We then compared the connection probabilities obtained from the grown libraries (each one having 425,000 different molecules) and from the drug database. It appears clearly in Figure 9-(a) that our algorithm produces molecules with statistics similar to the training database (R=0.767, slope=0.82). When connections are made randomly (Figure 9-(b)), the statistics are not correlated (R=0.222, slope=0.04). Since FOG2.0 creates molecules without considering a protein, the same analysis for two sets of ~73,000 ligands produced by OpenGrowth in the active site of the HIV-1 protease is presented in Figure 9-(c) and Figure 9-(d). Apparently, growing in the active site of the protein keeps the connection probabilities similar to those created from the training database.

Three outliers can be observed in Figure 9-(b) and Figure 9-(d), at coordinates (0.00;0.47), (0.09;0.23) and (0.71;0.36) for Figure 9-(b) (purple points); these points correspond respectively to the transitions H₂O->NH₃, CH₄->NH₃ and NH₃->CH₄. The reason for these outliers hinges on how the growth is performed: once a fragment has been chosen and added to the current state of the ligand, the program searches for intramolecular steric clashes (see Supporting Information). When clashes are detected, the fragment is always rejected and a new fragment is chosen for addition at the same position. If, after a given number of tests (e.g. 50 times), no fragments have been successfully added, a new position/hydrogen is chosen in the current ligand. Many bulky fragments will therefore be rejected during this stage, and small ones (NH₃, CH₄, H₂O) are more likely to be accepted because they are less prone to steric clashes. This search for steric clash therefore explains the presence of a few outliers in Figure 9-(b) and Figure 9-(d) and the non-perfect correlation in Figure 9-(a) and Figure 9-(c).
(a) FOG probabilities (with FOG2.0).

(b) Random connections (with FOG2.0).
Figure 9. Probabilities of connections from *de novo* molecules (grown with FOG2.0 or with OpenGrowth) versus the same probabilities in the drug database. Each point represents one probability, for example the probability that a phenyl ring is connected to an amine. We have underlined in purple the outliers discussed in the text.
Properties of output molecules

To evaluate the properties of the new molecules, different sets of ligands were grown in one conformation of the HIV-1 protease (coming from PDB ID 2AQU). The first set made with random connections between fragments; the second set made of molecules grown with FOG probabilities which have failed the 3mer screen; the third set also made with FOG probabilities but with molecules that passed the 3mer screen. During the growth, the program may generate several times the same molecule: ~10% of the molecules are redundant with some others when random connections are used, whereas only ~3% are redundant with others with FOG probabilities. After removing these redundant molecules, the three sets have sizes of 117,647, 53,759 and 21,754 molecules respectively (29% of all the molecules grown with FOG probabilities passed the 3mer screen). We compared the properties of these three sets with two drug libraries: the one that we have presented before (named ChEMBL drugs) and one made by Hutter with 2665 non-redundant molecules which “were chosen in a way that reflects the proportional distribution of all agents among the therapeutic categories”80.

We first investigated the synthetic accessibility of the molecules by using the Sylvia program from Molecular Networks GmbH81. Sylvia returns a number between 1 and 10 after analyzing each molecule in terms of properties such as number of rings, number of stereocenters, similarities to a starting material database or molecular graph. The lower the score, the easier the synthesis of the molecule should be. We report Figure 10-(a) the distribution of Sylvia scores for the five sets of molecules. Molecules that were grown randomly (in yellow pentagones) have a high Sylvia score with a peak around 7.5. When FOG probabilities are used but the 3mer test failed (in orange circles), the distribution is shifted toward lower values of the score (peak at 6.5); passing the 3mer screen further decreases the score (in red triangles, peak at 5.5). For the two set of authentic molecules (in blue diamond and green squares), the peak is centered at 4.0. Thus, we validate here our strategy (connection probabilities and 3mer screen) to increase the synthetic accessibility of de novo ligands.
While this result was previously demonstrated for molecules grown in 2D with FOG\textsuperscript{20}, we show here that we can access compounds with similar properties grown in the steric and energetic constraints of a binding pocket.

We then focused on the pharmacokinetic properties. Predicting ADME (Absorption, Distribution, Metabolization, Excretion) is a formidable challenge for computational drug design\textsuperscript{3, 82}; the approach used by QikProp (Schrödinger, LLC) is to compute 43 descriptors or properties such as number of hydrogen bond donor atoms, polarizability, octanol/water partition coefficient or electron affinity\textsuperscript{83} for each molecule. One must note that some key predictors of pharmacokinetic properties, such as metabolic liability and intrinsic clearance values, are not calculated by QikProp. Thus, even if QikProp cannot be used to “predict pharmacokinetic properties” (strictly speaking), it is a useful tool to identify “drug-like” molecules. For each property, 95% of drugs (from a pre-defined list) fall in a given range. Any time a computed property for a new molecule falls out of this range, one “star” is added. The number of computed stars for each molecule is then used as an estimation of its “drug-like” properties (the lower, the better). We present in Figure 10-(b) the analysis for the five sets of molecules. New ligands grown with FOG probabilities and which passed the 3mer screen (in red) are more likely to have a low number of stars than random molecules or molecules that did not pass the 3mer screen (yellow and orange). As expected, the two sets of drugs also have a higher probability of being given a low number of stars (blue and green). For instance, 66% of the ligands grown with FOG and which passed the 3mer screen have 0 or 1 stars, whereas this value falls to 53% for FOG molecules that failed the 3mer screen and 44% for molecules that were randomly grown. For the Hutter and ChEMBL databases, there are 83% and 79% molecules that have 0 or 1 stars. For the randomly grown library, 10% of the molecules have 6 stars or more, which falls to 7% for FOG molecules that failed the 3mer screen, 3% for FOG molecules that passed the 3mer screen and less than 2% for the two drug databases. So here again, the selection of molecules in the context of a binding pocket does not preclude the generation of drug-like compounds,
and the implementation of connection probabilities and 3mer screen increases the chance to make new ligands with good ADME properties. It must be pointed out that even if the size of the five sets are not the same, we have checked that the same distribution (for Sylvia and for QikProp) is obtained when we randomly down-sample the libraries.

Figure 10. Properties of the output molecules.
We then analyzed the diversity of the grown ligands. Of the 21,754 molecules created with FOG probabilities that passed the 3mer screen, 3% were redundant with others in the library. We then computed the Tanimoto score between all pairs of the remaining 21,138 molecules (corresponding to more than 223 million of scores). Tanimoto score is a measure of similarity between two molecules: it is a number between 0 (very different) and 1 (very similar). The distribution of the observed scores is presented in Figure 11. The peak of the distribution is around 0.2, which signifies highly different molecules. Thus, even if the growth is biased towards a given chemical space, the OpenGrowth algorithm grows molecules that occupy a large part of this chemical space.

![Figure 11. Distribution of the Tanimoto score between all pairs of 21,138 molecules. A score of 1.0 between two molecules means that they are identical.](image)

**Predicting binding energy of de novo ligands**

To evaluate if our program can grow molecules that are predicted to be good binders, we took the ~75,000 ligands grown with FOG connection probabilities that were used in the previous section. They were grown in one conformation of the HIV-1 protease (from PDB ID 2AQU) using a threshold of 20
fragments and the SMOG2001 scoring function. Since we wanted to focus here on the binding free energy, we also considered molecules that failed the 3mer-screen. Scores ranged from -90 to 0 in SMOG2001 internal units which correlate to binding free energy (for comparison, scores for the eight authentic HIV-1 protease inhibitors ranged from -44 to -24 SMOG2001 units). We picked 10 of the best de novo ligands and applied the MD protocol defined previously (in section “Computing accurate protein/ligand binding energy”). We display in Table 3 some of the predicted properties for these molecules and the highest Tanimoto score between each ligand and the full ZINC library (which is made of ~19 million purchasable compounds as updated on November 28th 2014). The structures of the nearest neighbors are presented in Supporting Information: some similarities were observed after visual inspections, however no significant similarities could be found. The most similar molecule came from the analysis of Ligand014_18 for which ZINC67956366 was the closest. The four molecules with sub-micromolar predicted $K_d$ are displayed in Figure 12.

<table>
<thead>
<tr>
<th>Ligand</th>
<th>SMOG2001</th>
<th>MW</th>
<th>3mer</th>
<th>Predicted $K_d$ (nM)</th>
<th>Sylvia</th>
<th>QikProp</th>
<th>Tanimoto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ligand007_14</td>
<td>-84.50</td>
<td>520</td>
<td>Pass</td>
<td>1,000</td>
<td>5.66</td>
<td>1</td>
<td>0.67</td>
</tr>
<tr>
<td>Ligand041_521</td>
<td>-83.89</td>
<td>430</td>
<td>Fail</td>
<td>363</td>
<td>5.86</td>
<td>4</td>
<td>0.70</td>
</tr>
<tr>
<td>Ligand105_64</td>
<td>-83.35</td>
<td>392</td>
<td>Fail</td>
<td>1,630</td>
<td>5.27</td>
<td>1</td>
<td>0.61</td>
</tr>
<tr>
<td>Ligand015_407</td>
<td>-83.01</td>
<td>445</td>
<td>Fail</td>
<td>24,000</td>
<td>7.14</td>
<td>1</td>
<td>0.91</td>
</tr>
<tr>
<td>Ligand057_1140</td>
<td>-81.46</td>
<td>557</td>
<td>Fail</td>
<td>98,900</td>
<td>7.13</td>
<td>1</td>
<td>0.50</td>
</tr>
<tr>
<td>Ligand014_520</td>
<td>-80.21</td>
<td>508</td>
<td>Pass</td>
<td>11,200</td>
<td>6.44</td>
<td>0</td>
<td>0.67</td>
</tr>
<tr>
<td>Ligand007_351</td>
<td>-80.01</td>
<td>550</td>
<td>Pass</td>
<td>110,000</td>
<td>6.62</td>
<td>0</td>
<td>0.56</td>
</tr>
<tr>
<td>Ligand060_479</td>
<td>-79.79</td>
<td>575</td>
<td>Fail</td>
<td>227</td>
<td>7.11</td>
<td>3</td>
<td>0.51</td>
</tr>
<tr>
<td>Ligand014_279</td>
<td>-79.55</td>
<td>522</td>
<td>Pass</td>
<td>18.3</td>
<td>5.42</td>
<td>1</td>
<td>0.73</td>
</tr>
<tr>
<td>Ligand014_18</td>
<td>-79.12</td>
<td>532</td>
<td>Fail</td>
<td>82.1</td>
<td>5.93</td>
<td>1</td>
<td>0.69</td>
</tr>
</tbody>
</table>

Table 3. Ten of the best de novo ligands (according to SMoG2001) grown. The SMOG2001 score is in internal units. MW is the molecular weight in g/mol. The 3mer column displays the success or failure of the 3mer screen. “Predicted $K_d$” is the predicted dissociation constant calculated using Figure 8. The Sylvia score is in internal units and ranges between 1 and 10. QikProp displays the number of “stars” per
molecule. The Tanimoto column shows the highest score for the comparison between each molecule and the full ZINC library.

![Molecule Images]

Figure 12. Four predicted sub-micromolar ligands to HIV-1 protease.

Therefore, in less than a week we were able to generate new molecules that are predicted to bind to the HIV-1 protease in the nanomolar range. These molecules might constitute interesting starting points for finding a first hit, especially Ligand014_279 and Ligand014_18 due to their low Sylvia scores for synthetic accessibility. We noted, however, that some of the top scored ligands (according to SMOG2001) are in fact predicted to be poor binders by MD simulations, with predicted \( K_d \) above 100\( \mu \)M. This should certainly be attributed to the limitations of predicting power of SMOG2001, as will be discussed later.

**Seeding a fragment**

Most of drug discovery projects start with the identification of a hit that is later optimized to a lead. This is especially true in fragment-based drug discovery (which has emerged as a promising alternative to high-throughput screening) where low-molecular weight fragments that binds the target protein are identified and grown to larger molecules. A growth with OpenGrowth can start with a seed to mimic this kind of strategy. Thus, OpenGrowth can provide a support to fragment-based drug discovery projects by generating new ideas on how to evolve a fragment into a small molecule. During the
development of atazanavir (sold under the name Reyataz by Bristol-Myers Squibb), a first scaffold was identified (see Figure 13)\(^4\). Subsequent variations of this aza-dipeptide moiety and lead optimization led to the approval of the drug by the FDA in 2003\(^5\). We grew ~170,000 ligands using the seed displayed in Figure 13. Growth was only allowed from the marked atoms that were identified in the input file as hydrogen atoms from an amide or a methyl, respectively. Thus, we utilized FOG probabilities to grow molecules. Ten of the best ligands are listed in Table 4 with some of the predicted properties and the highest Tanimoto scores with respect to the full ZINC library. Eight of them have sub-micromolar predicted $K_d$ and could have been served as starting point for the optimization to provide new hits. Four of them are displayed in Figure 14. We note that starting from a seed allow obtaining more homogeneous predicted $K_d$: eight out of the 10 bests are in the hundreds of nanomolar range, and the two remainings are in the low micromolar range. When no seed was used (Table 3), predicted $K_d$ are very diverse and range from 18nM to 110$\mu$M.

![Core scaffold / Seed](image1.png)

![Atazanavir](image2.png)

**Figure 13.** Atazanavir and the seed used for growth.

<table>
<thead>
<tr>
<th>Ligand</th>
<th>SMOG2001</th>
<th>MW</th>
<th>3mer</th>
<th>Predicted $K_d$ (nM)</th>
<th>Sylvia</th>
<th>QikProp</th>
<th>Tanimoto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seed068_232</td>
<td>-94.08</td>
<td>586</td>
<td>Fail</td>
<td>223</td>
<td>6.60</td>
<td>3</td>
<td>0.65</td>
</tr>
<tr>
<td>Seed072_1149</td>
<td>-90.68</td>
<td>612</td>
<td>Fail</td>
<td>3,870</td>
<td>7.49</td>
<td>3</td>
<td>0.58</td>
</tr>
<tr>
<td>Seed042_1142</td>
<td>-88.23</td>
<td>614</td>
<td>Pass</td>
<td>118</td>
<td>6.79</td>
<td>3</td>
<td>0.60</td>
</tr>
<tr>
<td>Seed108_652</td>
<td>-87.15</td>
<td>552</td>
<td>Pass</td>
<td>184</td>
<td>6.38</td>
<td>3</td>
<td>0.58</td>
</tr>
<tr>
<td>Seed079_39</td>
<td>-86.68</td>
<td>598</td>
<td>Pass</td>
<td>607</td>
<td>6.14</td>
<td>2</td>
<td>0.57</td>
</tr>
<tr>
<td>Seed040_105</td>
<td>-86.63</td>
<td>597</td>
<td>Fail</td>
<td>311</td>
<td>7.27</td>
<td>2</td>
<td>0.67</td>
</tr>
<tr>
<td>Seed001_274</td>
<td>-85.44</td>
<td>555</td>
<td>Pass</td>
<td>3,220</td>
<td>6.45</td>
<td>2</td>
<td>0.60</td>
</tr>
</tbody>
</table>
Table 4. Ten of the best de novo ligands (according to SMOG2001) grown around a seed. The SMOG2001 score is in internal units. MW is the molecular weight in g/mol. The 3mer column displays the success or failure of the 3mer screen. “Predicted $K_d$” is the predicted dissociation constant calculated using Figure 8. The Sylvia score is in internal units and ranges between 1 and 10. QikProp displays the number of “stars” per molecule. The Tanimoto column shows the highest score for the comparison between each molecule and the full ZINC library.

<table>
<thead>
<tr>
<th>Seed</th>
<th>Score</th>
<th>MW</th>
<th>3mer</th>
<th>$K_d$</th>
<th>Sylvia</th>
<th>QikProp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seed004_360</td>
<td>-85.04</td>
<td>555</td>
<td>Fail</td>
<td>102</td>
<td>6.55</td>
<td>1</td>
</tr>
<tr>
<td>Seed033_1099</td>
<td>-85.02</td>
<td>551</td>
<td>Pass</td>
<td>252</td>
<td>6.27</td>
<td>1</td>
</tr>
<tr>
<td>Seed082_1316</td>
<td>-85.01</td>
<td>625</td>
<td>Fail</td>
<td>484</td>
<td>7.40</td>
<td>4</td>
</tr>
</tbody>
</table>

Figure 14. Four predicted sub-micromolar ligands to HIV-1 protease grown around a seed.

We also compared each one of the 170,000 grown ligands with the atazanavir structure by computing the Tanimoto scores. The most similar molecule (Seed101_146) has a Tanimoto score with atazanavir of 0.80, a SMOG2001 score of -62.18 units and is ranked (according to SMOG2001 score) 28,713 in the list of all molecules. Its structure is displayed on Figure 15; we also show the superimposition of the positions of atazanavir and Seed101_146 in the active site. When comparing the
two molecules (see atazanavir in Figure 13), several common features appear: (1) the phenyl-pyridine moiety was grown by OpenGrowth at the same position than the original structure. This moiety (present at one of the extremity of the active site) develops hydrophobic interactions with Val82 and Pro81 from chain A and Ile50 from chain B. (2) On the other side of the molecule, an isopropyl moiety was grown in place of a tertbutyl in atazanavir, the two being very similar. (3) An amine was added in beta of the amide on the left whereas an amide is present in atazanavir, the two moieties sharing the nitrogen atom at the same place. Overall, Seed101_146 shares several interesting features with the FDA-approved drug. However, it must be pointed out that Seed101_146 is predicted as a weak binder, which suggests room for improving our scoring method. The use of more accurate scoring functions than SMOG2001 will remove this limitation and open many interesting applications for this program.

(a) Seed101_146.  
(b) Superimposition with the position of atazanavir. 

Figure 15. Most similar grown structure to atazanavir and superimposition of the structures in the active site of the protein. See Figure 14 for the structure of atazanavir.

Using the protein flexibility option

We first used the protein flexibility approach with 10 conformations of the HIV-1 protease and the SMOG2001 scoring function. The 10 conformations were obtained by performing 10 independent molecular dynamics simulations of the apo protein as explained previously (in section “Computing accurate protein/ligand binding energy”) and starting from PDB ID 2AQU. We clustered the last four
nanoseconds of each simulation and kept the central structure of each cluster (which is the conformation with the smallest average root-mean-square deviation to all the other structures in a given trajectory). We grew ~10,000 molecules with OpenGrowth and picked 10 of the best structures to compute the binding free energy more accurately (see Table 5). However, we found that the ligands grown with this method were smaller than the ones previously presented: the lowest SMOG2001 score is only of -50.20 units and the molecular weight is on average 220 g/mol (whereas Table 3 and Table 4 show ligands of ~540 g/mol). Consequently, the predicted \( K_d \) are higher (\( K_d = 638 \text{nM} \) for Flex073_23 or \( K_d = 1.81 \text{µM} \) for Flex071_6). The reason for this behavior is very likely the hard-wall potential used with the SMOG2001 scoring function that imposes many more constraints; we envisioned that a scoring function lacking this constraint would probably perform better for predicting good binders. We computed the Tanimoto score between each one of these ligands and the full ZINC library and the structures of the nearest neighbors are presented in Supporting Information: similarities can be found, but none of the grown ligands are identical to molecules contained in the ZINC library (even for pairs with a Tanimoto score of 1.00).

<table>
<thead>
<tr>
<th>Ligand (10 structures)</th>
<th>SMOG2001</th>
<th>MW</th>
<th>3mer</th>
<th>Predicted ( K_d ) (nM)</th>
<th>Sylvia</th>
<th>QikProp</th>
<th>Tanimoto</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flex073_23</td>
<td>-50.20</td>
<td>246</td>
<td>Fail</td>
<td>638</td>
<td>4.14</td>
<td>0</td>
<td>0.84</td>
</tr>
<tr>
<td>Flex095_24</td>
<td>-50.15</td>
<td>262</td>
<td>Fail</td>
<td>5,530</td>
<td>4.90</td>
<td>0</td>
<td>0.73</td>
</tr>
<tr>
<td>Flex097_66</td>
<td>-44.66</td>
<td>234</td>
<td>Pass</td>
<td>4,940</td>
<td>4.38</td>
<td>2</td>
<td>0.72</td>
</tr>
<tr>
<td>Flex112_6</td>
<td>-44.30</td>
<td>206</td>
<td>Pass</td>
<td>587,000</td>
<td>4.32</td>
<td>0</td>
<td>0.84</td>
</tr>
<tr>
<td>Flex120_64</td>
<td>-44.19</td>
<td>276</td>
<td>Pass</td>
<td>81,200,000</td>
<td>4.75</td>
<td>3</td>
<td>0.82</td>
</tr>
<tr>
<td>Flex097_42</td>
<td>-43.62</td>
<td>186</td>
<td>Pass</td>
<td>943,000</td>
<td>3.09</td>
<td>1</td>
<td>1.00</td>
</tr>
<tr>
<td>Flex063_14</td>
<td>-43.43</td>
<td>193</td>
<td>Fail</td>
<td>2,093,000</td>
<td>3.77</td>
<td>1</td>
<td>0.89</td>
</tr>
<tr>
<td>Flex088_16</td>
<td>-42.66</td>
<td>203</td>
<td>Pass</td>
<td>12,300</td>
<td>4.40</td>
<td>1</td>
<td>0.79</td>
</tr>
<tr>
<td>Flex071_6</td>
<td>-42.49</td>
<td>197</td>
<td>Pass</td>
<td>1,810</td>
<td>4.96</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>Flex100_7</td>
<td>-42.39</td>
<td>202</td>
<td>Pass</td>
<td>4,290</td>
<td>4.01</td>
<td>6</td>
<td>1.00</td>
</tr>
</tbody>
</table>

Table 5. Ten of the best *de novo* ligands (according to SMOG2001) grown in 10 structures of the protein.

The SMOG2001 score is in internal units. MW is the molecular weight in g/mol. The 3mer column displays the success or failure of the 3mer screen. “Predicted \( K_d \)” is the predicted dissociation constant
calculated using Figure 8. The Sylvia score is in internal units and ranges between 1 and 10. QikProp displays the number of “stars” per molecule. The Tanimoto column shows the highest score for the comparison between each molecule and the full ZINC library.

As stated previously, we are currently developing a new scoring function called SMOG2015. Besides using a more accurate training set, another kind of atom typing and three shells instead of two, this scoring function uses a repulsive term instead of a hard-wall potential for protein-ligand interaction. We present in Table 6 results of preliminary tests while growing compound in one and in five target structures. For both cases, we present the SMOG2015 score and the molecular weight of the 10 best ligands. It appears clearly that while using SMOG2015 with five structures, it is possible to grow ligands of the same size as with only one structures (on average, the molecular weight of the 10 best ligands is 598 g/mol when growing in one structure and 513 g/mol while growing in five structures). Thus, SMOG2015 can overcome the limitations of SMOG2001 while using the protein flexibility option.

<table>
<thead>
<tr>
<th>Ligand (1 structure)</th>
<th>SMOG2015</th>
<th>MW</th>
<th>Ligand (5 structures)</th>
<th>SMOG2015</th>
<th>MW</th>
</tr>
</thead>
<tbody>
<tr>
<td>Regular008_8</td>
<td>-618.84</td>
<td>633</td>
<td>RegularFlex105_36</td>
<td>-452.65</td>
<td>487</td>
</tr>
<tr>
<td>Regular041_254</td>
<td>-595.53</td>
<td>647</td>
<td>RegularFlex076_17</td>
<td>-446.33</td>
<td>501</td>
</tr>
<tr>
<td>Regular110_112</td>
<td>-590.16</td>
<td>569</td>
<td>RegularFlex101_52</td>
<td>-431.21</td>
<td>523</td>
</tr>
<tr>
<td>Regular048_147</td>
<td>-588.41</td>
<td>642</td>
<td>RegularFlex095_33</td>
<td>-430.81</td>
<td>482</td>
</tr>
<tr>
<td>Regular084_54</td>
<td>-587.17</td>
<td>590</td>
<td>RegularFlex027_29</td>
<td>-427.65</td>
<td>588</td>
</tr>
<tr>
<td>Regular007_1</td>
<td>-586.02</td>
<td>558</td>
<td>RegularFlex104_2</td>
<td>-427.52</td>
<td>460</td>
</tr>
<tr>
<td>Regular095_313</td>
<td>-582.59</td>
<td>562</td>
<td>RegularFlex091_44</td>
<td>-422.91</td>
<td>535</td>
</tr>
<tr>
<td>Regular087_277</td>
<td>-580.85</td>
<td>617</td>
<td>RegularFlex058_24</td>
<td>-420.48</td>
<td>551</td>
</tr>
<tr>
<td>Regular065_264</td>
<td>-580.18</td>
<td>600</td>
<td>RegularFlex119_18</td>
<td>-416.94</td>
<td>572</td>
</tr>
<tr>
<td>Regular109_98</td>
<td>-575.04</td>
<td>568</td>
<td>RegularFlex124_35</td>
<td>-412.81</td>
<td>433</td>
</tr>
</tbody>
</table>

Table 6. Preliminary results obtained when growing ligands with the SMOG2015 scoring function in one and in five target structures. The SMOG2015 score is in internal units. When 5 structures were used, we have used an arithmetic average of the scores. MW is the molecular weight in g/mol.
**Discussion and Conclusions**

We have presented here the foundations of a new lead design tool named OpenGrowth for use in the growth of ligands in the active site of a protein. Due to the open-source nature of this project, external contributors have the opportunity to improve the program. The efficiency of our algorithm (more than 150,000 new ligands can be grown in one week using 128 cores) allowed us to quickly identify new structures with a predicted $K_d$ with the HIV-1 protease in the nanomolar range. This makes OpenGrowth a very fast and efficient tool for drug discovery, allowing one to discover novel scaffolds for further optimization. Ligands grown by our program must be analyzed and refined before being tested experimentally, but several of the structures presented here need only small modifications to be synthesized for trial. For example, one could find that the carboxylic moiety in Ligand014_279 (see Figure 12) is unsuitable for HIV-1 protease, and thus decide to remove it. Similarly, Seed042_1142 and Seed004_360 (see Figure 14) could be optimized to be more stable *in vivo*. Importantly, the possibility of growing molecules from a seed in OpenGrowth can dramatically speed up the search of variations around a first hit shortening the time needed for the discovery of leads or candidates. As such, it is a perfect *in silico* tool to tackle the problem of fragment-based drug discovery. In addition, this strategy can also be used to build ligands to metalloproteins: describing correctly the metal-ligand interactions with a scoring function is challenging, thereby in the past our group has used a seeding approach to discover new ligands to human carbonic anhydrase$^\text{27}$.

The growth of molecules in OpenGrowth is biased to produce compounds that share properties with a training drug database, specifically the probability that two substructures are connected. To further increase the synthetic accessibility of the new molecules, we have also implemented the previously described 3mer screen. We have shown that when used together, these two options allow generating much better synthetic accessibility and pharmacokinetic/“drug-like” properties for the *de novo* molecules than for molecules created by random growth. We plan to include new options in
OpenGrowth to compute a synthetic accessibility score or pharmacokinetic/“drug-like” properties directly. These properties could be used during the growth process using multi-objective optimization: for example, addition of a new fragment might not only have to satisfy the Metropolis criterion, but also not increase the accessibility score of the molecule past a certain threshold. If the score increases too much after the addition of three fragments (e.g.) then this set of three fragments might be rejected. Such extended analysis will likely increase the success of the growth process.

One of the current limitations of OpenGrowth is the scoring function. Even though SMOG2001 has the same accuracy as the well-known GlideScore-SP with a correlation coefficient of $R=0.45$ with experimental data (see Supporting Information), this may be not enough to be used to guide growth with the Metropolis criterion or to efficiently sort structures. Moreover, the hard-wall potential may be a source of rejection for too many fragments. To overcome this, we are currently developing a more accurate scoring function in the same spirit as SMOG2001 (knowledge-based potential with a multi-shell approach), called SMOG2015. Its preliminary version is already implemented in OpenGrowth, and full details will be provided in a future publication. Briefly, a repulsive term and ligand entropy contribution (by the mean of the number of rotatable bonds$^{66}$) have been included in the new scoring function to increase its predictive power. Moreover, since many more complexes are available in the Protein Data Bank than in 2001, statistics that are more accurate have been obtained (see Supporting Information). Preliminary results using this scoring function are promising, since it allowed us to grow relatively large compounds in five target structures, whereas SMOG2001 failed to do the same. Due to the open-source nature of OpenGrowth, the implementation of existing scoring functions in our program by external contributors is also possible (and is explained in the user manual). In the future, the rotameric search, the calculation of the score for the Metropolis criterion and the final energy prediction may use different energy functions. For example, the final energy prediction could be similar to a MMPBSA calculation. Regarding the protein flexibility, future releases of OpenGrowth could include weights for the different
protein structures according to their free energies; moreover, in case we do not want the ligand to bind
to off-target proteins, it is also possible to include a penalty for those proteins.

Besides the demonstrated applications of OpenGrowth presented here, we want to underline
that the fields of applications of this program are very wide. For example, peptides can theoretically be
grown using the current version of OpenGrowth, but we plan to simplify this process by preparing the
needed fragments; this approach could serve to design new inhibitors to proteases by including non-
natural amino-acids in the fragment library. One can also think of growing compounds to strengthen
protein-protein interactions. Future releases of the program will allow the possibility to grow covalent
ligands using the seeding approach, which would be a major breakthrough in the realm of drug design.
Thus, the ease of use, the flexibility and the open nature of the OpenGrowth program makes it a strong
advancement over previous tools for drug design.

Supporting Information Available

More details on the growth (preparation and addition of a fragment), the optimization of the ligands,
score computation, the FOG probabilities and the nearest neighbors to new ligands are available in the
Supporting Information. This material is available free of charge via the Internet at http://pubs.acs.org.

Corresponding Author Information

Email: shakhnovich@chemistry.harvard.edu. Phone number: +1-617-495-4130.

Acknowledgments

This work was supported by Defense Advanced Research Projects Agency (DARPA) contract HR0011-11-
C-0093. The funders had no role in study design, data collection and analysis, decision to publish or
preparation of the manuscript. The authors thank Dr. Peter Kutchukian for his comments on the
manuscript. N.C. thanks the Région Rhône-Alpes for an Explora’Pro fellowship. The computations in this
paper were run on the Odyssey cluster supported by the FAS Division of Science, Research Computing
Group at Harvard University.

References

1. Cavasotto, C. N.; Phatak, S. S., Homology modeling in drug discovery: current trends and


920-937.

4. (a) Kitchen, D. B.; Decornez, H.; Furr, J. R.; Bajorath, J., Docking and scoring in virtual screening
for drug discovery: methods and applications. Nat Rev Drug Discov 2004, 3 (11), 935-49; (b) Waszkowycz,
B.; Clark, D. E.; Gancia, E., Outstanding challenges in protein–ligand docking and structure-based virtual

5. (a) Irwin, J. J.; Shoichet, B. K., ZINC--a free database of commercially available compounds for
virtual screening. J Chem Inf Model 2005, 45 (1), 177-82; (b) Irwin, J. J.; Sterling, T.; Mysinger, M. M.;
Bolstad, E. S.; Coleman, R. G., ZINC: a free tool to discover chemistry for biology. J Chem Inf Model 2012,
52 (7), 1757-68.

6; (b) Novac, N., Challenges and opportunities of drug repositioning. Trends Pharmacol Sci 2013, 34 (5),
267-72; (c) Sardana, D.; Zhu, C.; Zhang, M.; Gudivada, R. C.; Yang, L.; Jegga, A. G., Drug repositioning for

7. (a) Riggs, B. L.; Hartmann, L. C., Selective estrogen-receptor modulators -- mechanisms of action
and application to clinical practice. N Engl J Med 2003, 348 (7), 618-29; (b) Vogel, V. G.; Costantino, J. P.;
Wickerham, D. L.; Cronin, W. M.; Cecchini, R. S.; Atkins, J. N.; Bevers, T. B.; Fehrenbacher, L.; Pajon, E. R.,


A. D.; Farkas, Ö.; Foresman, J. B.; Ortiz, J. V.; Cioslowski, J.; Fox, D. J. *Gaussian 09*, Gaussian, Inc.: Wallingford, CT, USA, 2009.


Homeyer, N.; Gohlke, H.; Roitberg, A. E., MMPBSA.py: An Efficient Program for End-State Free Energy
65. Hou, T.; Yu, R., Molecular Dynamics and Free Energy Studies on the Wild-type and Double
66. (a) Hou, T.; Wang, J.; Li, Y.; Wang, W., Assessing the Performance of the MM/PBSA and


