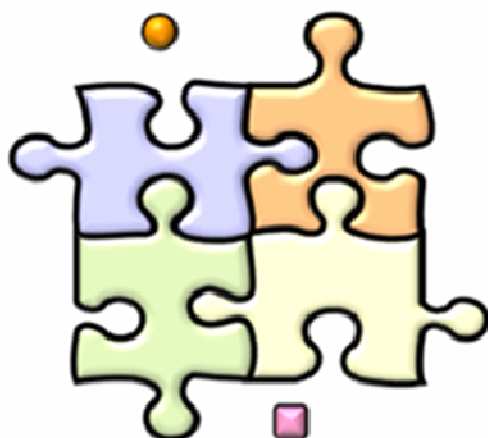




Automatic Tailoring & Transplanting Version 2.0 User Manual



Shanghai Institute of Organic Chemistry
Chinese Academy of Sciences

Table of Contents

1. Introduction	1
1.1 AutoT&T as a novel <i>de novo</i> design approach.....	1
1.2 The basic framework and running modes of AutoT&T v.2.....	2
1.3 The post-processing utilities	6
2. Download and Installation of AutoT&T v.2.....	7
2.1 System requirements.....	7
2.2 Download and installation	7
3. Main Modules in AutoT&T v.2	10
3.1 Overview	10
3.2 Usage of AutoT&T2 modules	11
3.2.1 The LeadOpt2 module.....	11
3.2.2 The GrowLeadOpt Module.....	15
3.2.3 The LinkLeadOpt Module	17
3.2.4 The Optimize module.....	18
3.2.5 The Score module	20
3.2.6 The Cluster module	21
3.2.7 The Filter module.....	23
4. Demo Web Interface.....	26
5. Application Examples.....	28
5.1 Test case 1: Single-round optimization.....	28
5.2 Test case 2: Multi-round optimization.....	30
5.3 Test case 3: Multi-thread optimization.....	32
5.4 Test case 4: Design of covalent binders.....	36
Copyright and Contact Information	40
References	41

1. Introduction

1.1 AutoT&T as a novel *de novo* design approach

Since structure-based design has become the main-stream strategy in modern drug discovery, an ambitious approach called *de novo* design started to emerge in the early 1990s. Utilizing the power of computer, *de novo* design methods generate putative ligands to a given target protein automatically by incremental construction of molecular structures inside the binding pocket. An obvious advantage of such methods is that the solutions provided by them are not restricted to existing compounds. *De novo* design methods can also be used to conduct structural optimization of a given lead molecule. Following a systematic procedure, they are in principle able to explore a broader region of the chemical space than human experts.

The Automatic tailoring and transplanting (AutoT&T) method is developed in our group as a new type of *de novo* design method.[1] Unlike conventional build-up methods, AutoT&T does not rely on a pre-compiled fragment library for constructing molecular structures. Instead, AutoT&T uses a library of reference molecules for this purpose, which can be a random or purposely compiled assembly of real molecules. To conduct structural optimization of a given lead molecule, AutoT&T detects the suitable fragments on reference molecules, and then transplant those fragments onto the lead molecule to generate new structures (Figure 1).

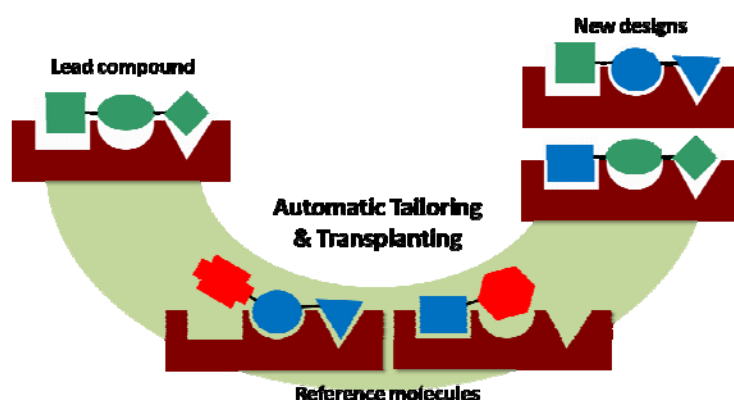


Figure 1. The basic idea of AutoT&T

As implied above, the building blocks used by AutoT&T are not retrieved from a pre-compiled fragment library but are truncated from a pool of real molecules in a dynamic

manner. Consequently, one does not experience the headache of compiling an “ideal” fragment library. Besides, the building blocks used in structural construction are not restricted to simple fragments. Complex fragments, if they match the structure of the lead molecule in a certain way, are acceptable as well. Because building blocks are directly truncated from real molecules, the ligand structures generated by AutoT&T are generally more diverse and synthetically more feasible than the outcomes by random connection of simple fragments.

AutoT&T also has some technical advantages over conventional build-up methods. First, AutoT&T does not need to perform conformational sampling during structural construction. It is because the binding poses of all reference molecules are generated in prior, for example, through a virtual screening job towards the given target protein. In this way, the outcomes of virtual screening, which typically consume a lot of CPU time to obtain but virtually have no use once finished, can be “recycled” by AutoT&T for lead design and optimization. In contrast, a conventional build-up method needs to perform some conformational sampling whenever connecting two fragments in order to put the resulting structure in a reasonable conformation. Second, AutoT&T is able to conduct a systematic crossover between the lead molecule and all given reference molecules because the possible matches between them are a finite number. A conventional build-up method constructs molecular structures by assembling fragments in an incremental manner. It thus needs a sampling algorithm during fragment assembling to deal with the “combinatorial explosion” problem. Due to the reasons mentioned above, AutoT&T is not only able to produce more reasonable designs, it is also more efficient than conventional de novo design methods.

The original version of AutoT&T was published in 2011.[1] Since then, we have refined the structural operation algorithms implemented in AutoT&T in several aspects to improve its efficiency. The new version of AutoT&T, i.e. AutoT&T v.2, is faster by up to a few thousand folds in a multi-round optimization job. It is also able to perform optimization based on multiple lead molecules or design of ligand molecules from scratch.

1.2 The basic framework and running modes of AutoT&T v.2

The AutoT&T2 software consists of three main modules (Figure 2). The structural operation module, which is the core module, is used to carry out structural optimization of the lead molecule by tailoring and transplanting suitable fragments from reference molecules. The binding affinity evaluation module is used to calculate the binding score of a

ligand molecule (or a fragment) needed by the structural operation module. Same as AutoT&T, the PLP scoring function is implemented in this module due to its excellent speed, acceptable accuracy, and technical convenience. In addition, a post-processing module is provided for assessing the ligand molecules generated by the structural operation module in order to select the promising candidates. This module includes a range of utilities for conducting *in situ* minimization of ligand molecules, clustering them by their chemical structures, and assessing their “drug-likeness”.

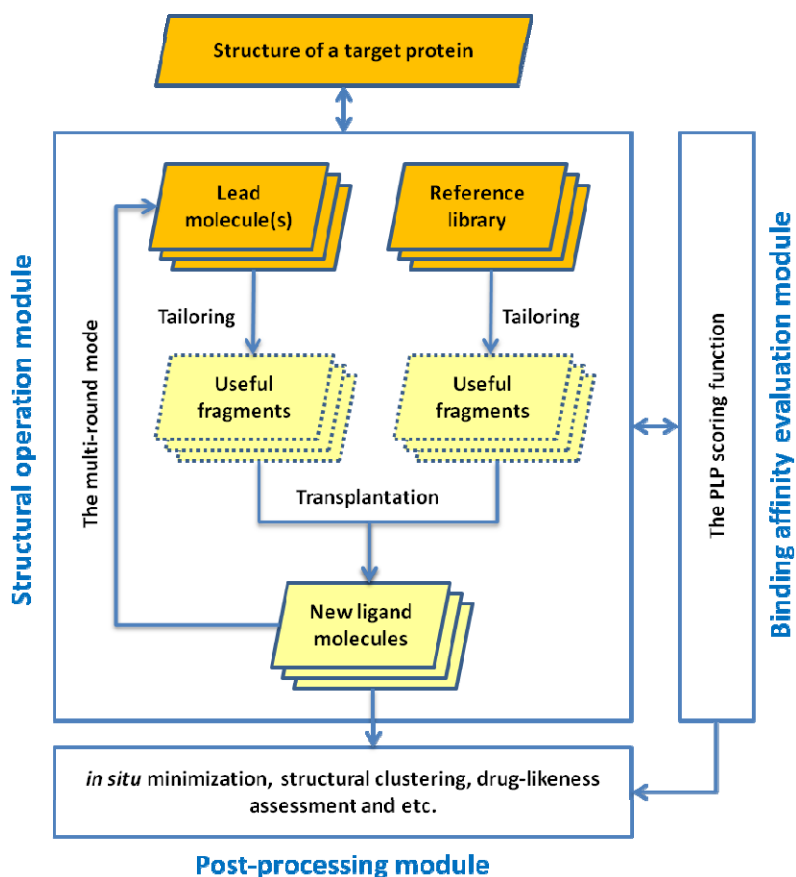


Figure 2. The overall framework of AutoT&T

Two main running modes are enabled in AutoT&T2. The first one is the **standard optimization mode** (Figure 3), which is also available in the previous version. In this mode, structural crossover between a given lead molecule and a library of reference molecules are conducted, i.e. useful fragments on the reference molecules are transplanted onto the lead molecule to generate new molecules. The user can set the maximal round of structural operations, but the program may automatically end up if no more sites on lead molecule are left for further optimization. The total number of output molecules by this running mode is typically proportional to the size of the given reference library.

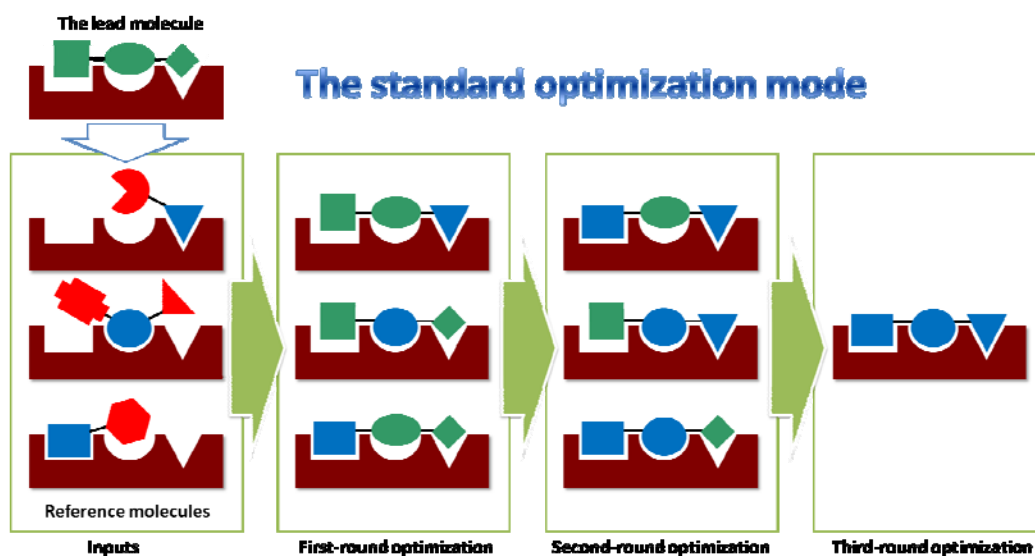


Figure 3. Illustration of the standard optimization mode in AutoT&T2.

The second running mode is the so-called **multi-thread optimization mode** (Figure 4), which is a new feature in AutoT&T2. In this running mode, the starting point is a group of lead molecules given by the user. Each lead molecule is structurally optimized in turn while the other lead molecules in the pool are used as the reference molecules. In other words, this running mode is designed to perform structural crossover among multiple lead molecules. This is an effective approach frequently adopted by medicinal chemists to develop new compounds based on known active compounds. It needs to be pointed out that in this running mode, the input molecules do not have to be known binders to a target protein. Instead, they can be a group of arbitrary molecules. If so, this running mode actually generate ligand molecules for the target protein from scratch, which further expands the possible applications of AutoT&T2.

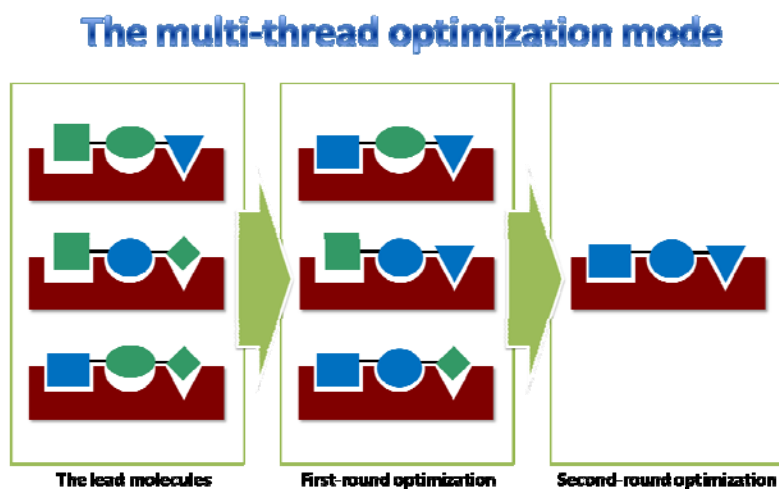


Figure 4. Illustration of the multi-thread optimization mode in AutoT&T2.

Besides the two main running modes, two supplementary running modes are also given, i.e. the **growing mode**, and the **linking mode**, which mimic the functions of conventional build-up methods more closely. They may be more suitable or more convenient in certain cases.

In the growing mode (Figure 5), the given lead is usually a fragment occupying a key site inside the binding point as a “seed”. Then, a suitable fragment truncated from a reference molecule is installed onto the lead molecule. This process is repeated until the ligand molecule fills up the entire binding pocket.

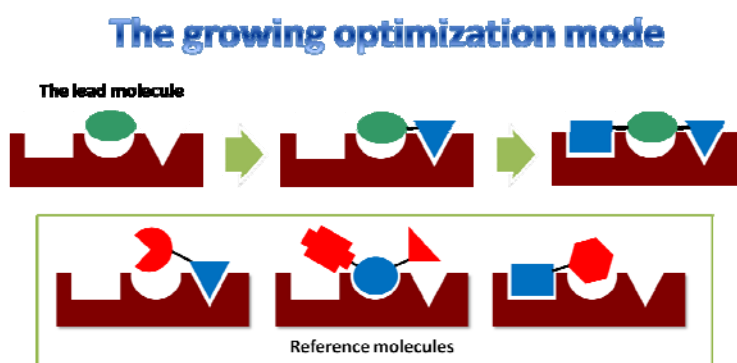


Figure 5. Illustration of the growing optimization mode in AutoT&T2.

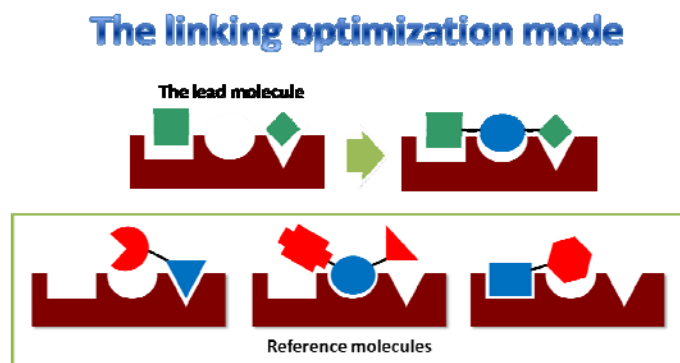


Figure 6. Illustration of the linking optimization mode in AutoT&T2

In the linking mode (Figure 6), the given lead usually consists of multiple fragments occupying different key sites inside the binding point. Then, the program tries to find a suitable fragment from a reference molecule that is able to connect at least two given fragments. This process is repeated until all given fragments are connected into an integrated molecule structure. Such a task is difficult for conventional build-up methods but relatively easy for AutoT&T2.

1.3 The post-processing utilities

A few utilities are included in the post-processing module in AutoT&T2. They are provided to process and evaluate the outputs of the structural operation module, usually thousands of newly generated ligand molecules, to select the promising candidates for further consideration. These utilities include:

- **In situ energy minimization.** AutoT&T2 constructs ligand structures by assembling fragments. The resulting molecular structures may not be in ideal binding mode. This function conducts energy minimization of the given ligand molecules within the geometrical constraints inside the binding pocket. Energy minimization is conducted either by using the Tripos force field [2] or the AMBER force field [3]
- **Computation of binding scores.** Binding scores of the given ligand molecules can be computed by the piecewise linear pairwise (PLP) scoring function.[4]
- **Clustering of ligand molecules by their chemical structures.** AutoT&T2 typically generates thousands of ligand molecules in a single job. Many of them may share a similar chemical scaffold. It is thus helpful to cluster these molecules according to their chemical structures so that the user can find the representative structures more efficiently. The popular ECFP fingerprint [5] is adopted in AutoT&T2 to encode chemical structures.
- **Assessment of “drug-likeness”.** A function is provided to compute several key descriptors for drug-likeness assessment, including molecular weight, number of heavy atoms, number of hydrogen bond donors, number of hydrogen bond acceptors, number of rotatable single bonds, number of rings, and the octanol/water partition coefficient (logP). The logP value is computed with the XLOGP3 method developed in our group.[6]

It should be emphasized that all of the utilities included in the post-processing module are optional. It is because many computational tools, either implemented in commercial software or available from academic groups, can conduct the same tasks. The user is free to employ the tools in his/her own favor for processing the outcomes of the structural operation module instead of the default utilities provided by AutoT&T2.

2. Download and Installation of AutoT&T v.2

2.1 System requirements

Each module of AutoT&T2 can be run separately through a command line. They can run in DOS on Microsoft Windows platforms, or SHELL (csh, tcsh, bash) on Linux/Unix/Mac OS X platforms. Supported operating systems include:

- Microsoft Windows[®] XP / 2003 / 2008 / Vista / Win7 / Win8 / Win10
- Linux[®], Kernel 2.4 or higher (gcc 3.2 or higher compiler: <http://gcc.gnu.org/>)

2.2 Download and installation

AutoT&T2 is written in the ANSI C++ language. It has been tested on Windows and Linux platforms. Users can download the complete package of AutoT&T2, including the executable and source codes, user manual, and demo examples, as a RAR file from our group web site. It can be easily installed through following steps:

1) Obtain the AutoT&T2 package






The AutoT&T2 package can be downloaded from our group web site at:



<http://www.sioc-ccbg.ac.cn/software/ATT2/>.

The user needs to sign a license agreement first to get this package. The license agreement can be found and signed on the same web page. The user also needs to supply some necessary contact information to complete the registration.

2) Decompress the package

After the package is decompressed, one can get a directory named as "ATT2/", which includes the following sub-directories:

	Makefile/	Makefile scripts for compiling source codes
	bin/	Default directory for saving executable codes of AutoT&T2
	bin-linux/	Back up of executable codes for Linux platforms
	bin-win/	Back up of executable codes for Windows platforms
	examples/	Several demo examples for running AutoT&T2

	manual/	User manual
	modules/	Source codes of all relevant modules in AutoT&T2

Special note: For users of the Windows system, one also needs to copy the runtime DLL library files included in the RAR package to the Windows system directory, i.e. C:\windows\system32\.

3) Compile the source codes

The pre-compiled executable codes of all major modules of AutoT&T have been provided under the "[bin-linux/](#)" and "[bin-win/](#)" subdirectories. One can also choose to compile the sources codes by oneself as follows.

For users of Linux systems, one should go to the "[Makefile/](#)" subdirectory, run the "[make_all](#)" script, and then move the resulting executable codes to the "bin/" directory.

For users of Windows systems, [MinGW](#), i.e. "Minimalist GNU for Windows" available from <http://www.mingw.org/>, is required for compiling the source codes. MinGW provides a complete Open Source programming tool set, which is suitable for the development of native MS-Windows applications. It does not depend on any third-party C-runtime DLLs. It does need a number of DLLs provided by Microsoft as components of the operating system

In brief, MinGW includest the following components:

- A port of the GNU Compiler Collection (GCC), including C, C++, ADA and Fortran compilers;
- GNU Bin utils for Windows (assembler, linker, archive manager)
- A command-line installer, with optional GUI front-end, ([mingw-get](#)) for MinGW and MSYS deployment on MS-Windows
- A GUI first-time setup tool ([mingw-get-setup](#)), to get you up and running with [mingw-get](#).
- MSYS, a contraction of "Minimal SYStem", is a Bourne Shell command line interpreter system. Offered as an alternative to Microsoft's cmd.exe, this provides a general purpose command line environment, which is particularly suited to use with MinGW, for porting of many Open Source applications to the MS-Windows platform.

Once MinGW is installed, one can compile AutoT&T2 with the "make" command in the

MSYS command line of MinGW just as in a Linux shell. No modification on the source codes is necessary. One just needs to copy some DLL library files provided by MinGW into the “C:\windows\system32” directory under your Windows system. For the convenience of the user, these DLL files are also saved in the “bin-win/” subdirectory along with the pre-compiled executable codes.

4) Setting environment variables

The default working directory of AutoT&T2 is the “bin/” directory, where all executable codes can be found. If one wishes to run AutoT&T2 at any path of his/her system for the sake of convenience, some environment variables need to be set as follows. This step is highly recommended.

a) In Microsoft Windows system, just attach the path of bin/ of AutoT&T2 to the system environment variable. For Windows XP system (applicable to Win7, Windows 2000, Windows 2003 and other Windows as well) : open “*control panel → performance and maintenance → system → advance → environment variable*”, edit the *PATH* variable, attach the path of bin/ of AutoT&T2 to existing content of *PATH*.

b) For Linux System, if you use *Csh* or *Tcsh* as login shell, please add the following lines to your personal profile (~/.cshrc or ~/.login or ~/.profile):

```
setenv ATT2_HOME root path of AutoT&T2 installation
setenv ATT2_BIN $ATT2_HOME/bin
set path = ($path $ATT2_BIN)
```

If one uses *Bash* as login shell, add the following lines to your personal profile (~/.bashrc):

```
export ATT2_HOME= root path of AutoT&T2 installation
export ATT2_BIN=$ATT2_HOME/bin
export $PATH=$ATT2_BIN:$PATH
```

For other login shells, please consult your system administrator on how to add new environment variables.

3. Main Modules in AutoT&T v.2

3.1 Overview

All of the executable binary codes under the “bin/” directory of AutoT&T2 are summarized in Table 1, including structural operation modules “LeadOpt2”, “GrowLeadOpt” and “LinkLeadOpt”, and post-processing modules “Optimize”, “Score”, “FrameworkCluster”, “Cluster” and “Filter”.

Table 1. Overview of AutoT&T2 modules

Module Name		Function
Windows	Linux	
LeadOpt2.exe	LeadOpt2	Optimize a given lead molecule. AutoT&T v.2 also supports optimization of multiple lead molecules, i.e. the multi-thread optimization mode.
GrowLeadOpt.exe	GrowLeadOpt	Optimize a given lead molecule with the growing method, see Figure 5 (page 5).
LinkLeadOpt.exe	LinkLeadOpt	Optimize a given lead molecule with the linking method, see Figure 6 (page 6).
Optimize.exe	Optimize	Perform energy minimization of ligand structures inside the binding pocket.
Score.exe	Score	Compute the binding affinity between ligand and target protein with the PLP scoring function.
Cluster.exe	Cluster	Cluster a group of given molecules by their structural similarity.
FrameworkCluster.exe	FrameworkCluster	Cluster a group of given molecules via the molecular framework analysis
Filter.exe	Filter	Screen a group of given molecules by drug-likeness properties

3.2 Usage of AutoT&T2 modules

3.2.1 The LeadOpt2 module

“LeadOpt2” is the core structural operation module in AutoT&T2. The synopsis of running “LeadOpt2” is as follows:

```
LeadOpt2 -l lead.mol2 -vs ref_lib.mol2 -p protein.pdb -o outputs.mol2
```

Here, “lead” is the lead molecule to be optimized (in Mol2 format); “ref_lib” is the library of reference molecules (in Mol2 format); “protein” is the structure of target protein (in PDB format). “Outputs” is the output file that records the final results, i.e. new ligand molecules (in Mol2 format).

Note that: (1) To run LeadOpt2 properly, both the lead and the whole reference library should be docked into the binding pocket on the target protein in prior. (2) “lead” and “ref_lib” may be in SDF format as well, which needs to be indicated by the suffix of the file name, such as “lead.sdf” and “ref_lib.sdf”.

Besides the standard mode described above, two new running modes are enabled by LeadOpt2 in AutoT&T v.2, i.e. batch optimization of multiple lead molecules and multi-thread optimization among multiple lead molecules (Figure 4 on page 5). To enable batch optimization of multiple lead molecules, one needs to use the **-L** flag to assign the input lead molecules, such as:

```
LeadOpt2 -L leads.mol2 -vs ref_lib.mol2 -p protein.pdb -o outputs.mol2
```

To enable multi-thread optimization among multiple lead molecules, one still needs the **-L** flag to assign the input lead molecules. Note that in this mode, the reference library is not necessary:

```
LeadOpt2 -L leads.mol2 -p protein.pdb -o outputs.mol2
```

There are some extra optional parameters in the LeadOpt2 module:

Table 2. Other optional parameters for running LeadOpt2

Short name	Full name	Description
-i	--iteration	Maximal rounds of optimization, normally between

		1~5
-t	--top	Maximal molecules kept after each round of optimization, normally below 10000 (see note A below).
-c	--constrain	Specify the desired optimization sites on the lead molecule, format is “-c (id1, id2) [(id3, id4)]” (see note B below). This flag is not applicable to the batch optimization mode or the multi-thread optimization mode.
-r	--recap	Enable the RECAP rules to assess matched bonds (see note C below).
-ih	--includeH	Consider chemical bonds connecting hydrogen atoms in detecting matched bonds (see note D below)
-mm	--matchmethod	Designate the algorithm for detecting matched bonds, either AP (atom-pair based algorithm) or BC (bond-center based algorithm). The default option is AP (see note E below).
-d	--distance	Set the distance cutoff for detecting matched bonds, where the default value for the AP algorithm is 1.0Å; and the default value for the BC algorithm is 0.5Å (see note E below).
-a	--angle	Set the angle cutoff in detecting matched bonds, where the default value is corresponding to 15° for both AP or BC algorithm (see note E below).
-v	--version	Display the version of the LeadOpt2 module
-h	--help	Display the help information

Special notes:

(A) This parameter sets the maximal number of ligand molecules that will enter the next round of optimization. The value of 10000 is more than adequate for most standard jobs.

(B) Users can either let LeadOpt2 automatically detect appropriate optimization sites on the given lead molecule, or assign one or more chemical bonds (cannot be in a ring) as optimization sites. When using parameter -c to assign optimization sites, fragments

connecting to the first atom id will be retained, while fragments connecting to the second atom id will be replaced if there are proper substitutes with higher binding affinity. For example, id1 and id3 denote the atom id that should be kept on the lead molecule; while id2 and id4 denote the atom id bound to id1 and id3, respectively. Atoms id2 and id4 will be replaced by fragments transplanted.

If multiple bonds are assigned as optimization sites and users do not provide the `-i` flag, **LeadOpt2** will still perform multi-round optimization according to the number of optimization sites. The flowchart of multi-round tailoring and transplanting is shown in Figure 7.

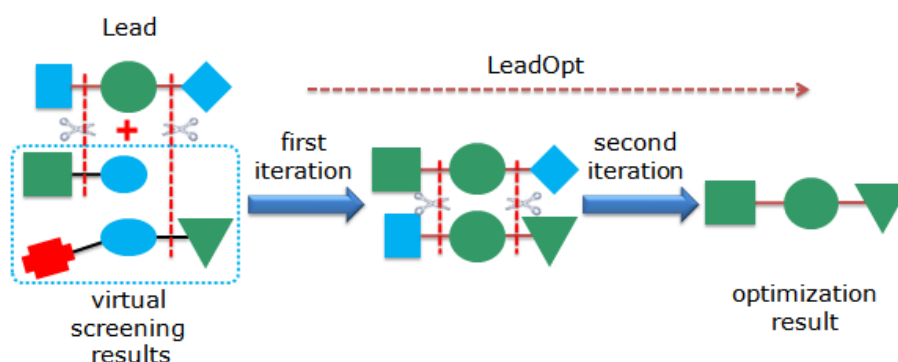


Figure 7. Flowchart of multi-round optimization by using LeadOpt2

(C) The REtrosynthetic Combinatorial Analysis Procedure (RECAP) proposed by Lewell et al [7] is implemented in AutoT&T to deal with this problem. According to this scheme, a total of 11 types of chemical bonds are defined to be “breakable”, each of which is related to a certain type of real chemical reaction (Figure 8). Breaking or recombining the chemical structures at these bonds may produce molecules that are more feasible for organic synthesis. RECAP analysis is optional in AutoT&T. If enabled, whenever a pair of matched bonds is found between the lead molecule and a reference molecule, the program will check if this bond belongs to one of these 11 types defined in the RECAP scheme. If not, this matched bond pair will be ignored.

Introduction of the RECAP analysis takes the synthetic feasibility into account directly in structural operations. But it should be pointed out that the RECAP analysis is a very simple treatment. The user is encouraged to employ other more advanced methods for assessing the final outputs of AutoT&T in terms of synthetic feasibility.

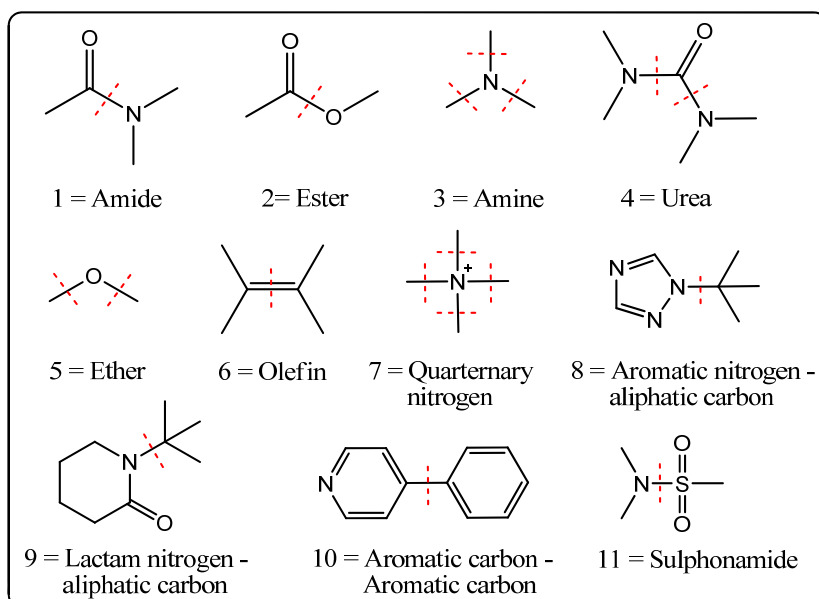


Figure 8. Eleven types of breakable chemical bonds defined in RECAP analysis

Table 3. Two algorithms for matched bonds detection

Atom-pair based	Bond-center based
<p>● atom — bond</p>	<p>● atom — bond</p>
<p>Matched conditions:</p> <ol style="list-style-type: none"> 1. Neither of the two bonds locates on a ring. 2. Distances r_1 and r_2 between each terminal atom of two bonds are both smaller than 1.0 \AA. 3. The angle ϑ between two bonds $\leq 15^\circ$. 	<p>Matched conditions:</p> <ol style="list-style-type: none"> 1. Neither of the two bonds locates on a ring. 2. Distance d between each center of the two bonds is smaller than 0.5 \AA. 3. The angle ϑ between two bonds $\leq 15^\circ$.

(D) By default, AutoT&T only considers chemical bond formed between two heavy atoms as potential site for conducting fragment transplanting. This treatment reduces the complexity in structural operation. However, a limitation by this treatment is that a fragment cannot be “grown” from a terminal atom linking with a hydrogen atom on the lead

molecule. To overcome this limitation, the user may enable this parameter to allow chemical bond connecting a hydrogen atom, i.e. X—H, as potential sites for optimization.

(E) Two algorithms are implemented in AutoT&T for judging if two given bonds are geometrically overlapped, i.e. "matched". One is based on matching atom pairs; while the other is based on matching bond centers (see Table 3). Both of them require that the matched bonds should not locate on a ring. The atom-pair based algorithm considers the distances between both ends of the two given bonds (r_1 and r_2); while the bond-center based algorithm considers the distance between the centers of two bonds (d).

3.2.2 The GrowLeadOpt Module

This module, together with the LinkLeadOpt module, is inherited from AutoT&T v.1. These two modules are designed to perform lead optimization in a way similar to conventional build-up methods. They serve as supplementary approaches to the standard mode of AutoT&T because they may be more suitable or more convenient in particular cases.

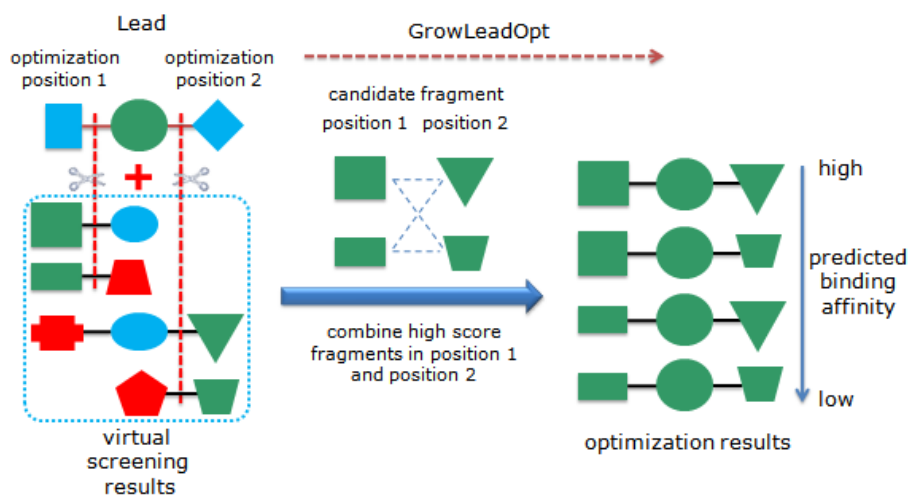


Figure 9. Illustration of the basic idea of GrowLeadOpt

GrowLeadOpt is designed to perform optimization on one or more sites while keeping the core fragment of the lead molecule. GrowLeadOpt searches for proper fragments in all reference molecules for each given optimization site, and then sort them by predicted binding affinities. Finally, new ligand molecules are assembled with these candidate fragments (see Figure 9).

The synopsis of running GrowLeadOpt is as follows:

**GrowLeadOpt -l lead.mol2 -vs ref_lib.mol2 -p protein.pdb -o output
-c (id1, id2) [(id3, id4)]**

Here, “*lead*” is the lead molecule to be optimized (in Mol2 format); “*ref_lib*” is the library of reference molecules (in Mol2 format); “*protein*” is the structure of target protein (in PDB format). “*Outputs*” is the output file that records the final results, i.e. new ligand molecules (in Mol2 format). The atom ids following the “-c” flag indicates the desired optimization site. Id1 and id3 are the atom ids at each desired optimization site which are on the core fragment of the lead molecule, while id2 and id4 are the atom ids connecting to atoms id1 and id3, respectively, which will be replaced by new fragments.

Note that: (1) To run [GrowLeadOpt](#) properly, both the lead and the whole reference library should be docked into the binding pocket on the target protein in prior. (2) “*lead*” and “*ref_lib*” may be in SDF format as well, which needs to be indicated by the suffix of the file name, such as “*lead.sdf*” and “*ref_lib.sdf*”. (3) All of the desired optimization site must be assigned on the lead molecule.

There are some extra optional parameters in the [GrowLeadOpt](#) module:

Table 4. Optional parameters for running GrowLeadOpt

Short name	Full name	Description
-t	--top	Maximal molecules kept after each round of optimization, normally below 10000
-r	--recap	Enable the RECAP rules to assess matched bonds (see Page 14)
-ih	--includeH	Consider chemical bonds connecting hydrogen atoms in detecting matched bonds (see Page 14)
-mm	--matchmethod	Designate the algorithm for detecting matched bonds, either AP (atom-pair based algorithm) or BC (bond-center based algorithm). The default option is AP (see Page 14-15)
-d	--distance	Set the distance cutoff for detecting matched bonds, where the default value for the AP algorithm is 1.0Å; and the default value for the BC algorithm is 0.5Å (see Page 14-15)

-a	--angle	Set the angle cutoff in detecting matched bonds, where the default value is corresponding to 15° for both AP or BC algorithm (see Page 14-15).
-v	--version	Display the version of the GrowLeadOpt module
-h	--help	Display the help information

3.2.3 The LinkLeadOpt Module

Similar to the GrowLeadOpt module, this module is also designed to perform lead optimization as conventional build-up methods. LinkLeadOpt generates ligand structures that connect several fragments placed inside the binding pocket on the target protein (see Figure 10). Thus, it provides an automatic way to conduct “fragment-based design”. If multiple optimization sites are assigned on the given fragments, LinkLeadOpt will search among the reference molecules for suitable linkers that can connect all of these optimization sites. If such a linker is indeed possible, LinkLeadOpt will transplant it onto the lead molecule (i.e. the given fragments) and evaluate whether the new ligand has a higher binding affinity.

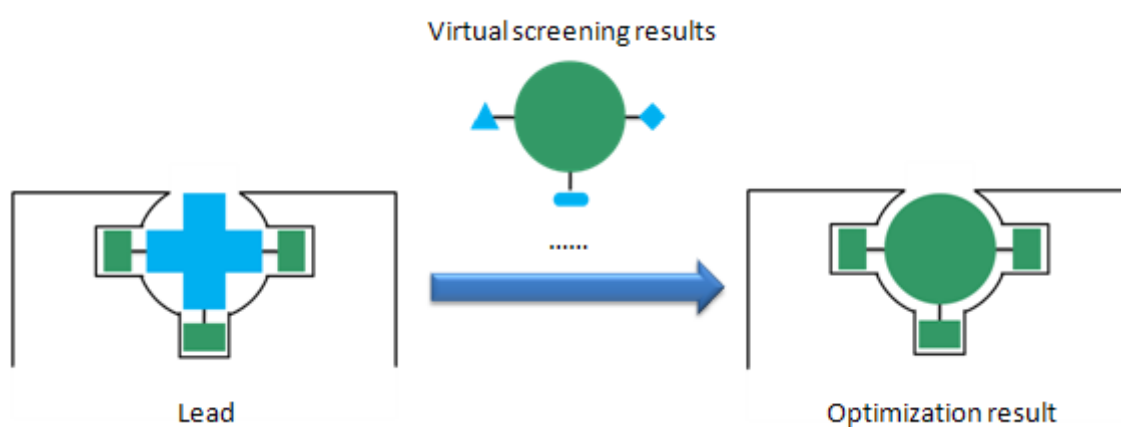


Figure 10. Schema of optimization in LinkLeadOpt

The typical synopsis of running LinkLeadOpt is as follows:

```
LinkLeadOpt -l lead.mol2 -vs ref_lib.mol2 -p protein.pdb -o output.mol2
-c (id1, id2) [(id3, id4)]
```

Here, “lead” is the lead molecule to be optimized (in Mol2 format); “ref_lib” is the library of reference molecules (in Mol2 format); “protein” is the structure of target protein (in PDB format). “Outputs” is the output file that records the final results, i.e. new ligand molecules (in Mol2 format). The atom ids following the “-c” flag indicates the desired

optimization site. Id1 and id3 are the atom ids at each desired optimization site which are on the core fragment of the lead molecule, while id2 and id4 are the atom ids connecting to atoms id1 and id3, respectively, which will be replaced by new fragments.

Note that: (1) To run **LinkLeadOpt** properly, both the lead and the whole reference library should be docked into the binding pocket on the target protein in prior. (2) “lead” and “ref_lib” may be in SDF format as well, which needs to be indicated by the suffix of the file name, such as “lead.sdf” and “ref_lib.sdf”. (3) At least two optimization sites should be assigned on the lead molecule.

The **LinkLeadOpt** module also uses the same set of optional parameters as **GrowLeadOpt**. The user may refer to Table 4 (page 15-16) for the descriptions of those parameters.

3.2.4 The Optimize module

This is one of the post-processing modules included in the AutoT&T2 package. The **Optimize** module can be used to optimize the binding mode of generated ligand molecules. AutoT&T2 generates ligand molecules basically through assembling chemical fragments. The resulting molecular structures may not be in an optimal conformation. Thus, energy minimization of the generated ligand structures inside the binding pocket on the target protein, i.e. *in situ* minimization, is necessary to refine the binding mode.

Note that application of the **Optimize** module, as other post-processing functions included the AutoT&T v.2 package, is optional. The user may use other suitable software to complete the same task. This gives the user flexibility to obtain the optimal results.

The typical synopsis of the Optimize module is as follows:

```
Optimize -l ligand.mol2 -p protein.pdb -o output.mol2
```

Here, “ligand” is the ligand molecule that requires energy minimization (in the Mol2 format or the SDF format); “protein” is the target protein (in PDB format); “output.mol2” is the final results.

In the current version, the Tripos force field and the AMBER force field are implemented for energy minimization in this module. The potential energy functions of these two force fields are as follows:

The Tripos force field:

$$V_{(r^N)} = \sum_{bonds} \frac{1}{2} k_b (l - l_0)^2 + \sum_{angles} \frac{1}{2} k_\theta (\theta - \theta_0)^2 + \sum_{oops} \frac{1}{2} k_{oop} d^2 \\ + \sum_{torsions} \frac{1}{2} V_n [1 + S \cos(|n| \cdot \omega)] + \sum_{j=1}^{N-1} \sum_{i=j+1}^N \left\{ E_{ij} \left[\frac{1.0}{a_{ij}^{12}} - \frac{2.0}{a_{ij}^6} \right] + 332.17 \times \frac{q_i q_j}{D_{ij} r_{ij}} \right\}$$

The AMBER force field:

$$V_{(r^N)} = \sum_{bonds} \frac{1}{2} k_b (l - l_0)^2 + \sum_{angles} \frac{1}{2} k_\theta (\theta - \theta_0)^2 + \sum_{torsions} \frac{1}{2} V_n [1 + \cos(n\omega - \gamma)] \\ + \sum_{j=1}^{N-1} \sum_{i=j+1}^N \left\{ 4\epsilon_{i,j} \left[\left(\frac{\sigma_{ij}}{r_{ij}} \right)^{12} - \left(\frac{\sigma_{ij}}{r_{ij}} \right)^6 \right] + \frac{q_i q_j}{4\pi\epsilon_0 r_{ij}} \right\}$$

The available minimization algorithms for energy minimization include: Simplex, Powell, Steepest Descent, and Conjugate Gradient. Users can choose the preferred force field and the minimization algorithm, set maximal steps of minimization and allow early termination of program. During minimization, the target protein structure is kept fixed; while the ligand molecule is treated as flexible. Here, bond lengths, bond angles, and torsion angles on the ligand structure may be adjusted during energy optimization.

The optional parameters for running the **Optimize** module are summarized in Table 5. If no optional parameter is given, then the Tripos force field, the Simplex algorithm and a maximal of 100 steps are set by default.

Table 5. Optional parameters for running the Optimize module

Short name	Full name	Description
-g	--gaff	Choose the AMBER GAFF force field
-t	--taff	Choose the Tripos force field
-ie	--ignoreE	Ignore electrostatic energy
-sp	--Simplex	Set the maximal steps for the Simplex algorithm
-pw	--Powell	Set the maximal steps for the Powell algorithm
-sd	--SteepestDescent	Set the maximal steps for the Steepest Descent algorithm
-cg	--ConjugateGradients	Set the maximal steps for the Conjugate Gradients algorithm
-log	--logfile	Designate a log file
-v	--version	Display the version of the Optimize module
-h	--help	Display the help information

3.2.5 The Score module

This is one of the post-processing modules included in the AutoT&T2 package. The **Score** module is designed for evaluation of protein-ligand interactions. A binding score between a protein and a ligand is computed by using the piecewise linear potential (PLP) scoring function [4]. PLP is an empirical scoring function, which sums up distance-dependent potentials of the atom pairs between a protein and a ligand. It defines four types of heavy atoms on the protein and the ligand, i.e. hydrogen bond donor, hydrogen bond acceptor, hydrogen bond donor & acceptor, and non-polar atom. The hydrogen bond interactions and steric interactions between these atoms are considered, which are calculated on the basis of the potential function shown in Figure 11. The interaction types and parameters are provided in Table 6 and Figure 11 respectively.

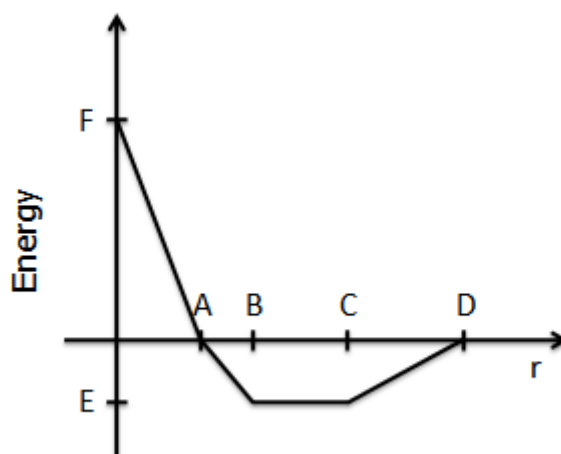


Figure 11. The stepwise potential function used in the PLP scoring function. Here A, B, C and D are distance cutoffs in Å; E and F are binding scores in an arbitrary unit. For steric interactions, A = 3.4, B = 3.6, C = 4.5, D= 5.5, E = -0.4, F = 20.0. For hydrogen bond interactions, A = 2.3, B = 2.6, C = 3.1, D= 3.4, E = -2.0, F = 20.0.

Table 6. The interaction types of different atom types on the protein and the ligand

Atoms on the protein					
		Hydrogen bond donor	Hydrogen bond acceptor	Hydrogen bond donor & acceptor	Non-polar atom
Atoms on the ligand	Hydrogen bond donor	I	II	II	I

Hydrogen bond acceptor	II	I	II	I
Hydrogen bond donor & acceptor	II	II	II	I
Non-polar atom	I	I	I	I

I: steric interaction; II: hydrogen bond interaction

The PLP scoring function is chosen over other scoring functions due to its reasonable performance and technical advantages. First, some benchmarks of scoring functions also suggest that PLP is among the top ones in terms of general performance. Second, PLP is a pure atom-based model. It is convenient to implement and also has a faster speed. Besides, it is possible with PLP to compute the binding score of any fragment in a molecule, which is important for AutoT&T.

The typical synopsis of the **Score** module is as follows:

```
Score -l ligand.mol2 -p protein.pdb -o output.log [-s]
```

Here, “*ligand*” is the ligand molecule that requires energy minimization (in the Mol2 format or the SDF format); “*protein*” is the target protein (in PDB format); “*output.dat*” is a text file recording the computed binding score. A more negative binding score indicates a stronger protein-ligand interaction. If there are multiple ligands in the input file, users can add the “-s” flag to sort the results by binding scores.

3.2.6 The Cluster module

This is one of the post-processing modules included in the AutoT&T2 package. The final outputs of AutoT&T2 normally consist of a large number of molecules, from hundreds to thousands. It is therefore necessary to cluster them according to their structural similarity and find the representative ones conveniently. Currently, three algorithms for calculating molecular similarity are implemented in this module, including an Atom Pair (AP) based algorithm [8], a topological torsion descriptor (TTD) based algorithm [9], and an algorithm combining AP and TTD. Two clustering algorithms are implemented, i.e. one is a K-means like algorithm [10] and the other is the Stochastic Cluster Analysis (SCA) algorithm.[11] By default, the TTD algorithm is used for computing structural similarity, and the SCA algorithm

is used for clustering.

The input file for this module can be either a Mol2/SDF-format file containing multiple molecules. In such a case, the command line is:

Cluster -i input.mol2 -o output.log

The input file for this module can also be a similarity matrix computed by other algorithms in the format shown below. In such a case, the command line is:

Cluster -im similarity_matrix -o output.log

```

N
sim11 sim12 sim13 ... sim1N
sim21 sim22 sim23 ... sim2N
sim31 sim32 sim33 ... sim3N
...                simij ...
simN1 simN2 simN3 ... simNN

```

Figure 12. An example of the similarity matrix needed by the Cluster module. N is the dimensional of this matrix, i.e. the total number of molecules under consideration; sim_{ij} is the structural similarity between the i th and j th molecule. When i is equal to j , $sim_{ii}=1.0$

In addition, users can output the similarity matrix computed by the Cluster module on multiple molecules:

Cluster -i input.mol2 -om similarity_matrix

The full list of supported parameters is shown in Table 7:

Table 7. Optional parameters for running the Cluster module

Short name	Full name	Description
-i	--in	Filename of input structures
-im	--inmatrix	Filename of input similarity matrix
-o	--out	Output filename of clustering results
-om	--outmatrix	Output filename of the resulting similarity matrix
-ofp	--outfileprefix	Prefix of output structure filename, available only if parameter -i is set; output structure file has the same file extension with input structure file.
-tt	--TT	Use topological torsion descriptor (TTD) based method

		to calculate molecular similarity
-ap	-- AP	Use Atom Pair (AP) based method to calculate molecular similarity
-ttap	-- TTAP	Use topological torsion descriptor (TTD) and Atom Pair (AP) combined method to calculate molecular similarity
-sca	-- SCA	Set the clustering distance for improved Stochastic Cluster Analysis (SCA) method, ranging from 0-1, and the bigger value means less cluster number.
-km	-- KMeans	Set the desired number of clusters for K-means like method
-v	--version	Print the version of Cluster module
-h	--help	Print the help information

3.2.7 The Filter module

This is one of the post-processing modules included in the AutoT&T2 package. The **Filter** module is designed for filtering the molecules generated by AutoT&T2 by their "drug-likeness" properties. The properties computed by this module include: molecular weight, number of heavy atoms, number of hydrogen bond donors, number of hydrogen bond acceptors, total number of hydrogen bond donors and acceptors, number of rotatable bonds, number of rings, and the logP value (computed by XLOGP3 [6]).

The typical synopsis of the **Filter** module is as follows:

Filter -i input.mol2 -o output.mol2 -rf rule.txt

Here, "*input*" records the molecules under filtering (in Mol2 or SDF format); "*output*" records the molecules passing filtering (in Mol2 or SDF format). "*rule.txt*" is the configuration file defining all of the applicable rules. Users can either assign parameters in command line as shown in Table 8, or set up all parameters in a single configuration file as shown in Table 9.

Table 8. Optional parameters for running the Filter module

Short name	Full name	Description
-i	--in	Filename of input structures
-o	--out	Filename of output results
-rf	--rule	Filename of filtering parameters

-l	--log	Filename of output logs
-mw		Range of molecular weight: 100:500 MW in range of 100-500 Da 100: MW larger than 100 Da :500 MW smaller than 500 Da
-hbd		Maximal number of hydrogen bond donors
-hba		Maximal number of hydrogen bond acceptors
-rot		Maximal total number of hydrogen bond donors and acceptors
-rng		Maximal number of rings
-hv		Maximal number of heavy atoms
-xp		Maximal logP value
-fg		Treat molecules containing multiple fragments
-v	--version	Print the version of Filter module
-h	--help	Print the help information

Table 9. Example of the configuration file for running the Filter module

```

#### DEFINE YOUR OWN FILTERING RULES BELOW ####
#### A RULE STARTED WITH "#" WILL BE IGNORED ####
#
# RANGE OF ACCEPTED MOLECULAR WEIGHT
MOLECULAR_WEIGHT      0    500
# RANGE OF ACCEPTED NO. OF HYDROGEN BOND DONOR
NUMBER_HB_DONOR       0    5
# RANGE OF ACCEPTED NO. OF HYDROGEN BOND ACCEPTOR
NUMBER_HB_ACCEPTOR    0    10
# RANGE OF ACCEPTED NO. OF H-BOND ATOM (N & O ATOM)
NUMBER_HB_ATOM        0    15
# RANGE OF PARTITION COEFFICIENT
LOGP                   0    5
# RANGE OF ACCEPTED NO. OF ROTATABLE BOND
NUMBER_ROTATOR        0    10
# RANGE OF ACCEPTED NO. OF RING NUMBER
NUMBER_RING           0    5
# RANGE OF ACCEPTED NO. OF HEAVY ATOM

```

NUMBER_HEAVY_ATOM	10	50
# EXCLUDE MOLECULE HAS MULTIPLE FRAGMENT		
EXCLUDE_MULTI_FRAG	YES	

4. Demo Web Interface

A web portal is freely accessible at <http://www.sioc-ccbg.ac.cn/software/att2>, which is provided to demonstrate the main function of AutoT&T v.2. Through this web portal, user can perform on-line lead optimization jobs by running the AutoT&T2 software at the server end and then retrieve the outputs. Note that only the standard mode, i.e. optimization based on a single lead molecule, is currently enabled on this web server. Users are encouraged to download the AutoT&T2 software to test its full functions in a more efficient manner.

AutoT&T v2.0
Automatic Tailoring and Transplanting

Upload Structure Files		
Target Protein:	<input type="button" value="选择文件"/> <input type="button" value="未选择文件"/> (* in PDB Format)	
Lead Compound:	<input type="button" value="选择文件"/> <input type="button" value="未选择文件"/> (* in Mol2/SDF Format)	
Reference Library:	<input type="button" value="选择文件"/> <input type="button" value="未选择文件"/> (* in Mol2/SDF Format)	
Optional Parameters		
Optimization Parameters	<input checked="" type="radio"/> Standard optimization on single lead <input type="text" value="5"/> Maximal rounds of optimization Method for searching matched bonds: <input type="text" value="Atom Pair based"/>	
	<input type="text" value="1.0"/> distance cutoff for matching bond (Angstrom) <input type="text" value="15"/> Angle cutoff for matching bond (degree) <input type="text" value="1000"/> Maximal molecules retained after each round <input type="checkbox"/> Consider A-H bonds in matching <input type="checkbox"/> Use RECAP rules to filter matched bonds	
	Internal Scoring Function	
	<input checked="" type="radio"/> PLP scoring function <input type="checkbox"/> Enable energy minimization Force Field: <input type="text" value="Tripos Force Field"/>	
	Final Energy Minimization [Optional]	
	Method for minimization: <input type="text" value="Simplex"/> <input type="text" value="100"/> Maximal steps for energy minimization <input type="checkbox"/> Ignore electrostatic energy	
	Final Clustering Parameters	
Method for computing similarity: <input type="text" value="Topological Torsion Descriptor, TTD"/> Method for clustering: <input type="text" value="Stochastic Cluster Analysis, SCA"/>		
Final Druglikeness Filtering Parameters	<input type="text" value="0"/> - <input type="text" value="500"/> Molecular weight <input type="text" value="0"/> - <input type="text" value="10"/> Hydrogen bond acceptors <input type="text" value="0"/> - <input type="text" value="5"/> Hydrogen bond donors <input type="text" value="0"/> - <input type="text" value="10"/> Number of rotatable bonds <input type="text" value="0"/> - <input type="text" value="5"/> Number of rings <input type="text" value="10"/> - <input type="text" value="50"/> Number of heavy atoms <input type="text" value="0"/> - <input type="text" value="5"/> LogP value <input checked="" type="checkbox"/> Ignore molecules with multi-fragments	
	<input type="button" value="Default Setting"/> <input type="button" value="Submit Job"/>	

Figure 13. The web interface for submitting AutoT&T2 jobs. Once the job is completed at the server end, a new web page will be displayed for the user to download the results.

To submit a job, user is required to upload the necessary inputs, including the structure files of the target protein, a lead molecule, and a reference library. Note that both the lead molecule and the whole reference library need to be docked into the binding site on the target protein in prior.

Then, user may set a number of optional parameters in the “optimization parameters” section, including the maximal round of structural operation and a few more parameters related to the detection of matched bonds. Descriptions of these parameters are given in the 3.2 section (page 11-15) in this manual.

In the "final energy minimization" section, user may enable an optional step for performing energy minimization on the generated molecules to get more reasonable binding poses. This task can be conducted by using the Tripos force field or the Amber force field implemented in AutoT&T v.2.

In the "final structural clustering" section, user may choose the algorithm for computing molecular similarity and the algorithm for performing clustering.

In the "final drug-likeness filtering parameters", user may set the acceptable range of a few common drug-likeness descriptors for filtering the molecules generated by AutoT&T v.2.

Then, use may click the "submit job" button to submit the job. Once the job is completed at the server end, a new web page will be displayed for the user to download the results.

5. Application Examples

This section describes four test cases, in which AutoT&T2 is applied to different tasks. All of the necessary material for running these test cases can be found under the ["examples/"](#) directory in the AutoT&T2 package.

5.1 Test case 1: Single-round optimization

This test case is designed to demonstrate the standard running mode of AutoT&T2, i.e. optimization of a single given lead molecule. In this test case, the p38 MAP kinase was chosen as the molecular target. The target structure was retrieved from PDB entry 1W82, which is a complex formed between p38 MAP kinase and a small-molecule inhibitor. A small-molecule p38 MAPK inhibitor was chosen as the lead molecule (Figure 14). This particular compound was selected because its size is relatively small and its potency is only at the micromolar range ($IC_{50} = 40 \mu\text{M}$) so there is still enough room for further optimization. The binding pose of this lead molecule was generated by using the GOLD molecular docking software (version 5.2, Cambridge Crystallography Data Center). The segment containing the indole moiety was kept as the core fragment; while the benzoic amide moiety was designated to be replaced in optimization (Figure 14).

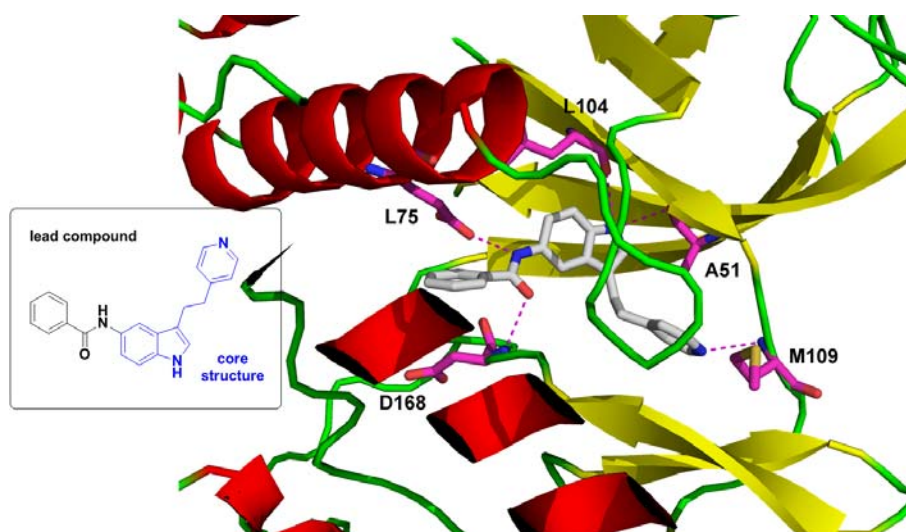


Figure 14. Binding mode of the lead molecule (in stick model) inside the binding pocket of p38 MAP kinase (in ribbon model). The p38 MAPK structure was taken from PDB entry 1W82. As for the lead molecule, the structure in blue was the core fragment to keep; while the benzoic amide moiety was to be optimized.

All of the necessary inputs for this test case can be found under the subdirectory

"examples/p38/". (1) "protein.pdb", the processed p38 MAPK structure retrieved from PDB entry 1W82; (2) "lead.sdf", structure of the lead molecule in SDF format; (3) "vs_results.sdf", the reference molecules, including a total of 1000 molecules randomly selected from the Available Chemical Directory database (ACD). Their binding poses inside the binding pocket on the target protein were also generated by the GOLD software. In this case, the C-N bond connecting the core fragment and the benzoic amide moiety was set as the optimization site, where fragment transplantation would be conducted (Figure 14).

First, one can use **GrowLeadOpt** to perform automatic tailoring and transplanting based on lead molecule structure and reference molecules. Here, the **-c** flag is used to assign the bond connecting atom #12 and #27 as the optimization site, where the fragment connecting to atom #12 will be retained; while fragment connecting to atom #27 will be replaced. The command line is:

```
GrowLeadOpt -l lead.sdf -p protein.pdb -vs vs_results.sdf -c (12, 27) -o result.sdf
```

Then, one can use the **Optimize** module to perform *in situ* energy minimization for the 46 new molecules recorded in "result.sdf". The command line is:

```
Optimize -l result.sdf -p protein.pdb -o optimized.sdf
```

Next, the binding scores between the ligand molecules and the target protein can be computed by the **Score** module. Those molecules are sorted by their binding scores. The command line is:

```
Score -l optimized.sdf -p protein.pdb -o score.log -s
```

If there are a large number of new molecules, one can do clustering with the "Cluster" module. In this case, the improved Stochastic Cluster Analysis (SCA) algorithm is applied to clustering with a similarity cutoff value of 0.6. Molecules in the N_{th} cluster are saved in "cluster_N.sdf" (N=1,2,3.....). The command line is:

```
Cluster -i optimized.sdf -om simmatrix.txt -ofp cluster -sca 0.6
```

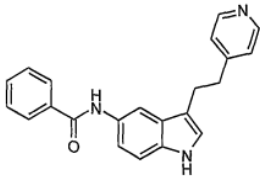
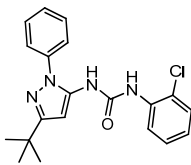
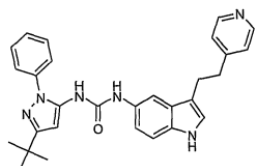
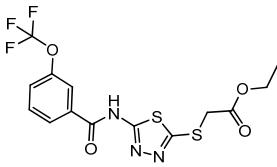
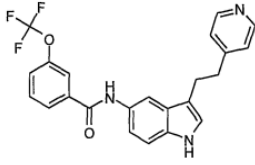
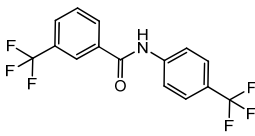
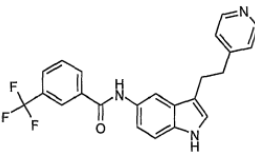
At the final step, one can use the **Filter** module to select the "drug-like" molecules among the outputs given by AutoT&T2. Here, one can set the desired filtering rules in "filter.rule" to process all molecules in "cluster_1.sdf" to "cluster_7.sdf". For demonstration, we only keep the molecules whose molecular weight are lower than 500Da, number of

hydrogen bond donors and acceptors are lower than or equal to 5. The final results are saved in "filter_result.sdf". The command line is:

Filter -i optimized.sdf -rf filter.rule -o filter_result.sdf

In total there are 44 molecules in "filter_result.sdf". Among them, some have already been reported in public patents with substantially higher potency than the lead molecule (Table 10)

Table 10. Some new ligand molecules generated by AutoT&T2 in this test case

The lead	Reference molecules	New ligands
 Lead (IC ₅₀ 40μM)	 MFCD04627975	 (IC₅₀ 145nM)
	 MFCD02169482	 (IC₅₀ 2μM)
	 MFCD00045090	 (IC₅₀ 4μM)

5.2 Test case 2: Multi-round optimization

This test case is designed to demonstrate how AutoT&T2 conducts multi-round optimization based on a single given lead molecule. Here, angiotensin converting enzyme (ACE) is chosen as the molecular target. Lisinopril was used as the lead molecule for structural optimization (Figure 15).

All of the necessary inputs for this test case can be found under the subdirectory "examples/ACE/". (1) "1086.pdb", the processed ACE structure retrieved from PDB entry 1086; (2) "1086_ligand_co2.mol2", structure of the lead molecule; (3)

"1086_charged_1001_vs.mol2", the reference molecules, including a total of 1000 molecules randomly selected from the Available Chemical Directory database (ACD). Their binding poses inside the binding pocket on the target protein were also generated by the GOLD software (version 5.2, Cambridge Crystallography Data Center). The three best-ranked binding poses were kept for each ACD molecule, and thus there were ~3000 molecular structures in the reference library.

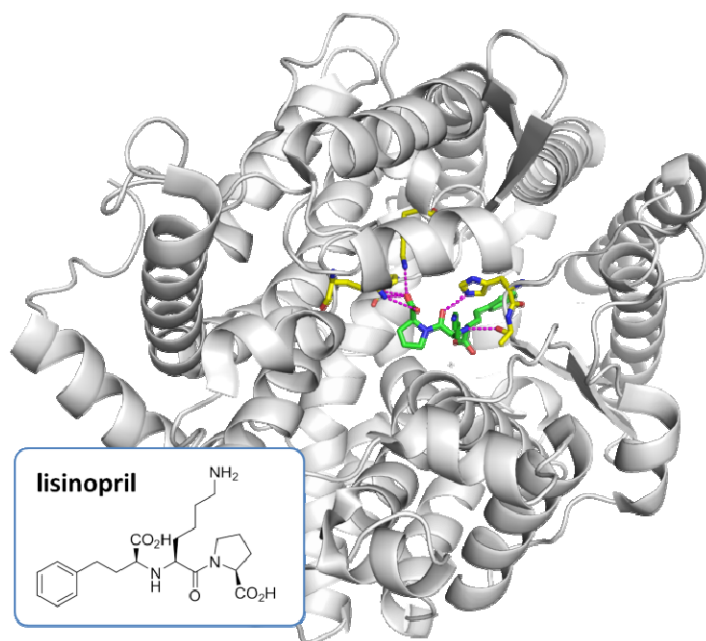


Figure 15. Structure of ACE (in gray ribbons) in complex with lisinopril (in green stick model) from PDB entry 1086.

A five-round optimization is performed by using AutoT&T2. The command line is:

```
LeadOpt2 -l 1086_ligand_co2.mol2 -p 1086.pdb -vs 1086_charged_1001_vs.mol2  
-i 5 -o ImproveATT_result.mol2
```

AutoT&T2 took 70 seconds to finish five rounds of optimization on our desktop workstation (Dell Precision T5610, dual Intel Xeon® E5-2609 v2 processors, Intel C602 motherboard, 16 GB DDR3 memory). It generated 8438 new ligand molecules in total. One can see that AutoT&T2 automatically ended at the fifth round because all possible matched bonds between the lead molecule and the reference molecules have already been examined thoroughly.

Table 11. Results generated by AutoT&T2 in a five-round optimization job

Round	CPU Time in second	Matched bonds under consideration	Number of generated molecules
1	13	614	56
2	5	938	844
3	23	4854	3443
4	29	6468	4095
5	0.3	0	0
Total	70	12874	8438

As the first test case, the user may also apply all the post-processing functions provided in the AutoT&T2 package to process the outputs given by AutoT&T2. The commands are similar and thus not repeated here.

5.3 Test case 3: Multi-thread optimization

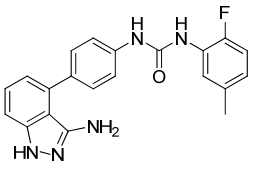
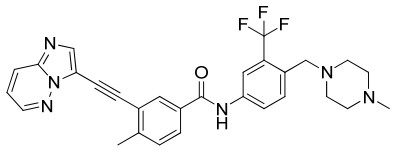
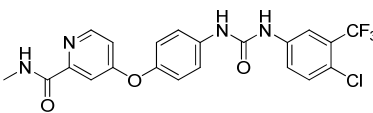
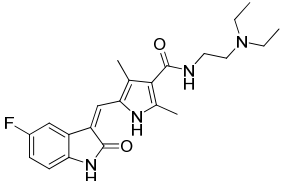
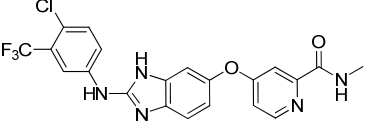
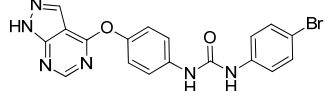
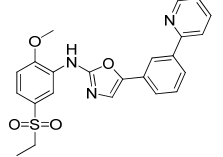
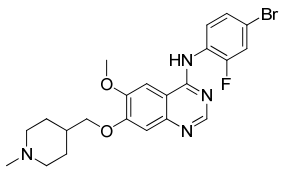
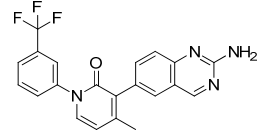
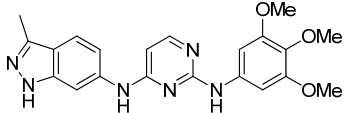
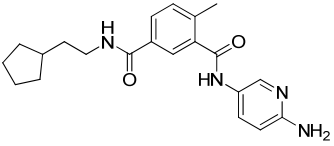
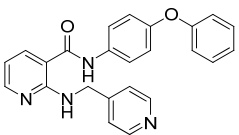
This test case is designed to demonstrate how AutoT&T2 conduct what we call "multi-thread optimization". In this running mode, AutoT&T2 can perform structural crossover among multiple given lead molecules. In fact, this is an effective approach frequently adopted by medicinal chemists to develop new active compounds based on known ones.

Vascular Endothelial Growth Factor Receptor 2 (VEGFR-2), an important anti-tumor drug target, is chosen as the molecular target in this test case. In order to assemble the inputs needed by AutoT&T2, we retrieved all VEGFR-2 inhibitors recorded in DrugBank v.4 [12] and PDBbind v.2014.[13] These VEGFR-2 inhibitors are 33 in total, including marketed drugs and drug candidates. We visually examined their chemical structures and selected 12 of them as the inputs for a multi-thread optimization job (Table 12). The others were excluded because they were structurally redundant to at least one of those selected ones, i.e. sharing a very similar structural scaffold and differing only by some substituent groups.

All of the necessary inputs for this test case can be found under the subdirectory "[examples/VEGFR2/](#)". (1) "[1ywn_protein.pdb](#)", the processed VEGFR-2 structure retrieved from PDB entry 1YWN; (2) "[mlead.mol2](#)", the 12 selected lead molecules. Binding poses of the 12 selected lead molecules were generated by using the GOLD software. For each lead molecule, five top-ranked binding poses were recorded. Thus, there were totally 60

molecular structures as the inputs for running this job.

Table 12. The twelve VEGFR-2 inhibitors used as inputs to a multi-thread optimization job.

 <p>Linifanib (DB06080^a)</p>	 <p>Ponatinib (DB08901, IC₅₀=95.6nM)</p>	 <p>Sorafenib (DB00398, IC₅₀=7nM)</p>
 <p>Sunitinib (DB01268, K_d=1.5nM)</p>	 <p>DB06938 (PDB entry 2QU5, K_i=8.7nM)</p>	 <p>CHEMBL2332847 (IC₅₀=83nM)</p>
 <p>DB07333 (PDB entry 1Y6B, IC₅₀=38nM)</p>	 <p>Vandetanib (DB05294)</p>	 <p>DB07528 (PDB entry 3CPC, IC₅₀=5μM)</p>
 <p>DB08519</p>	 <p>DB07537 (PDB entry 3CPB, IC₅₀=25μM)</p>	 <p>DB07183 (PDB entry 2P2I, IC₅₀=38nM)</p>

^a: Serial number in DrugBank.

A three-round multi-thread optimization is performed by using AutoT&T. The command line is:

LeadOpt2 -L mlead.mol2 -p 1ywn_protein.pdb -vs mlead.mol2 -o cross_3rd.mol2 -r -i 3

After three round multi-thread optimization, a total of 174 new ligands are generated. Then, the PLP scoring function is used to evaluate the binding affinities of these new ligands to VEGFR-2. The command line is:

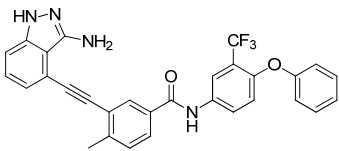
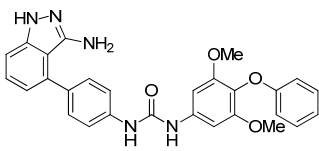
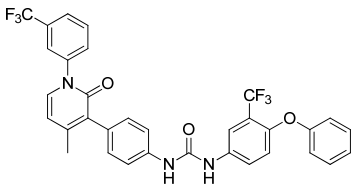
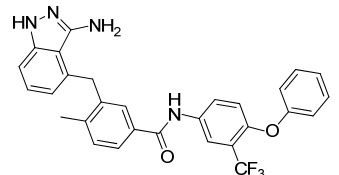
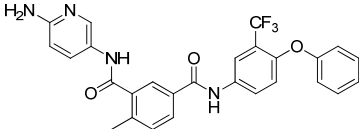
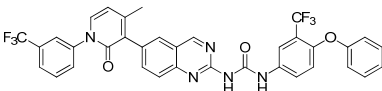
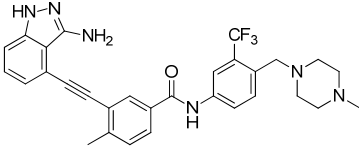
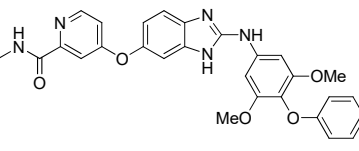
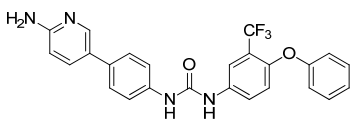
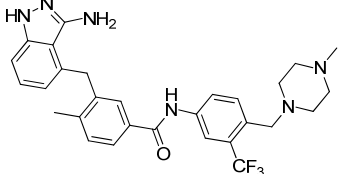
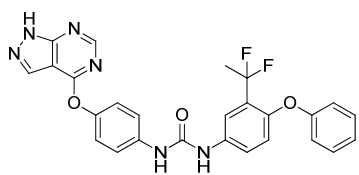
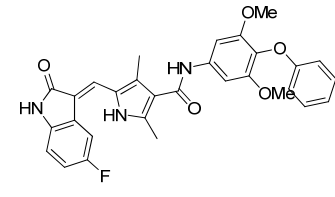
Score -l cross_3rd.mol2 -p 1ywn_protein.pdb -o score.log -s

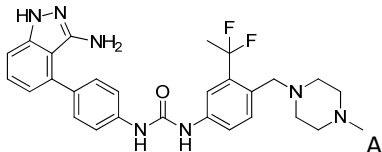
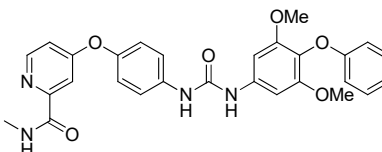
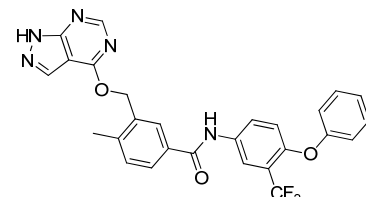
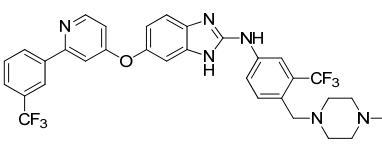
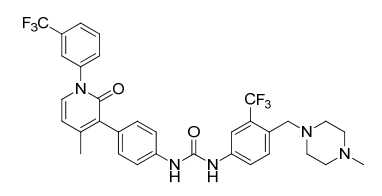
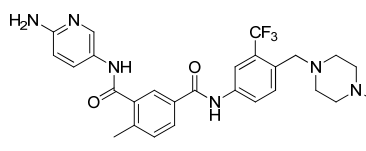
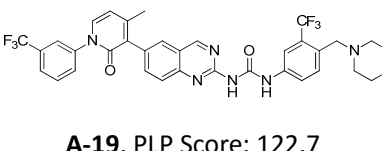
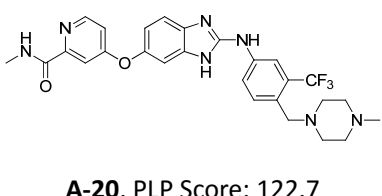
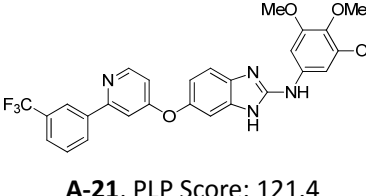
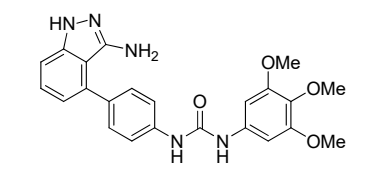
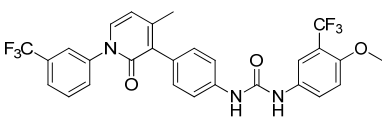
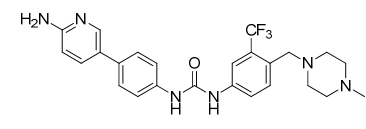
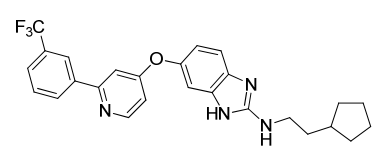
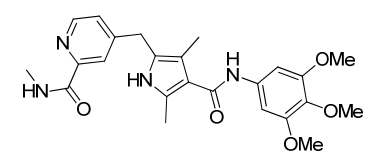
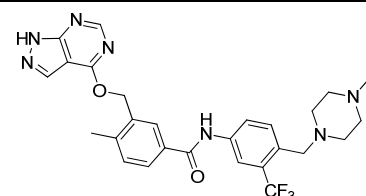
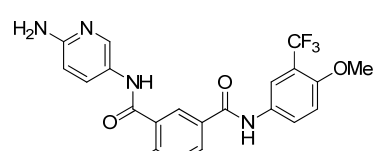
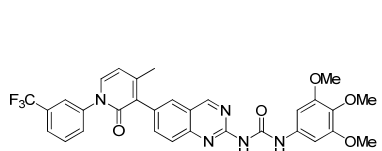
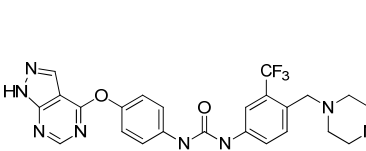
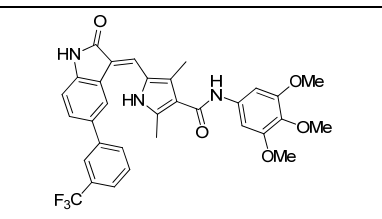
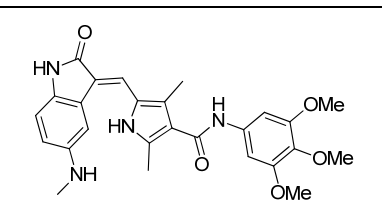
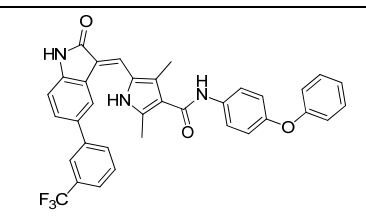
The ligands with their PLP scores above 100 are selected for discussion because the PLP score of the drug sorafenib to the protein is around this value. A total of 131 ligands generated by AutoT&T2 have their PLP scores above 100. In order to select the representative ones, these 131 selected molecules can be further clustered by their molecular framework. The command line for this task is:

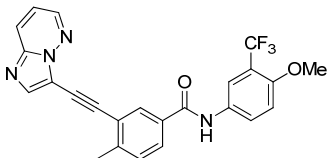
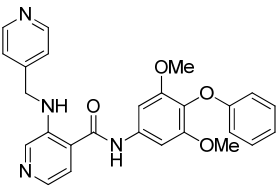
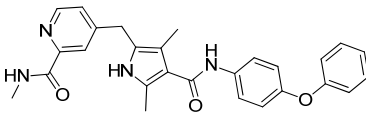
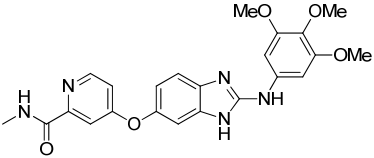
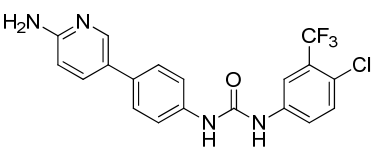
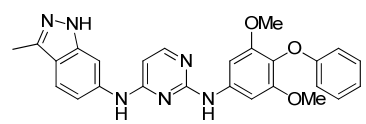
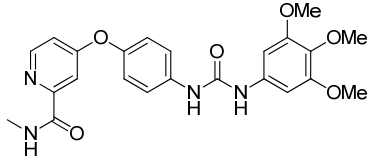
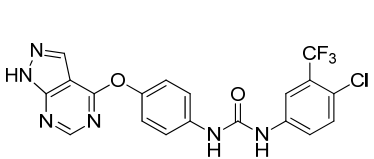
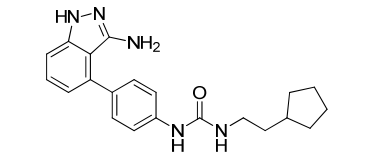
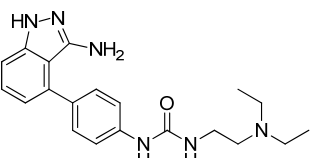
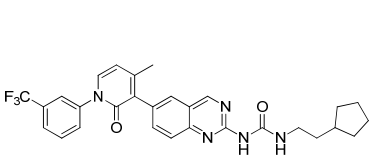
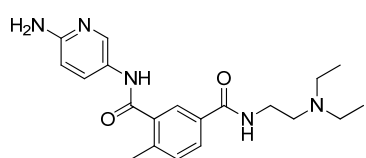
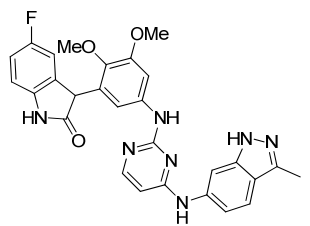
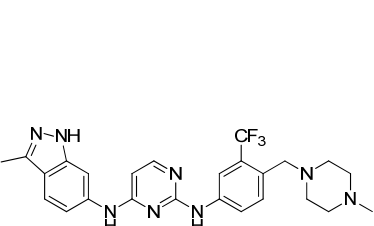
FrameworkCluster -i cross_select.mol2 -o clust.log -ap -sca 0.5

As result, these selected ligands can be further grouped into 47 molecular frameworks, which are summarized in Table 13 below.

Table 13. The representative ligand molecules generated by AutoT&T2 in this test case ^a

Selected ligand molecules given by AutoT&T2 ^b		
 A-1 , PLP Score: 140.8	 A-2 , PLP Score: 139.4	 A-3 , PLP Score: 138.9
 A-4 , PLP Score: 137.9	 A-5 , PLP Score: 135.7	 A-6 , PLP Score: 134.8
 A-7 , PLP Score: 133.0	 A-8 , PLP Score: 132.1	 A-9 , PLP Score: 132.0
 A-10 , PLP Score: 130.4	 A-11 , PLP Score: 129.2	 A-12 , PLP Score: 128.5

 <p>A-13, PLP Score: 127.5</p>	 <p>A-14, PLP Score: 127.4</p>	 <p>A-15, PLP Score: 127.1</p>
 <p>A-16, PLP Score: 126.1</p>	 <p>A-17, PLP Score: 125.4</p>	 <p>A-18, PLP Score: 124.1</p>
 <p>A-19, PLP Score: 122.7</p>	 <p>A-20, PLP Score: 122.7</p>	 <p>A-21, PLP Score: 121.4</p>
 <p>A-22, PLP Score: 120.1</p>	 <p>A-23, PLP Score: 120.1</p>	 <p>A-24, PLP Score: 119.9</p>
 <p>A-25, PLP Score: 119.7</p>	 <p>A-26, PLP Score: 119.0</p>	 <p>A-27, PLP Score: 118.2</p>
 <p>A-28, PLP Score: 117.1</p>	 <p>A-29, PLP Score: 117.1</p>	 <p>A-30, PLP Score: 116.6</p>
 <p>A-31, PLP Score: 114.1</p>	 <p>A-32, PLP Score: 113.9</p>	 <p>A-33, PLP Score: 113.7</p>

 <p>A-34, PLP Score: 113.5</p>	 <p>A-35, PLP Score: 113.4</p>	 <p>A-36, PLP Score: 113.3</p>
 <p>A-37, PLP Score: 113.2</p>	 <p>A-38, PLP Score: 111.4</p>	 <p>A-39, PLP Score: 110.9</p>
 <p>A-40, PLP Score: 108.8</p>	 <p>A-41, PLP Score: 107.9</p>	 <p>A-42, PLP Score: 106.5</p>
 <p>A-43, PLP Score: 105.8</p>	 <p>A-44, PLP Score: 104.6</p>	 <p>A-45, PLP Score: 103.2</p>
 <p>A-46, PLP Score: 102.5</p>	 <p>A-47, PLP Score: 102.8</p>	

5.4 Test case 4: Design of covalent binders

This test case is designed to demonstrate how AutoT&T2 can be applied to design of covalent binders to a given target. Development of covalent binders is an appealing approach in drug discovery. Some of the best-selling drugs are actually covalent binders.

The target protein selected in this test case is AmpC β -lactamase. A remarkable class of non- β -lactam inhibitors of serine β -lactamases all contain a boronic acid group.[14,15] A few examples are shown in Figure 16 below. Such a β -lactamase inhibitor forms a reversible

covalent bond between the boronic acid group and Ser64 inside the active site on β -lactamase.

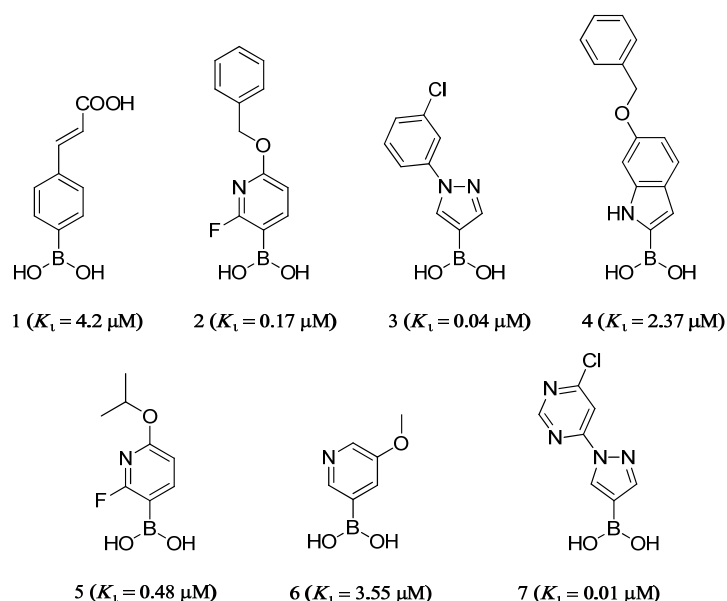


Figure 16. Chemical structures of several β -lactamase inhibitors containing a boronic acid group

All of the necessary inputs for this test case can be found under the subdirectory "[examples/AmpC/](#)". (1) "[1ke0_protein2.pdb](#)", the processed protein structure of AmpC β -lactamase, which is retrieved from PDB entry 1KE0. The Ser64 residue inside the binding pocket is designated to form the desired covalent bond. (2) "[lead1.mol2](#)", the lead molecule. Molecule 1 in Figure 16 was selected as the lead molecule for optimization. This molecule was placed manually at an appropriate position to form the desired covalent link between the boronic acid group and Ser64. Then, the bond connecting the boronic acid group with the rest part of the lead molecule was designated as the optimization site. (3) "[vs-specs.mol2](#)", the reference library, which consists of ~14000 molecules selected from SPECS catalog. These selected molecules were docked into the binding pocket on β -lactamase by using the GOLD software. Three top-ranked binding poses were saved for each molecule. Thus, the final reference library consisted of about 42,000 molecular structures.

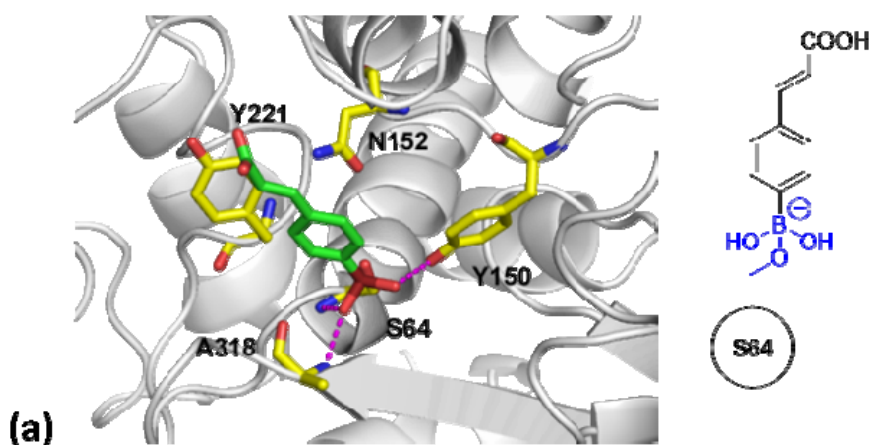
Since this job is about installing fragments on one particular site, one can use either [GrowLeadOpt](#) or [LeadOpt2](#). The "[-c](#)" flag is necessary to assign the bond connecting atom #1 and #4 as the desired optimization site. The fragment connecting to atom #1, i.e. the boronic acid group, will be retained; while the fragment connecting to atom 4# will be replaced. The command line is as follows:

LeadOpt2 -l lead1.mol2 -p 1ke0_protein2.pdb -vs vs-specs.mol2 -o result.mol2 -r -c (1,4)

It takes AutoT&T2 less than one minute to finish this job. A total of 182 new ligand molecules are generated and recorded in "result.mol2". Because the binding pocket of AmpC β -lactamase is relatively small, these molecules are filtered by the following rules: molecular weight ≤ 500 , hydrogen bond acceptor number ≤ 10 , hydrogen bond donor number ≤ 5 , rotatable bond number ≤ 10 , ring number ≤ 5 , log P value $\in [-2.0, 6.5]$, and heavy atom number $\in [5, 20]$. The command line for this task is:

Filter -i result.mol2 -o result_filter.mol2 -rf rule3.txt

As result, 37 molecules qualify for these filtering criteria, which are recorded in "result_filter.mol2". Some of the ligand molecules generated by AutoT&T2 have the boronic acid group installed directly on a five-member or six-member aromatic ring just as the known β -lactamase inhibitors shown in Figure 16. As example, binding modes of two of them (A-11 and A-14) as well as the lead molecule are shown in Figure 17. Both A-11 and A-14 may retain the hydrogen bond with Tyr150 as the lead molecule due to the common boronic acid group. But molecule A-11 may form additional hydrogen bonds between its triazole ring and Asn152 and Ser64. Moreover, the terminal substituent phenyl group on both molecules may form π - π stacking with Tyr221.



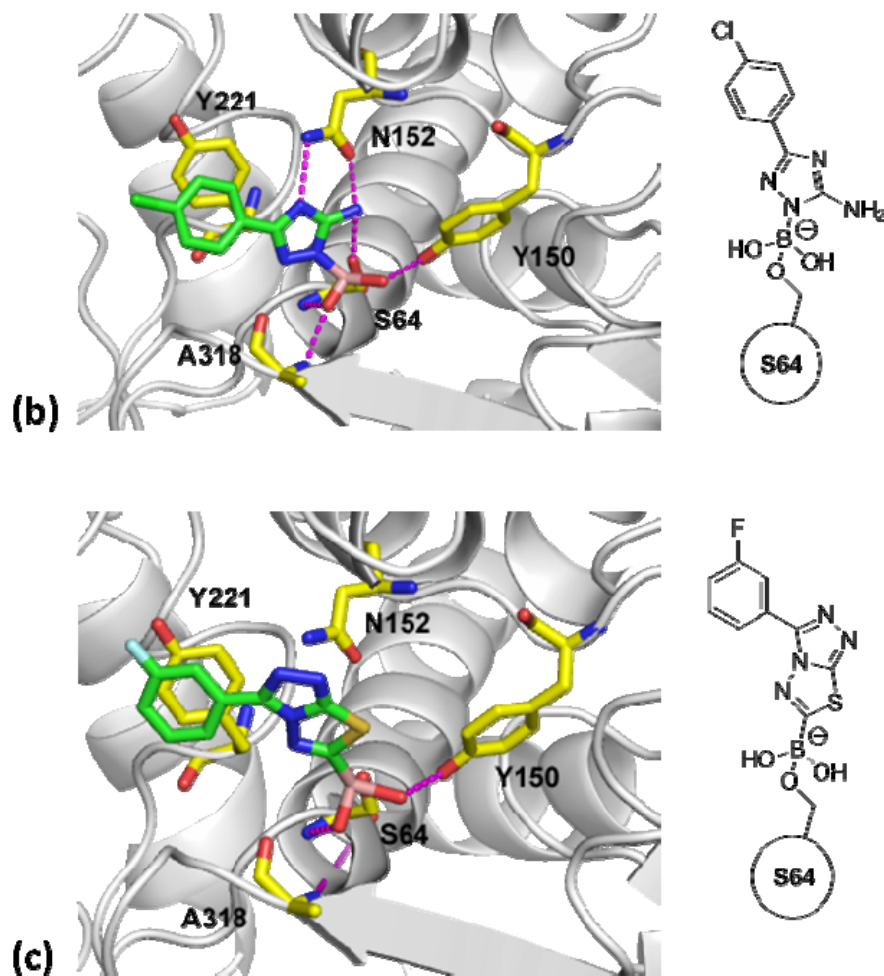


Figure 17. (a) Chemical structure and binding pose of a known AmpC β -lactamase inhibitor, which was used as the lead molecule in the third test case; (b) (c) Chemical structures and binding poses of **A-11** and **A-14**, which are two covalent ligands generated by AutoT&T2.

Copyright and Contact Information

The AutoT&T serial software is developed by Prof. Renxiao Wang's group at Shanghai Institute of Organic Chemistry, Chinese Academy of Sciences. All rights are reserved (Computer software registration numbers 2014SR135134, 2014SR135139, 2014SR135213, 2013SR160292, 2009SR061028, by National Copyright Administration of China).

If you encounter any technical problem in using AutoT&T, please contact us at:

Dr. Yan Li	kathyli@mail.sioc.ac.cn
Mr. Zhihai Liu	liuhai@mail.sioc.ac.cn
Prof. Renxiao Wang	wangrx@mail.sioc.ac.cn
State Key Lab of Bio-organic and Natural Products Chemistry Shanghai Institute of Organic Chemistry, Chinese Academy of Sciences. 345 Lingling Road, Shanghai 200032, China.	

To cite AutoT&T v.2:

Li, Y.; Zhang, Z.; Liu, Z.; Wang, R. AutoT&T v.2: An Efficient and Versatile Tool for Lead Structure Generation and Optimization, *J. Chem. Inf. Model.* (in revision)

To cite AutoT&T v.1:

Li, Y.; Zhao, Y.; Liu, Z.; Wang, R. Automatic tailoring and transplanting: A practical method that makes virtual screening more useful. *J. Chem. Inf. Model.* **2011**, *51*, 1474-1491.

References

- [1] Li, Y.; Zhao, Y.; Liu, Z.; Wang, R. Automatic Tailoring and Transplanting: A Practical Method that Makes Virtual Screening More Useful. *J. Chem. Inf. Model.* **2011**, *51*, 1474-1491.
- [2] Clark, M.; Cramer, R. D.; Van Opdenbosch, N. Validation of the general purpose Tripos 5.2 force field. *J. Comput. Chem.* **1989**, *10*, 982-1012.
- [3] Case, D. A.; Darden, T. A.; Cheatham, T. E.; Simmerling, C. L.; Wang, J.; Duke, R. E.; Luo, R.; Merz, K. M.; Pearlman, D. A.; Crowley, M.; Walker, R. C.; Zhang, W.; Wang, B.; Hayik, S.; Roitberg, A.; Seabra, G.; Wong, K. F.; Paesani, F.; Wu, X.; Brozell, S.; Tsui, V.; Gohlke, H.; Yang, L.; Tan, C.; Mongan, J.; Hornak, V.; Cui, G.; Beroza, P.; Mathews, D. H.; Schafmeister, C.; Ross, W. S.; Kollman, P. A. *AMBER 9*, University of California, San Francisco, 2006.
- [4] Verkhivker, G.; Appelt, K.; Freer, S.T.; Villafranca, J.E. Empirical free energy calculations of ligand-protein crystallographic complexes. I. Knowledge-based ligand-protein interaction potentials applied to the prediction of human immunodeficiency virus 1 protease binding affinity. *Protein Eng.* **1995**, *8*, 677-691.
- [5] Rogers, D.; Brown, R. D.; Hahn, M. Using extended-connectivity fingerprints with Laplacian-modified Bayesian analysis in high-throughput screening follow-up. *J. Biomol. Screen.* **2005**, *10*, 682-686.
- [6] Cheng, T.; Zhao, Y.; Li, X.; Lin, F.; Xu, Y.; Zhang, X.; Li, Y.; Wang, R.; Lai, L. Computation of octanol-water partition coefficients by guiding an additive model with knowledge. *J. Chem. Inf. Model.* **2007**, *47*, 2140-2148.
- [7] Lewell, X. Q.; Judd, D. B.; Watson, S. P.; Hann, M. M. RECAP-retrosynthetic combinatorial analysis procedure: a powerful new technique for identifying privileged molecular fragments with useful applications in combinatorial chemistry. *J. Chem. Inf. Comput. Sci.* **1998**, *38*, 511-522.
- [8] Sheridan, R. P.; Miller, M. D.; Underwood, D. J.; Kearsley, S. K. Chemical similarity using geometric atom pair descriptors. *J. Chem. Inf. Comput. Sci.* **1996**, *36*, 128-136.
- [9] Nilakantan, R.; Bauman, N.; Dixon, J. S.; Venkataraghavan, R. Topological torsion: a new molecular descriptor for SAR applications. Comparison with other descriptors. *J. Chem. Inf. Comput. Sci.* **1987**, *27*, 82-85.
- [10] Likas, A.; Vlassis, N.; Verbeek, J. J. The global k-means clustering algorithm. *Pattern Recognit.* **2003**, *36*, 451-461.
- [11] Reynolds, C. H.; Druker, R.; Pfahler, L. B. Lead discovery using stochastic cluster analysis (SCA): a new method for clustering structurally similar compounds. *J. Chem. Inf. Comput.*

- Sci.* **1998**, *38*, 305-312.
- [12] Law, V.; Knox, C.; Djoumbou, Y.; Jewison, T.; Guo, A. C.; Liu, Y.; Maciejewski, A.; Arndt, D.; Wilson, M.; Neveu, V.; Tang, A.; Gabriel, G.; Ly, C.; Adamjee, S.; Dame, Z. T.; Han, B.; Zhou, Y.; Wishart, D. S. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucleic Acids Res.* **2014**, *42*, D1091-D1097.
- [13] Liu, Z.; Li, Y.; Han, L.; Li, J.; Liu, J.; Zhao, Z.; Nie, W.; Liu, Y.; Wang, R. PDB-wide Collection of Binding Data: Current Status of the PDBbind Database. *Bioinformatics*, **2015**, *31*, 405-412.
- [14] London, N.; Miller, R. M.; Krishnan, S.; Uchida, K.; Irwin, J. J.; Eidam, O.; Gibold, L.; Cimermancic, P.; Bonnet, R.; Shoichet, B. K.; Taunton, J. Covalent docking of large libraries for the discovery of chemical probes. *Nat. Chem. Biol.* **2014**, *10*, 1066-1072.
- [15] Tondi, D.; Calo, S.; Shoichet, B. K.; Costi, M. P. Structural study of phenyl boronic acid derivatives as AmpC β -lactamase inhibitors. *Bioorg. Med. Chem. Lett.* **2010**, *20*, 3416-3419.