

分子对接

——使用 AutoDock 和 AutoDock Tools

一、分子对接简介及软件介绍

二、对接准备及对接操作

三、结果分析

一、分子对接简介及软件介绍

1. 分子对接理论基础

所谓分子对接就是两个或多个分子之间通过几何匹配和能量匹配而相互识别的过程。分子对接在酶学研究以及药物设计中具有十分重要的意义。在酶激活剂、酶抑制剂与酶相互作用以及药物分子产生药理反应的过程中，小分子（通常意义上的Ligand）与靶酶（通常意义上的Receptor）相互结合，首先就需要两个分子充分接近，采取合适的取向，使两者在必要的部位相互契合，发生相互作用，继而通过适当的构象调整，得到一个稳定的复合物构象。通过分子对接确定复合物中两个分子正确的相对位置和取向，研究两个分子的构象，特别是底物构象在形成复合物过程中的变化，是确定酶激活剂、抑制剂作用机制以及药物作用机制，设计新药的基础。

分子对接计算是把配体分子放在受体活性位点的位置，然后按照几何互补、能量互补化学环境互补的原则来实时评价配体与受体相互作用的好坏，并找到两个分子之间最佳的结合模式。分子对接最初思想起源于Fisher E.的“锁和钥匙模型”（图），认为“锁”和“钥匙”的相识别的首要条件是他们在空间形状上要互相匹配。然而，配体和受体分子之间的识别要比“锁和钥匙”模型复杂的多。首先，配体和受体分子的构象是变化的，而不是刚性的，配体和受体在对接过程中互相适应对方，从而达到更完美的匹配。其次，分子对接不但要满足空间形状的匹配，还要满足能量的匹配。配体和受体之间的通过底物分子与靶酶分子能否结合以及结合的程度最终是由形成此复合物过程的结合自由能变化 ΔG_{bind} 所决定的。

互补性（complementarity）和预组织（pre-organization）是决定分子对接过程的两个重要原则，前者决定识别过程的选择性，而后者决定识别过程的结合能力。互补性包括空间结构的互补性和电学性质的互补性。1958年Koshland提出了分子识别过程中的诱导契合（induced fit）概念，指出配体与受体相互结合时，受体将采取一个能同底物达到最佳结合的构象（图1）。而受体与配体分子在识别之前将受体中容纳配体的环境组织的越好，其溶剂化能力越低，则它们的识别效果越佳，形成的复合物也就越稳定。

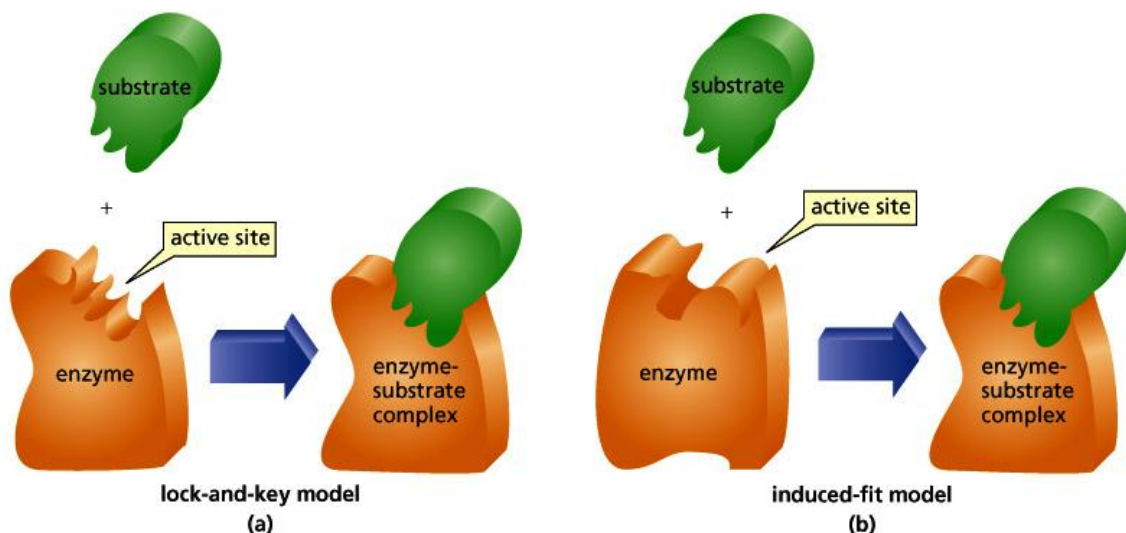


图 1: “锁和钥模型 (a)” 和 “诱导契合模型 (b)” 示意图

2. 分子对接常用方法

分子对接方法根据不同的简化程度可以大致分为以下三类：**(1) 刚性对接**；**(2) 半柔性对接**；**(3) 柔性对接**。刚性对接指在对接的过程中，研究体系的构象不发生变化；半柔性对接是指在对接过程中，研究体系尤其是配体的构象允许在一定的范围内变化；而柔性对接则是指在对接过程中，研究体系的构象基本上是可以自由变化的。当然，这只是一种简单的分类方法，而在很多分子对接程序中，实际上采取了多种处理方法。在这些分子对接方法中，刚性对接适合考察比较大的体系，比如蛋白质和蛋白质以及蛋白质和核酸之间的相互作用，他计算较为简单，原理也相对简单，主要是考虑构象之间的契合程度。半柔性对接适合于处理小分子和大分子之间的对接。在对接过程中，小分子的构象一般是可以变化的，但大分子则是刚性的。由于小分子相对较小，因此在一定程度考察柔性的基础上，还可以保持较高的计算效率，在药物设计尤其在基于分子数据库的虚拟筛选过程中，一般采用半柔性的分子对接方法。柔性对接方法一般用于精确考察分子之间的识别情况，由于在计算过程中体系的构象是可以变化的，因此柔性对接在提高了对接准确性的同时却需要耗费较长的计算时间。

上述各种几何优化方法可以获得分子对应与初始态的优势构象，但实际上这样的构象可以有很多，一般认为自由能最小的构象存在的概率高，全局极小可能是比较重要的构象。同时，配体和受体活性部位结合时的构象**不一定是全局极小构象，而可能是某一低能构象（药效构象）**。不管是寻找分子的全局极小构象还是药效构象，均要使用构象搜索方法。

分子对接的目的是找到底物分子和受体分子的最佳结合位置。因此，分子对接面临的最重要的问题就是如何找到最佳的结合位置以及如何评价对接分子之间的结合强度。当然，这两个问题也是相互关联的。如何找到最佳的结合位置就要牵涉到构象搜索方法。常用的构象搜索方法有系统搜索

法和非系统搜索法。系统搜索法通过系统的改变每一个扭转角产生所有可能的构象，从中挑选出能量较低的构象，但计算量非常大。所以通常使用非系统搜索法来寻找能量较低构象，常用方法有：

(1) 分子动力学方法 (Molecular dynamics, MD)；(2) 随机搜索 (Random search)；(3) 遗传算法 (Genetic algorithm, GA)；(4) 距离几何算法 (Distance geometry, DG) 等。随机搜索又包含：(1) 完全随机算法；(2) 蒙特卡罗法 (Monte Carlo, MC)；(3) 模拟退火法 (Simulated annealing, SA)。分子对接的方法很多，下表 (表 1) 列出了针对不同对接体系的常用对接方法：

表 1: 常用的分子对接方法

对接类型	对接方法类型	对接方法
Flexible-Ligand Docking 柔性配体对接	Systematic 系统方法	Conformational 构象
		Fragmentation 片段生长
		Database 数据库
	Random/stochastic 随机方法	Monte Carlo (MC) 蒙特卡罗
		Genetic algorithm (GA) 遗传算法
		Tabu Search 禁忌搜索
	Simulation methods 模拟方法	Molecular dynamics (MD) 分子动力学
Energy minimization 能量最小化		
Flexible-Protein Docking 柔性蛋白对接	Molecular dynamics (MD) 分子模拟	
	Monte Carlo (MC) 蒙特卡罗	
	Rotamer libraries 旋转异构体库	
	Protein-ensemble grids 蛋白集合栅格	
	Soft-receptor modeling 软受体建模	

3. 分子对接软件

第一个分子对接程序是 UCSF Kuntz 小组于 1982 年开发的 DOCK，早期的版本以刚性对接为主，从 4.0 版开始考虑配体的柔性。像这样的半柔性 (刚性受体—柔性配体: rigid receptor-flexible ligand docking) 对接程序还有 AutoDock、FlexX 等。同时考虑配体和受体柔性的对接程序主要有 FlexiDock，它采用遗传算法来对配体和受体的结合构象进行优化，在初始条件较好的情况下，FlexiDock 可以比较精确的确定配体和受体的结合状态，但计算时间较长。下表为一些有代表性的分子对接软件 (表 2) 及软件的被引次数对比 (图 2)。

表 2: 有代表性的分子对接软件

名称	优化方法	评价函数	速度	Protein-protein (peptide) docking	Protein-ligand docking
DOCK	片断生长	分子力场、表面匹配得分、 化学环境匹配得分	快		✓
AutoDock	遗传算法	半经验自由能评价函数	一般		✓
ICM-Docking	随机全局优化	半经验自由能评价函数	快		✓
GOLD	遗传算法	半经验自由能评价函数	快		✓

FlexX	片断生长	半经验自由能评价函数	快		✓
Affinity	蒙特卡罗/分子力学/分子动力学	分子力场	慢		✓
ZDock&RDock	几何匹配/分子动力学	CAPRI* /分子力场	慢	✓	
FlexiDock	遗传算法	分子力场	慢		✓
eHiTS	系统搜索	半经验自由能评价函数	快		✓
Hex	几何匹配	CAPRI*	快	✓	

注：CAPRI, Critical Assessment of PRediction of Interactions, 相互作用预测临界评价

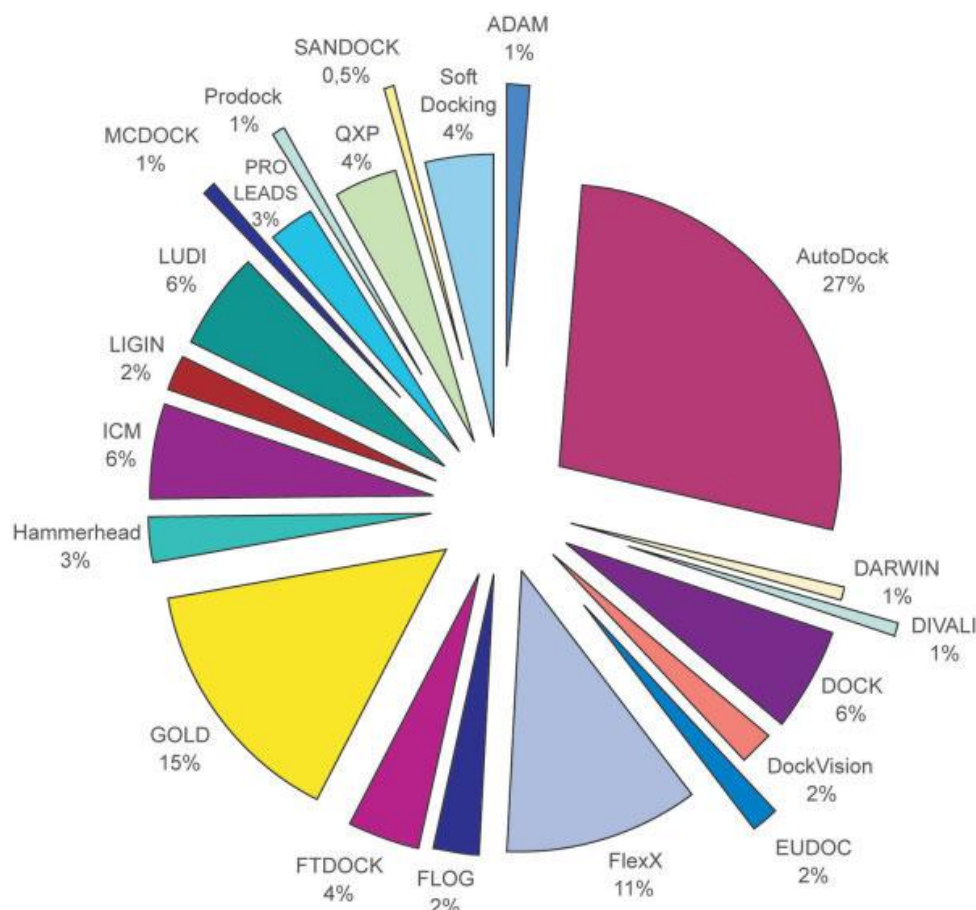


图 2: 分子对接软件——最常用的分子对接软件被引次数统计 (ISI Web of Science, 2005)

4. AutoDock 以及 ADT (AutoDock Tools)

4.1. AutoDock

AutoDock 是 The Scripps Research Institute 的 Olson 科研小组使用 C 语言开发的分子对接软件包, 目前最新的版本为 4.01。AutoDock 其实是一个软件包, 其中主要包含 AutoGrid 和 AutoDock 两个程序。其中 AutoGrid 主要负责格点中相关能量的计算, 而 AutoDock 则负责构象搜索及评价。

AutoDock 在早期版本中使用的是模拟退火算法 (Simulated Annealing Algorithm) 来寻找配体与受体最佳的结合位置状态, 而从 3.0 版本开始使用一种改良的遗传算法, 即拉马克遗传算法 (Lamarckian Genetic Algorithm, LGA)。测试结果表明, LGA 比传统的遗传算法和模拟退火具有

更高的效率。在 LGA 方法中，作者把遗传算法和局部搜索（Local search）结合在一起，遗传算法用于全局搜索，而局部搜索用于能量优化。LGA 算法引入了拉马克的遗传理论，这个操作过程可下图（图 3）。

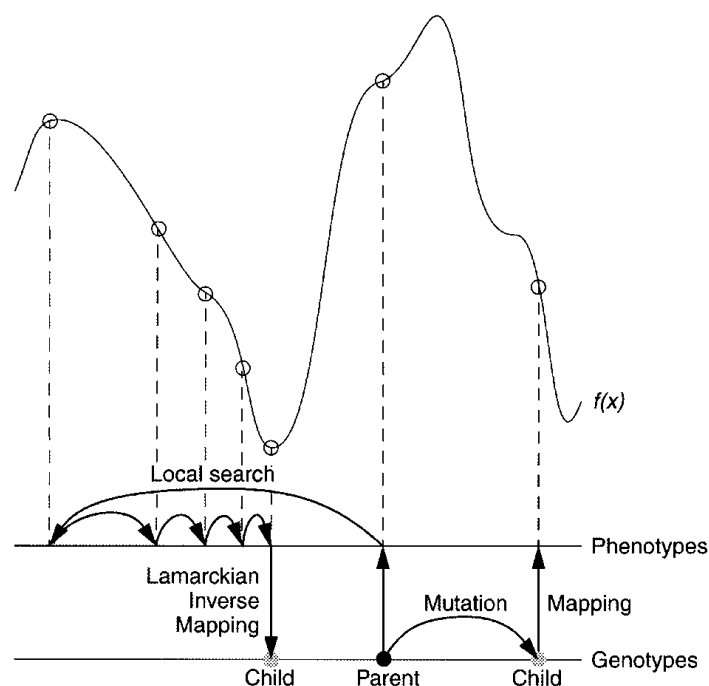


图 3: LGA 算法操作过程图

同时在 AutoDock 中配体和受体之间结合能力采用能量匹配来评价。在 1.0 和 2.0 版本中，能量匹配得分采用简单的基于 AMBER 力场的非键相互作用能。非键相互作用来自于三部分的贡献：范德华相互作用，氢键相互作用以及静电相互作用。而在 3.0 之后的版本中 AutoDock 提供了半经验的自由能计算方法来评价受体和配体之间的能量匹配。

为了加快计算速度，AutoDock 采用格点对接的方法，但与 DOCK 中格点对接的处理方法有明显的区别。DOCK 中，格点上保存的不是能量，而是仅与受体有关的特征量。而在 AutoDock 中，格点上保存的是探针原子和受体之间的相互作用能。

对于范德华相互作用的计算，每个格点上保存的范德华能量的值的数目与要对接的配体上的原子类型（表 3）的数目一样。如果一个配件中含有 C、O 和 H 三种原子类型，那么在每个格点上就需要用三个探针原子来计算探针原子与受体之间的范德华相互作用值。当配体和受体进行分子对接时，配体中某个原子和受体之间的相互作用能通过周围 8 个格点上的这种原子类型为探针的格点值用内插法得到。

表 3: AutoDock4 中的原子类型 (*为默认在 gpf 中存在的原子类型)

Atom Type	
H	Non H-bonding Hydrogen
HD*	Donor 1 H-bond Hydrogen

HS	Donor S Spherical Hydrogen
C*	Non H-bonding Aliphatic Carbon
A*	Non H-bonding Aromatic Carbon
N*	Non H-bonding Nitrogen
NA*	Acceptor 1 H-bond Nitrogen
NS	Acceptor S Spherical Nitrogen
OA*	Acceptor 2 H-bonds Oxygen
OS	Acceptor S Spherical Oxygen
F	Non H-bonding Fluorine
Mg	Non H-bonding Magnesium
MG	Non H-bonding Magnesium
P	Non H-bonding Phosphorus
SA*	Acceptor 2 H-bonds Sulphur
S	Non H-bonding Sulphur
Cl	Non H-bonding Chlorine
CL	Non H-bonding Chlorine
Ca	Non H-bonding Calcium
CA	Non H-bonding Calcium
Mn	Non H-bonding Manganese
MN	Non H-bonding Manganese
Fe	Non H-bonding Iron
FE	Non H-bonding Iron
Zn	Non H-bonding Zinc
ZN	Non H-bonding Zinc
Br	Non H-bonding Bromine
BR	Non H-bonding Bromine
I	Non H-bonding Iodine

静电相互作用的计算采用了一个静电势格点，在格点上储存受体分子的静电势。当配体和受体分子对接时，某个原子和受体之间的静电相互作用能通过周围格点上静电势以及原子上的部分电荷就可以计算得到。

计算氢键相互作用时，格点的处理和范德华相互作用有点类似，每个格点上需要保存配体分子中所有氢键给体与氢键受体之间的相互作用能量，而且这些能量都是在氢键在最佳情况下的氢键能量值。

以上格点能量的计算都是由 AutoDock 中的 AutoGrid 程序计算得出的，AutoDock 格点对接示意图如下图所示（图 4）。AutoDock 格点对接的基本流程如下：首先，用围绕受体活性位点的氨基酸残基形成一个范围更大的 Box，然后用不同类型的原子作为探针（probe）进行扫描，计算格点能量，此部分任务由 AutoGrid 程序完成。然后 AutoDock 程序对配体在 Box 范围内进行构象搜索（conformational search），最后根据配体的不同构象(conformation)，方向（orientation）、位置

(position) 及能量 (energy) 进行评分 (scoring), 最后对结果进行排序 (ranking)。

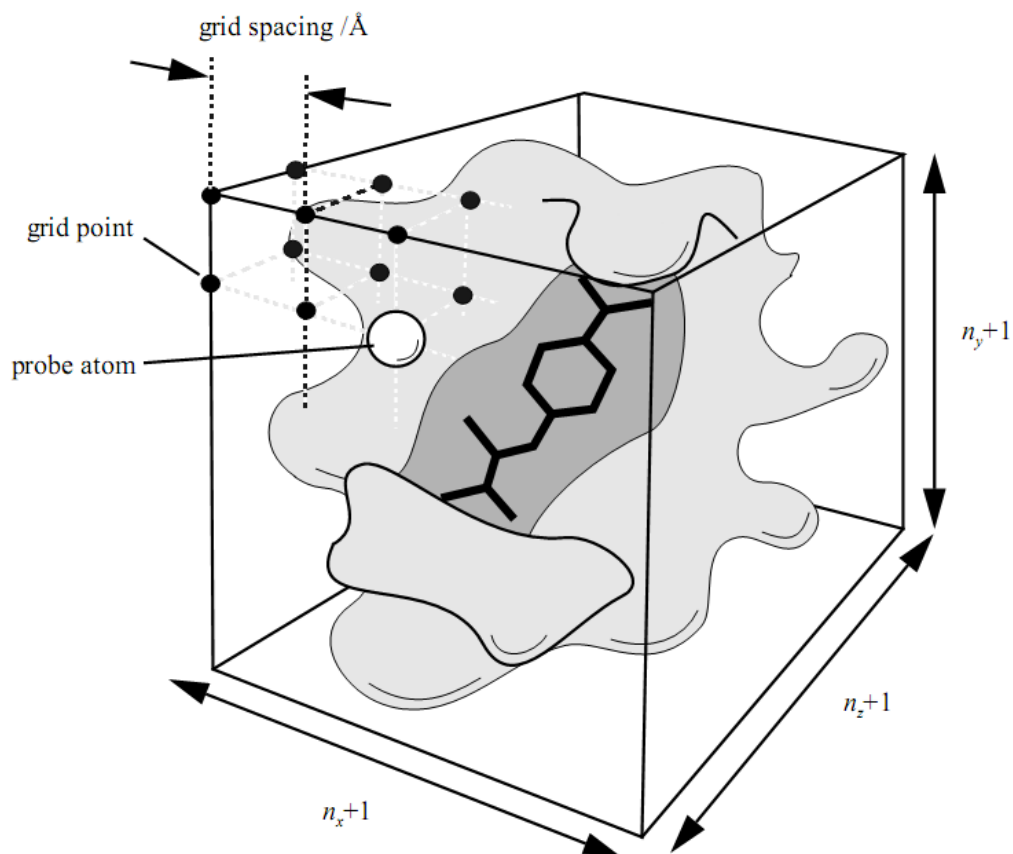
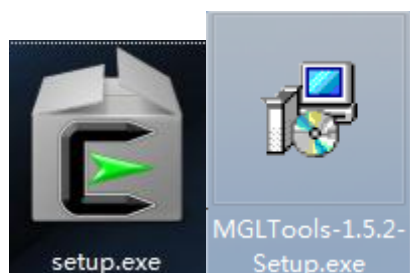


图 4: AutoDock 格点对接示意图

AutoDock 目前的版本只能实现单个配体和受体分子之间的对接, 程序本身还没有提供虚拟筛选功能 (Virtual Screening), 但是可以使用 Linux/Unix 中的 Shell 以及 Python 语言实现此功能。同时 AutoDock 本省所包含的 AutoDock 以及 AutoGrid 程序是完全在命令附下操作的软件, 没有图形界面, 但是如果使用 AutoDock Tools 程序, 就可以在几乎完全图形化的界面中完成分子对接以及结果分析等工作, 下面我们就介绍一下 AutoDock Tools。

软件安装



一) 下载并安装 **cygwin**.

1.网址: <http://www.cygwin.com>, 点击 setup.exe 即可下载。

2.双击图标即可安装,要在线安装,因为安装过程中要下载比较大的程序,默认安装路径为 C:\cygwin,在选择下载网址时,我选择了第一个网址,安装很顺利。

(二) 下载并解压 **autodocksuite-4.2.1-i86Cygwin.tar**

- 1.网址: <http://autodock.scripps.edu/downloads>
- 2.选择 AutoDock 4.2 Release Candidate 1 一列后面的 registration form 进行注册,完成后,点击 Windows / Cygwin 下载,自动命名为 autodocksuite-4.2.1-i86Cygwin.tar
- 3.解压缩至:\cygwin\usr\local\bin 文件夹,共有二个文件 Autodock4 和 Autogrid4。

(三) 下载并安装 **MGLTools-1.5.2-Setup**

- 1.网址: <http://mgltools.scripp... 5.2-Setup.exe>, 自动命名为 MGLTools-1.5.2-Setup.exe
- 2.双击图标进行安装,默认安装路径 C:\Program Files\MGLTools 1.5.2。
- 3.安装过程在线完成,会自动下载 Python2.5, 并进行安装,一路缺省设置即可,默认路径为 C:\phthon2.5。

(四) 下载并解压 MGLToolsPckgs.zip (MGL Tools 1.5.2 补充压缩包)

- 1.网址: <http://mgltools.scripp... oolsPckgs.zip>
- 2.解压至:\Python25\Lib\site-packages

安装完毕。

从开始>所有程序> MGLTools-1.5.2 即可运行 ADT。运行 ADT 只能双击在 C:\Python2.5\lib\site-packages\megltoolspckgs\autodocktools\bin 下的 runAdt.py,

4.2.AutoDock Tools

AutoDock Tools (以下简称 ADT) 是 The Scripps Research Institute, Molecular Graphics Laboratory (MGL)在 Python Molecular Viewer (以下简称 PMV, Python 语言开发)基础上开发的针对 AutoGrid 和 AutoDock 程序开发的图形化的分子可视化及对接辅助软件,目前最新版本为 1.5.2。在这里我们使用的版本为 1.5.1, 它的主界面主要包含以下几个部分 (图 5):

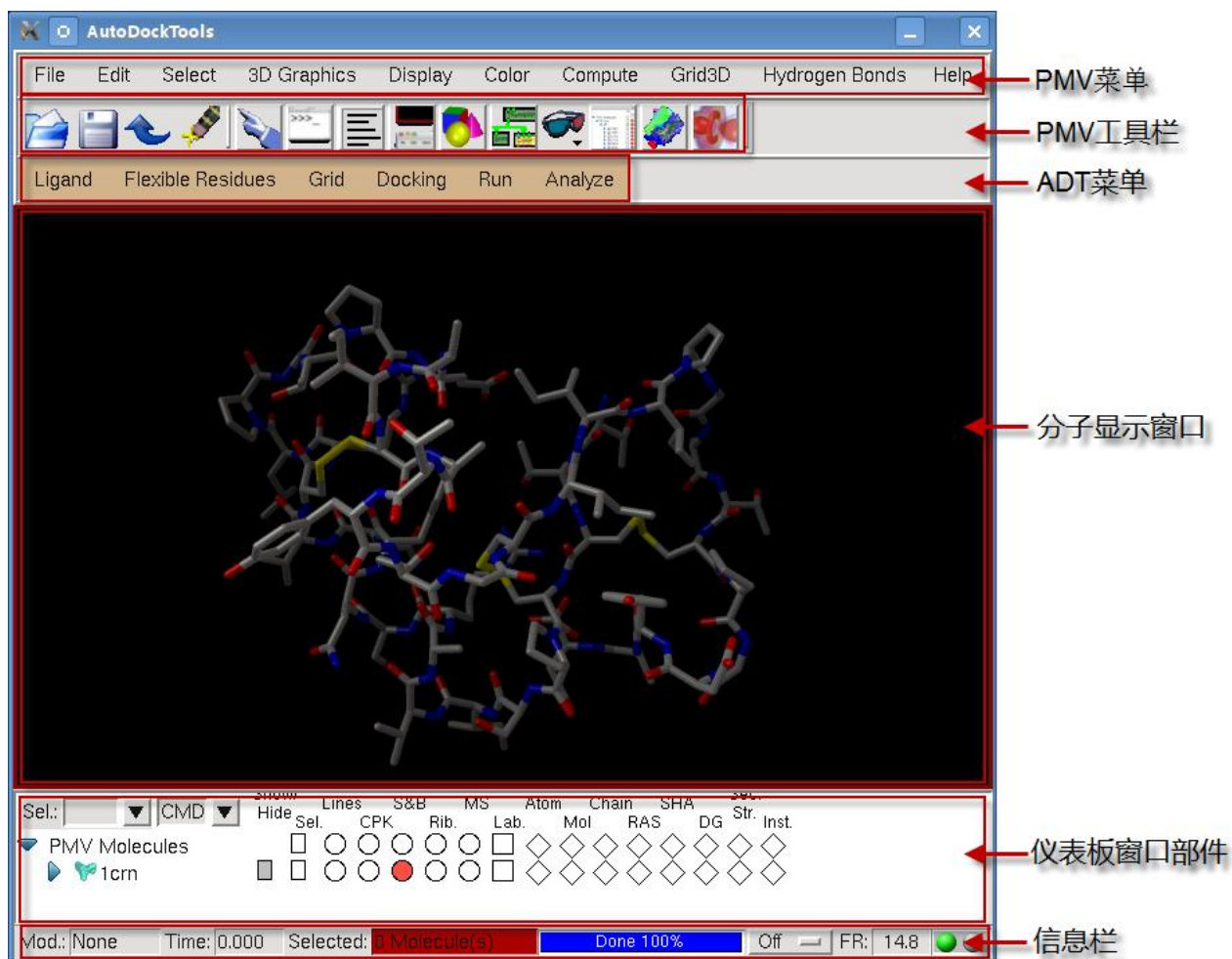


图 5: ADT1.5.1 的主界面及窗口部件

- (1) PMV 菜单: 主要通过使用菜单命令对分子进行相关的操作, 以及进行可视化设置;
- (2) PMV 工具栏: PMV 菜单中一些常用命令的快捷按钮;
- (3) ADT 菜单: AutoGrid 和 AutoDock 的图形化操作菜单;
- (4) 分子显示窗口: 3D 模型分子的显示和操作窗口;
- (5) 仪表板窗口部件: 快速查看及设置分子的显示模型以及着色方式;
- (6) 信息栏: 显示相关操作信息。

4.3.软件的获取及安装

AutoDock 程序包自版本 4.0 起成为自由软件 (Free Soft, 非免费软件), 只需在官方网站上完成注册 (<http://autodock.scripps.edu/downloads/autodock-registration>), 即可下载包含完整源代码的版本以及针对各种不同操作系统平台编译好的程序。

在这里我们下载包含源代码以及编译好的各种平台程序的完整软件包, 名称为

“**autodocksuite-4.0.1-all.tar.gz**”, 解压后该文件包包含三个目录: (1) “**bin**” 目录中是针对不同

平台编译好的 AutoGrid 以及 AutoDock 程序；(2) “**example**” 目录中包含一些程序运行的例子，可以进行软件测试以及相关操作的学习；(3) “**src**” 目录为 AutoGrid 以及 AutoDock 程序的源代码，可以自行编译成可执行的程序文件。“**bin**” 目录中编译好的针对不同平台的程序如下表所示（表 4）：

表 4: autodocksuite-4.0.1-all.tar.gz 中各种编译好的 AutoDock 程序

名称	硬件系统及操作系统平台
i86Cygwin	Intel x86 处理器, Window 系统 (使用 Cygwin)
i86Darwin8	Intel x86 处理器, Mac 操作系统
i86Linux2	Intel x86 处理器, Linux 操作系统
ia64Linux2	Intel 安腾 64 位处理器, Linux 操作系统
ppcDarwin8	IBM Power 处理器, Mac 操作系统
sgi4DIRIX646	Sgi 图形工作站, IRIX 操作系统
sun4SunOS5	Sun 服务器, Sun 操作系统

绝大多数情况下，这些编译好的程序在相应的平台下均能正确运行，但如碰上特殊情况或是针对自己的实际要求对程序进行了修改，那就需要重新编译 AutoGrid 以及 AutoDock 程序。下面就以 Linux 系统为例解释编译过程：

首先需要确认 Linux 中已经安装了 C 或 C++ 开发工具（如：gcc, g++ 等）以及 Make 工具等，一般情况下默认安装绝大多数版本的 Linux 操作系统这些个工具都会安装。之后打开控制台：

```
tar -xzf autodocksuite-4.0.1-all.tar.gz #解压缩软件包

cd autodocksuite-4.0.1/src/autogrid-4.0.0 #切换到 autogrid-4.0.0 目录

./configure #设置编译环境

make #编译 autogrid, 完成后此目录中会有 autogrid4 程序生成

cd ../autodock-4.0.1 #切换到 autodock-4.0.1 目录

./configure #设置编译环境

make #编译 autodock, 完成后此目录中会有 autodock4 程序生成
```

完成编译后，将 “**autogrid4**” 以及 “**autodock4**” 程序拷贝到需要的目录即可。

相比 AutoDock 的编译，ADT 的安装就显得尤其简单：首先，到 ADT 的官方网站下载网站（<http://mgltools.scripps.edu/downloads>）下载相应操作系统的版本，在这里我们下载 Linux 下的 Standalone installer 的版本：MGLTools-1.5.1-Linux-x86-Install（需要 GLIBC_2.3, libstdc++.5.X 才能正常运行）。下载完毕后，直接双击程序文件即可在 Linux

下进行完全图形化的安装，与 Windows 下安装程序一样。但是，如果 MGLTools-1.5.1-Linux-x86-Install 不能运行，则下载 `mglttools_i86Linux2_1.5.2.tar.gz`，解压缩后运行 `install.sh` 进行安装，一样会出现安装界面。需要注意的是目前 ADT 最新版本为 1.5.2，但安装方式完全一样，1.5.2 版本主要增加了针对不同版本 AutoDock 程序的切换功能，与 AutoDock 的兼容性更好。

二、对接准备及对接操作

在这一部分中我们将采用非常经典的 HIV 蛋白及其抑制剂（PDB ID: 1HSG）来作为例子对 AutoDock 的整个操作分析过程做一个详细的讲解。

以下操作过程全部在 ubuntu 8.04 操作系统下完成，桌面系统为 KDE4.0；AutoDock 为在此操作系统下进行了重新编译；ADT 版本为 1.5.1。

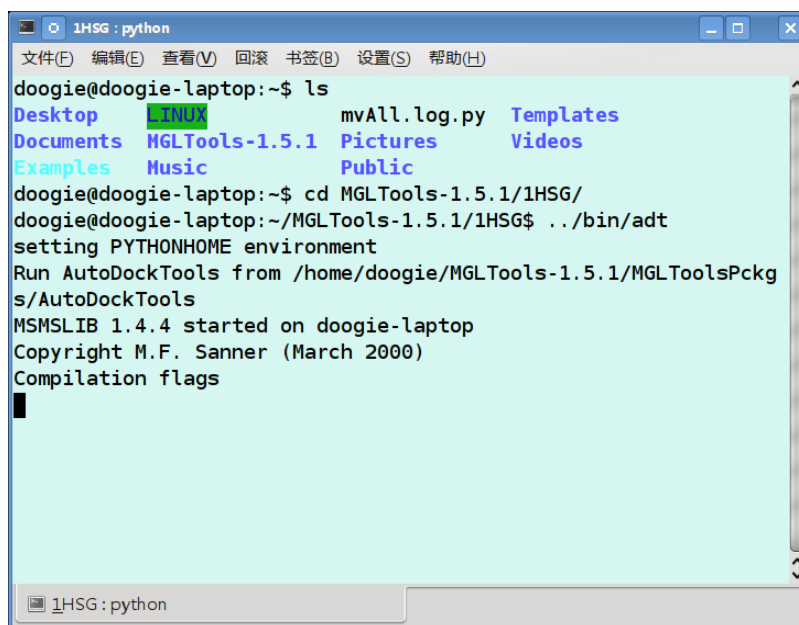
1. 获取 Receptor（受体）及 Ligand（配体）结构文件

我们可以从 www.pdb.org 网站上下载“`1hsg.pdb`”文件，运用 ADT、VMD 以及 PyMol 之类的常用分子显示及编辑软件来将“`1hsg.pdb`”中的蛋白质受体和小分子配体分离开来，保存成两个单另的文件以备后面使用。同时还可以在 AutoDock 网站下载“`tutorial4.tar.gz`”

（<http://autodock.scripps.edu/faqs-help/tutorial/using-autodock-4-with-autodocktools>，受体文件：“`hsg1.pdb`”以及配体文件：“`ind.pdb`”）来进行操作。下面我们就以从 AutoDock 网站上获取的结构文件来进行后面的操作。

2. 设置工作目录及工作环境

新建工作文件夹“`1HSG`”，将编译好的“`autogrid4, autodock4`”程序以及“`hsg1.pdb, ind.pdb`”两个 PDB 文件拷贝到此文件夹下。打开控制台，切换目录到该文件夹下，运行 ADT，这样 ADT 的默认路径就是“`1HSG`”文件夹，此后所有输入/输出文件的默认路径都是 `1HSG`，方便后面操作（图 6）。



```
1HSG : python
文件(F) 编辑(E) 查看(V) 回滚 书签(B) 设置(S) 帮助(H)
doogie@doogie-laptop:~$ ls
Desktop      .INIX      mvAll.log.py  Templates
Documents   MGLTools-1.5.1  Pictures      Videos
Examples     Music       Public
doogie@doogie-laptop:~$ cd MGLTools-1.5.1/1HSG/
doogie@doogie-laptop:~/MGLTools-1.5.1/1HSG$ ../bin/adt
setting PYTHONHOME environment
Run AutoDockTools from /home/doogie/MGLTools-1.5.1/MGLToolsPckgs/
AutoDockTools
MSMSLIB 1.4.4 started on doogie-laptop
Copyright M.F. Sanner (March 2000)
Compilation flags
█
```

图 6: 控制台窗口切换工作路径并启动 ADT

3. 准备 receptor 分子

PMV 菜单: **File** → **Read Molecule** 选择 “hsg1.pdb “ 打开 Receptor 分子:

PMV 菜单: **Select** → **Select From String** 打开 **Select From String** 对话框, 在 **Residue** 框中输入 **HOH***, **Atom** 框中输入 *****, 点击 **Add** 选中所有水分子。最后, 点击 **Dismiss** 按钮关闭 **Select From String** 对话框 (图 7)。

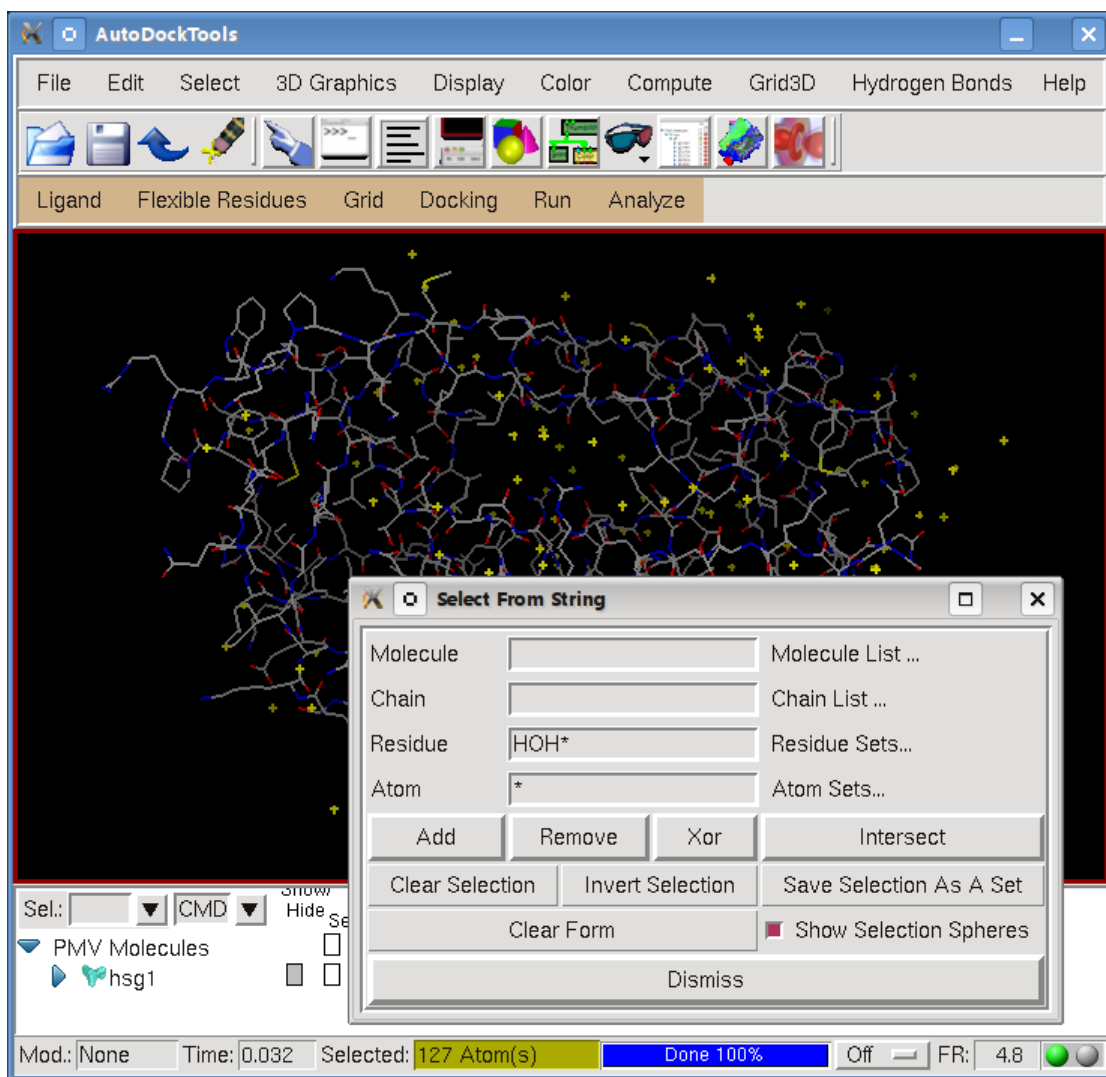


图 7: **Select From String** 对话框及被选中的 H₂O 分子（黄色“+”表示被选中的原子，没有 H，所以选中的都是 O 原子）

水分子在哪，如何选中？

PMV 菜单: **Edit** → **Delete** → **Delete AtomSet** 删除已选择的水分子，点击弹出的警告窗口上的 **CONTINUE** 按钮删除。

PMV 菜单: **Edit** → **Hydrogens** → **Add** 为 Receptor 分子加 H（X 射线衍射无法获 H 的坐标数据，图 8）。

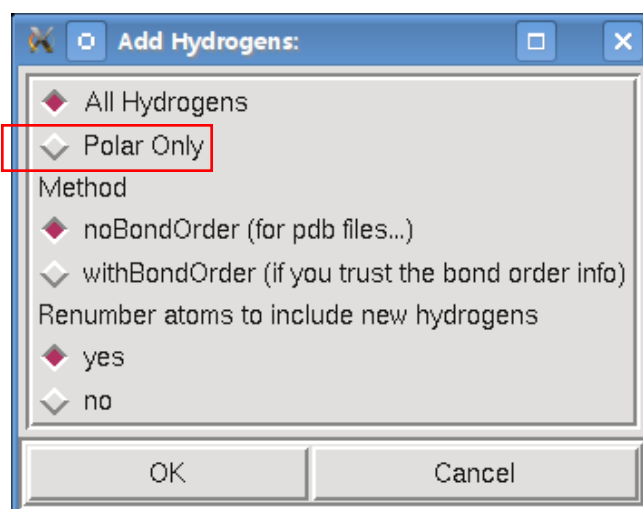


图 8: Add Hydrogens 对话框

PMV 菜单: **File** → **Save** → **Write PDB** 保存修改过的 receptor 分子 (图 9)。

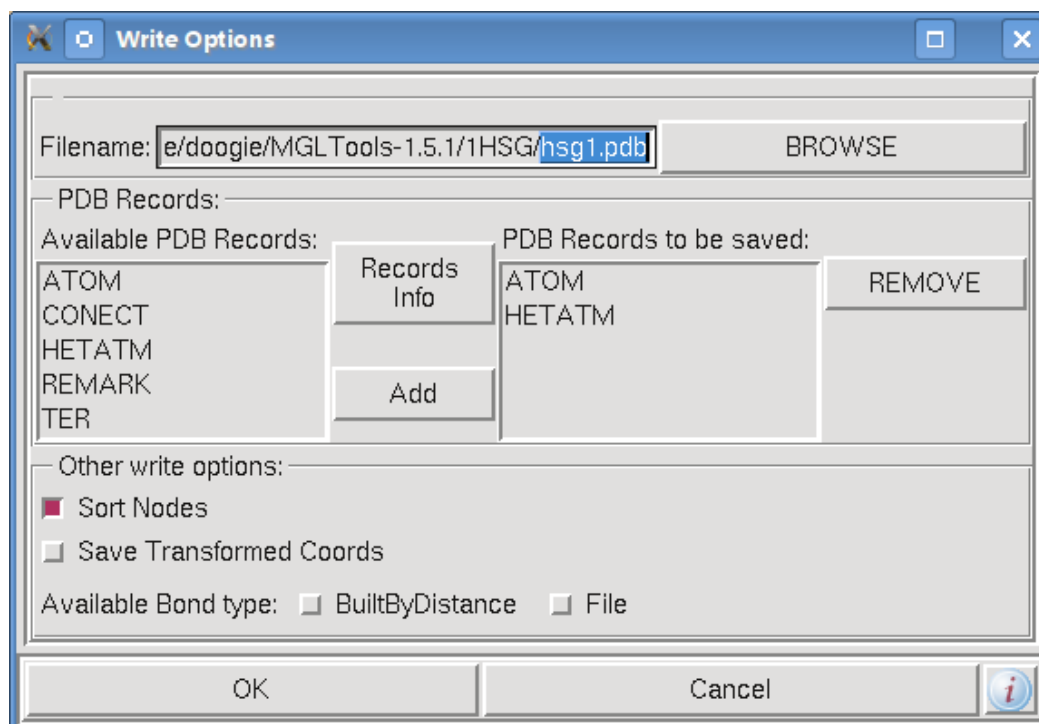


图 9: Write PDB 对话框。(注: Other write option 选择 Sort Nodes)

4. 准备 Ligand 分子

PMV 菜单: **Display** → **Show/Hide Molecule** 隐藏 hsg1 分子 (图 10)。



图 10: **Show/Hide Molecule** 对话框

ADT 菜单中: **Ligand** → **Input** → **Open ...** 在弹出的对话框中将文件类型由 PDBQT 改为 PDB, 选择 “**ind.pdb**”, 打开 Ligand 分子 (图 11)。

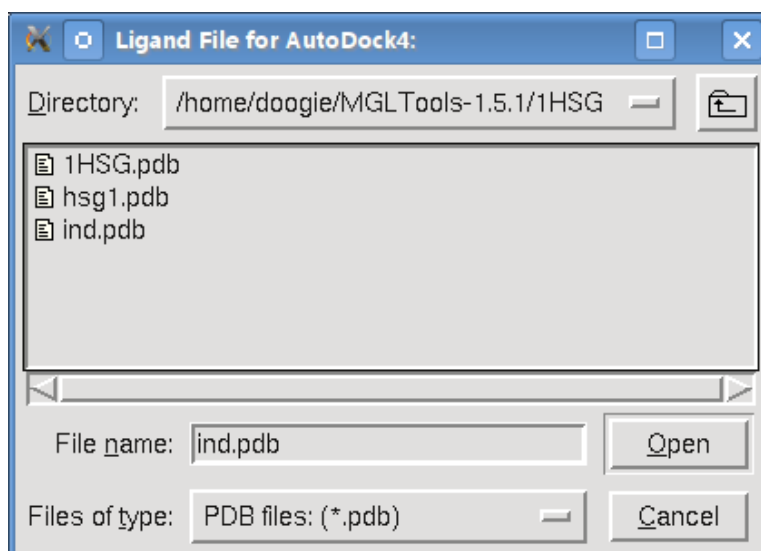


图 11: **Ligand File for AutoDock** 对话框

在打开 Ligand 分子时 ADT 会对该分子进行初始化, 初始化包含一系列操作:

- ADT 检测 Ligand 分子是否已经加了电荷, 如果没有, 则自动加上 Gasteiger 电荷。需要注意的是: 如果要使 Gasteiger 计算正确, 就必须将 Ligand 上的所有 H 加上, 包括极性的以及非极性的。如果电荷全部为 0, 则 ADT 会试图加上电荷。同时还将检测每个残基上的总电荷是否为整数。
- ADT 检测并合并非极性的 H。
- ADT 将 Ligand 中的每个原子指派为 “AutoDock 原子类型”, 原子类型见前一部分。

以上初始化操作完成后, 弹出 **summary for ind** (图 12) 信息窗口, 包含了以上操作的统计信息, 点击 **OK** 关闭。

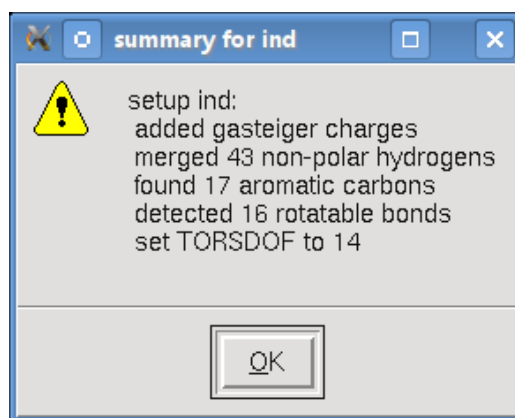


图 12: **summary for ind** 信息窗口

此步骤没有图形显示

*按ctrl+n
可以显示出小分子
配体

ADT 菜单: **Ligand** → **Torsion Tree** → **Detect Root...** ADT 自动判定 Ligand 的 Root;

ADT 菜单: **Ligand** → **Torsion Tree** → **Show Root Expansion** 显示Root扩展信息;

ADT 菜单: **Ligand** → **Torsion Tree** → **Show/Hide Root Marker** 显示/隐藏Root标记;

ADT 菜单: **Ligand** → **Torsion Tree** → **Choose Torsions...** 选择Ligand中可扭转的键, 弹出

Torsion Count对话框中点击**Make all active bonds non-rotatable**, **Make all rotatable bonds** 这个没有 rotatable 以及 **Make all amide bonds rotatable**, 最后结果如下图 (图13) 所示: 该Ligand分子32个键中共有14个被设置成可扭转的键 (rotatable, 绿色), 点击**Done**确定并关闭此对话框。

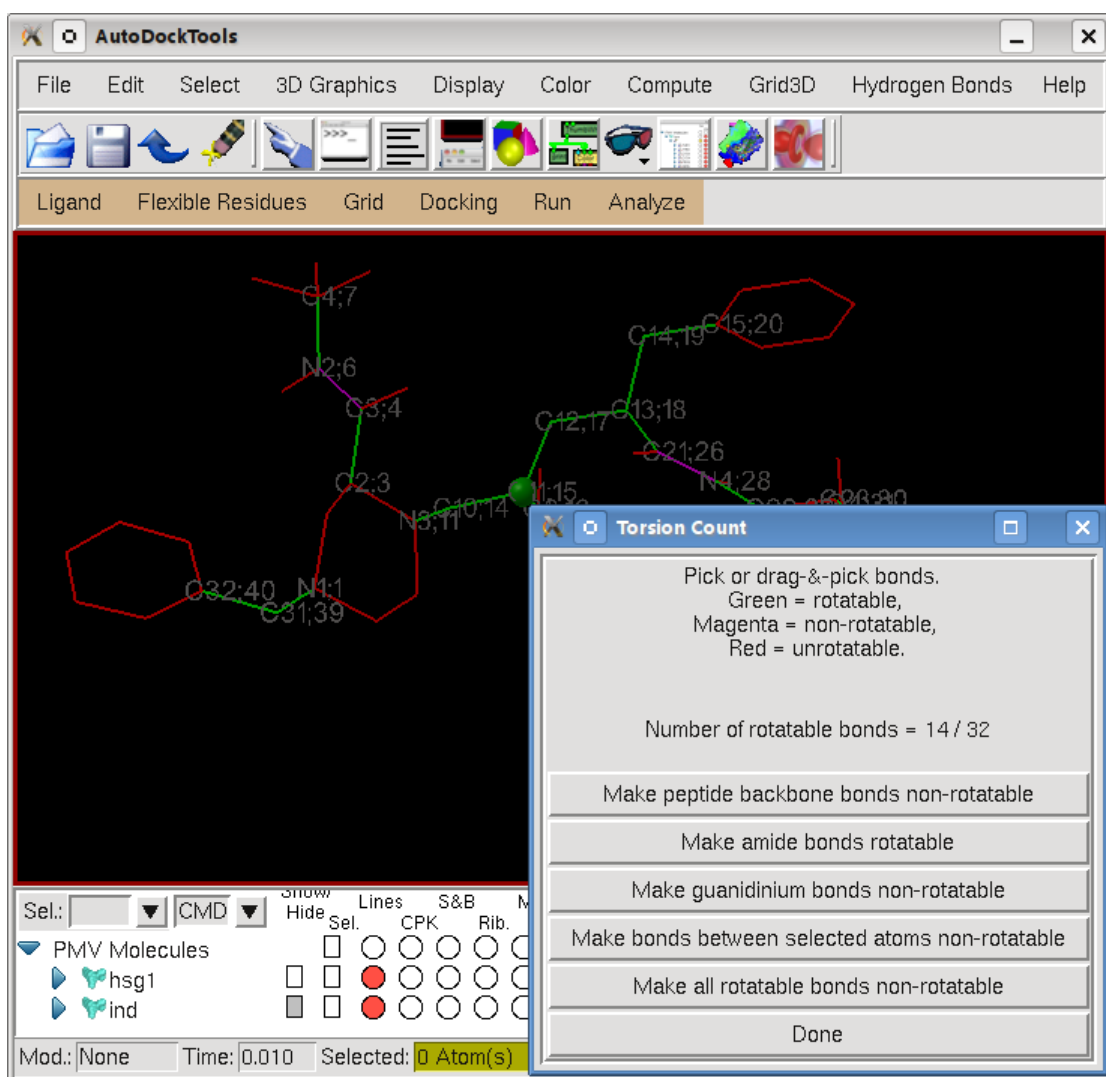


图 13: **Torsion Count** 对话框及 Ligand 分子 Torsion 设置效果 (绿色圆球表示 Root, 绿色键表示可扭转, 紫色表示非扭转, 红色表示不可扭转)

ADT 菜单: **Ligand** → **Torsion Tree** → **Set Number of Torsions...** 弹出 **Set Number of Active Torsions** 对话框, 以设置可活动的键的数量, 同时指定设定活动键时是需要移动最少的原子 (fewest atoms) 还是最多的原子 (most atoms)。在这里我们设置 **fewest atoms**, 数量为 6 (图 14), 点击 **Dismiss** 关闭对话框。

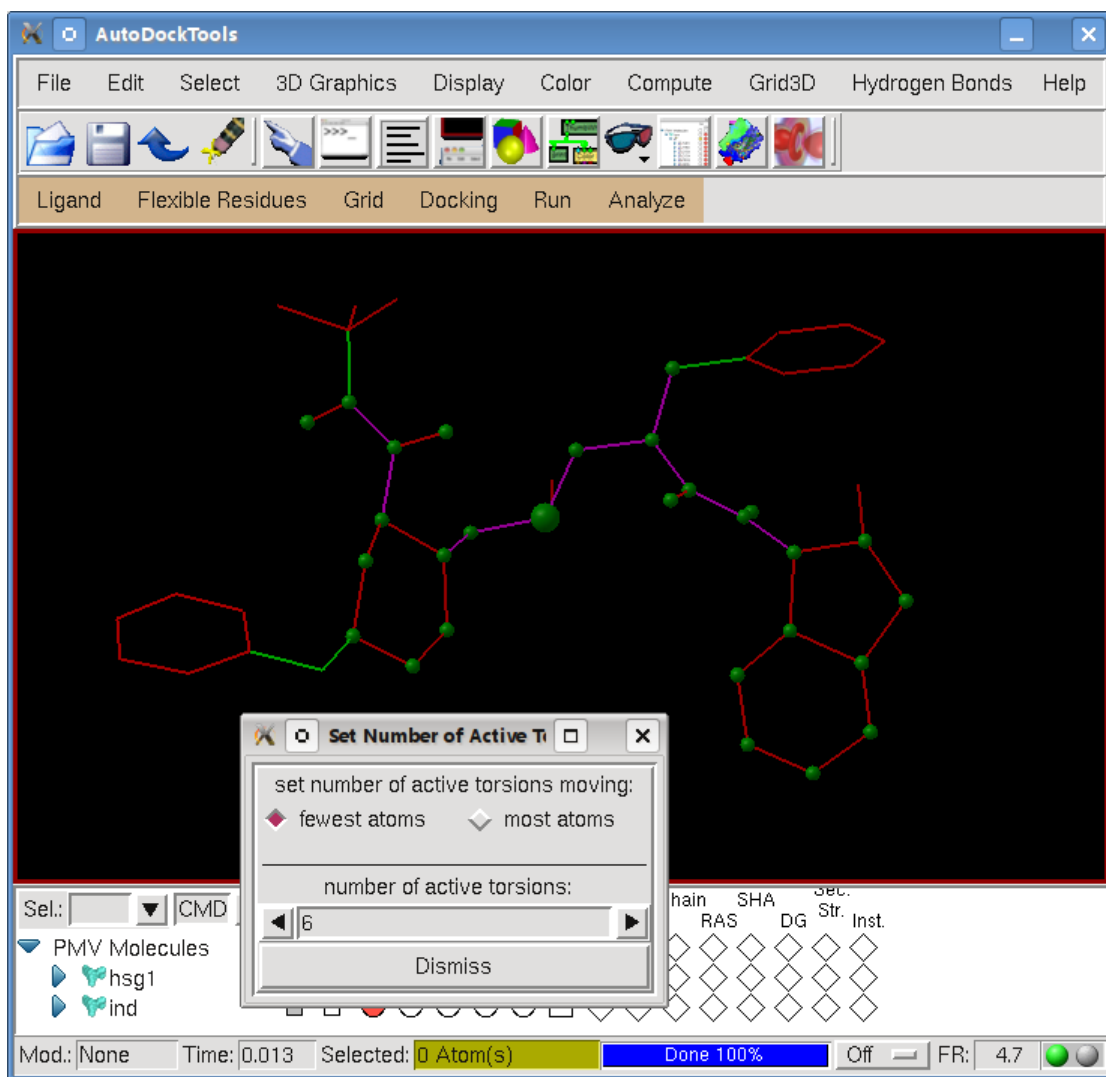


图 14: Set Number of Active Torsions 对话框

至此 Ligand 准备完毕，需将其保存为含有原子坐标、AutoDock 原子类型、电荷以及可扭转键等信息的 PDBQT 格式的文件：

ADT 菜单：Ligand → Output → Save as PDBQT... 将准备好的 Ligand 分子保存为“ind.pdbqt”文件。

5. 准备柔性残基文件

在 AutoDock3 的对接中 Ligand 是柔性的而 Receptor 则是刚性的；而在 AutoDock4 中不但 Ligand 可以是柔性的，而且 Receptor 中的部分氨基酸残基也可以设为柔性的，这就进一步提高了对接的精度。

ADT 菜单：Flexible Residues → Input → Choose Macromolecule... 选择需要设置柔性残基的大分子，在这里我们选择之前编辑好的 hsg1 分子。如果之前删除了该分子，则在 ADT 菜单：FlexibleResidues → Input → Open Macromolecule... 打开该分子。此时会弹出对话框，询问是否合并没有极性的 H，点击 Yes，之后会弹出统计信息窗口，点击 Yes 关闭（图 15）。

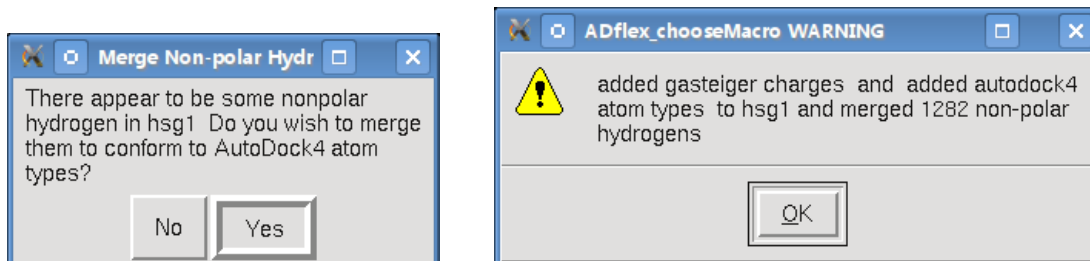


图15: Merge Non-polar Hydrogens及ADflex_chooseMacro对话框

ADT菜单: **Select** → **Select From String** 选择需要设置成为柔性的残基。在**Select From String**对话框中的**Residue**框中输入**ARG8**, 点击**Add**选中编号为ARG8的残基(图), 再点击**Dismiss**关闭对话框。下图(图16)可见, hsg1两个亚基上的8号ARG氨基酸残基被选中了。

哪些残基需要设置成柔性的残基。

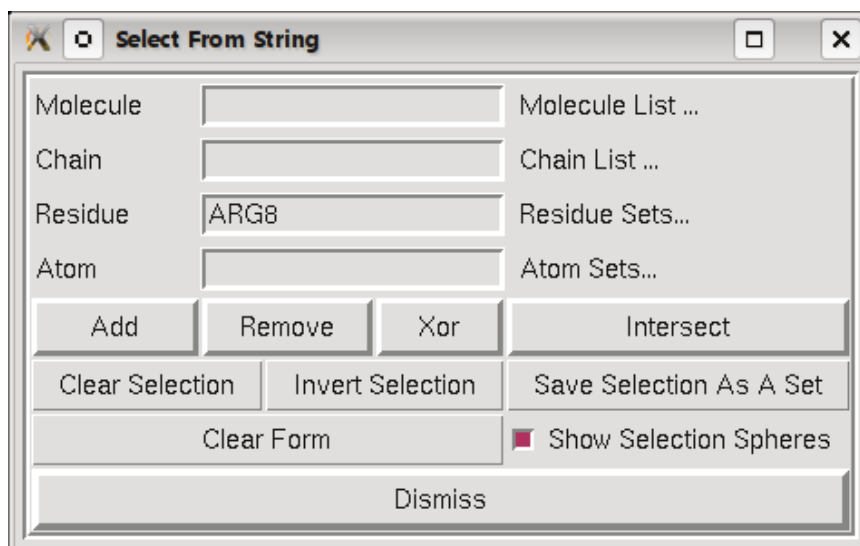


图16: 使用Select From String对话框选中ARG8残基

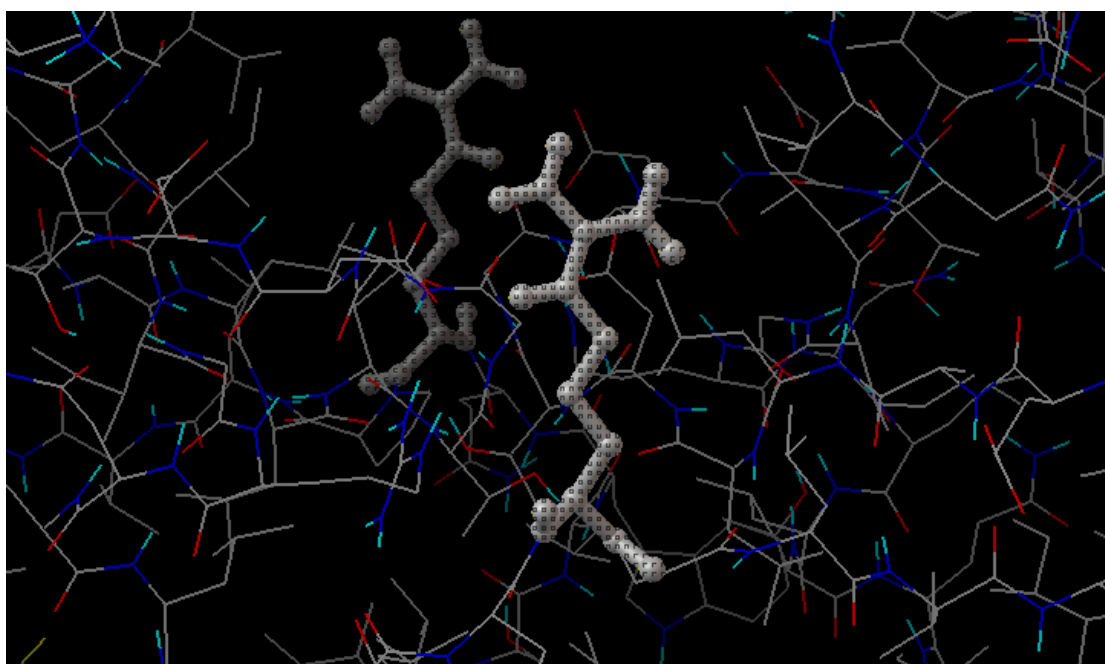


图17: hsg1两个亚基上被选中的8号ARG氨基酸残基(为了便于观察,将这两个残基按Ball and Stick模型显示)

ADT菜单: **Flexible Residues** → **Choose Torsions in Currently Selected Residues...** 将选中的残基标记为柔性残基并设置可扭转键的数量。在分子显示窗口中分别点击两个残基上CA和CB原子之间的键,使之变成非扭转的(non-rotatable, 紫色),这样两个残基中的32个键共有6个是可扭转(rotatable)的(图18),点击**Close**关闭对话框:

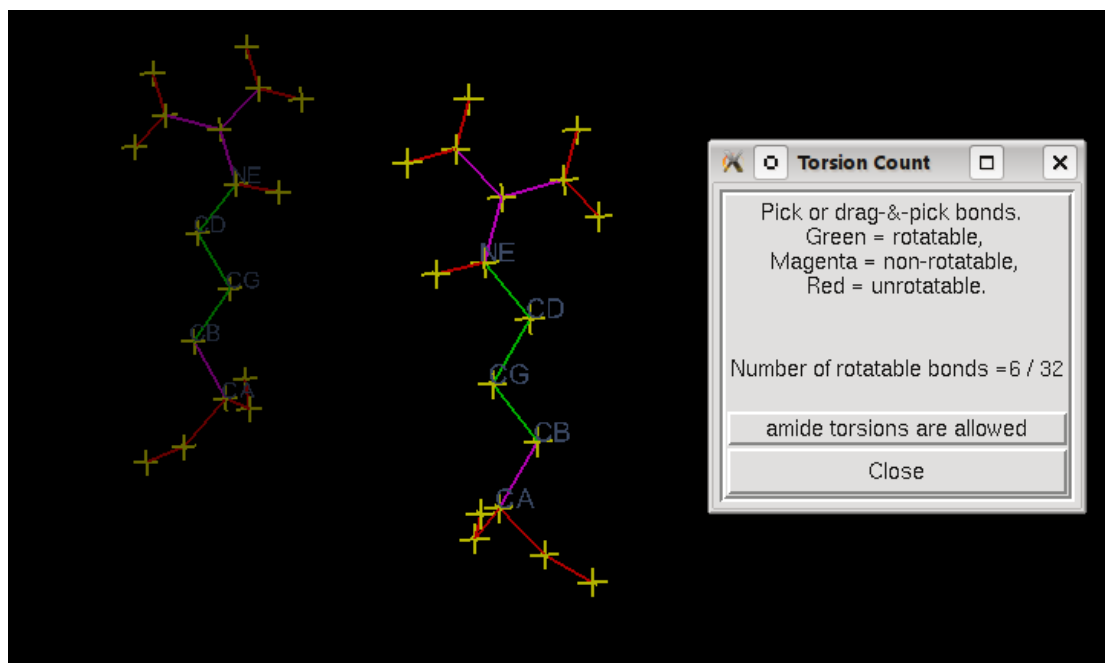


图18: 设置柔性残基上的可扭转键

ADT菜单: **Flexible Residues** → **Output** → **Save Flexible PDBQT...** 保存柔性残基文件,文件名“**hsg1_flex.pdbqt**”;

ADT菜单: **Flexible Residues** → **Output** → **Save Rigid PDBQT...** 保存刚性残基文件,文件名“**hsg1_rigid.pdbqt**”;

PMV菜单: **Edit** → **Delete** → **Delete Molecule** 删除**hsg1**分子。

6. 准备大分子

ADT菜单: **Grid** → **Macromolecule** → **Open...** 打开之前处理好并保存的“**hsg1_rigid.pdbqt**”,弹出对话框询问是否保留之前已经加上的电荷以代替ADT自动加上Gasteiger电荷,点击**Yes**,再点击**OK**关闭弹出的警告窗口(图19)。

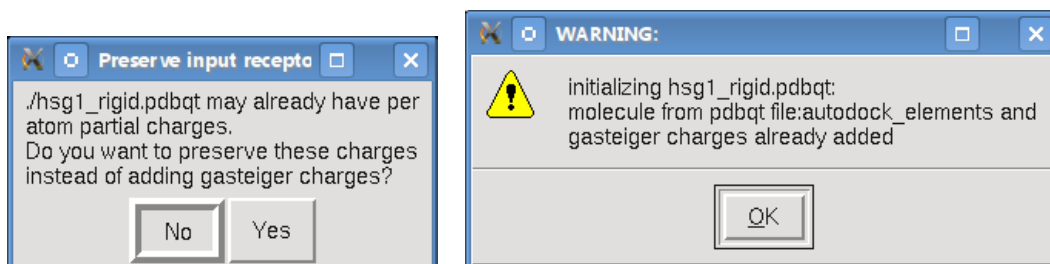


图19: 打开大分子 (Macromolecule) 文件后弹出的对话框

7. 准备 AutoGrid 参数文件

ADT 菜单: **Grid** → **Set Map Types** → **Choose Ligand...** 选择之前准备好的Ligand分子“ind”，如果之前删除了该分子则通过ADT菜单: **Grid** → **Set Map Types** → **Open Ligand...** 打开“ind.pdbqt”文件。

ADT 菜单: **Grid** → **Grid Box...** 打开**Grid Options**对话框，将格子的大小设置为X, Y, Z: **60, 60, 66**，格点间隔为默认值**0.375 Å**，这样格子中共包含249307个格点，然后将格子中心设为**2.5, 6.5, -7.5** (x, y, z)。设置完成后点击Grid Options菜单中: **File** → **Close saving current** 保存并关闭对话框 (图20)。

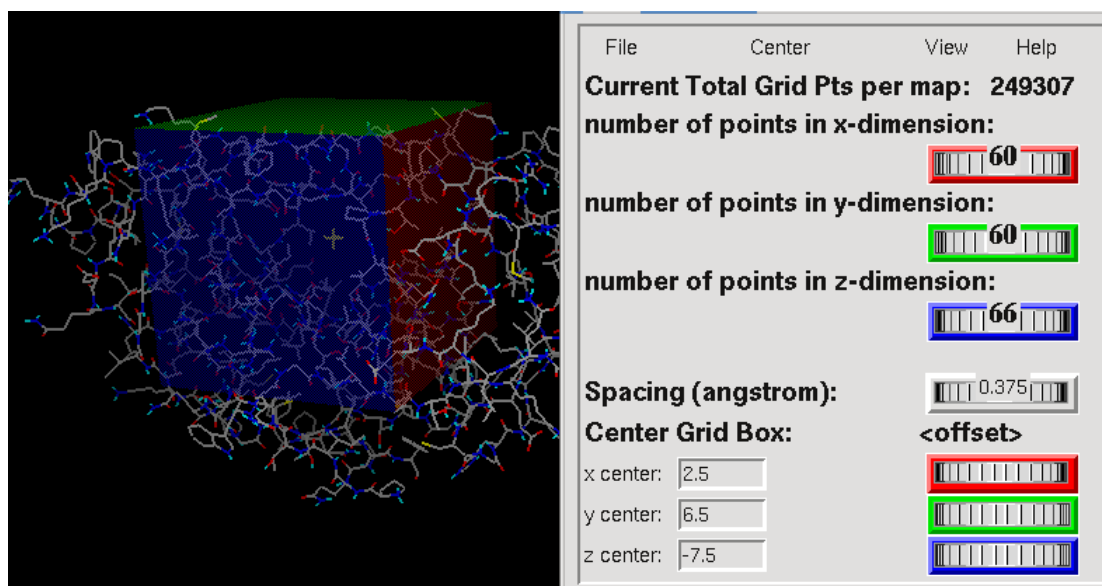


图20: 打开**Grid Options**对话框，设置Grid参数

ADT 菜单: **Grid** → **Output** → **Save GPF...** 将刚才设置好的Grid参数保存成GPF文件 (Grid Parameter File, 格子参数文件) “hsg1.gpf”。

完成后还可以通过点击ADT菜单: **Grid** → **Edit GPF...** 手工编辑修改刚才通过ADT生成的GDF文件 (图21):

何时需要何种修改?

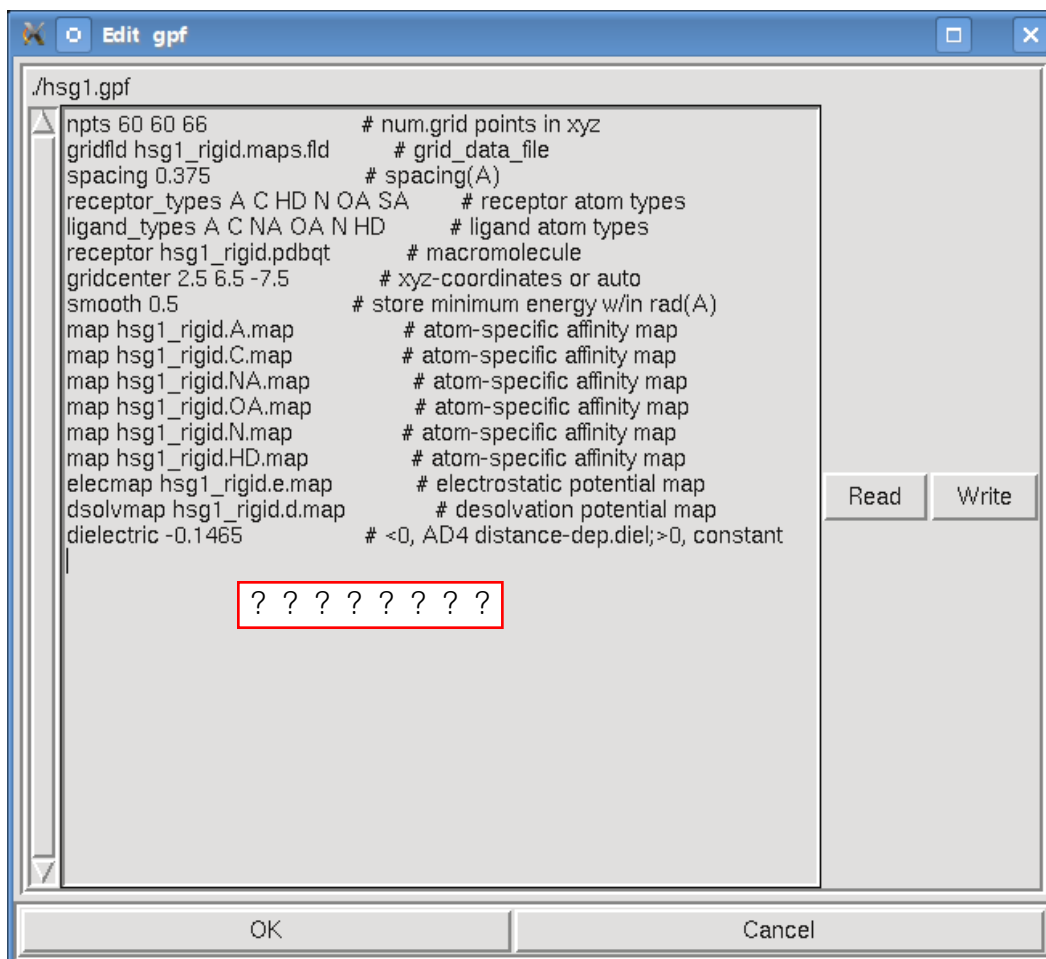


图21: Edit gpf对话框

8. 运行 Autogrid4

ADT 菜单: **Run** → **Run AutoGrid...** 启动AutoGrid图形界面 (图22): **Host Name**(主机名)如果不是在远程计算机上运行程序则无需修改, 点击**Program Pathname** (程序路径及名称)后的**Browse**按钮, 选择之前放入1HSG文件夹的**autogrid4**程序, 以替换界面中默认的autogrid3, **Parameter File** (参数文件, 上一步准备好的AutoGrid参数文件)以及**Log File** (程序运行记录文件)程序一般情况都能自动设置好, 如须修改点击相应的**Browse**按钮选择正确的文件即可。

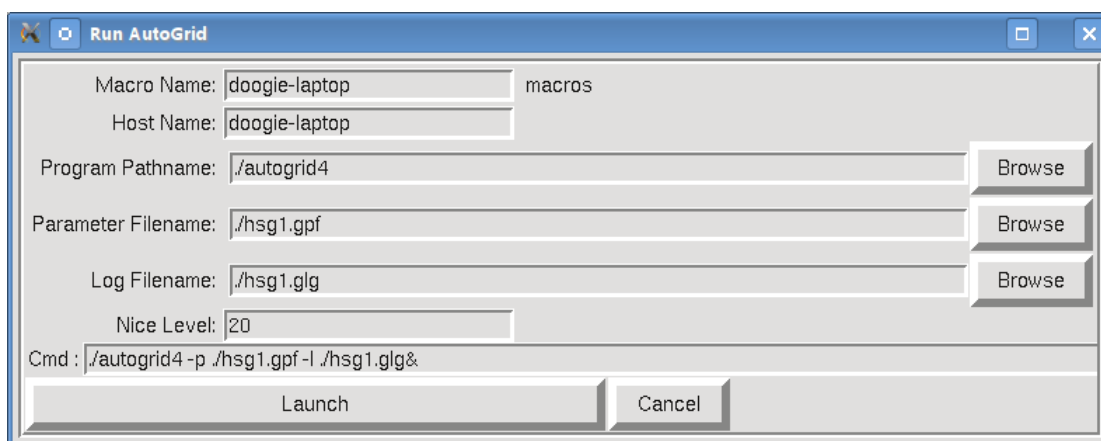


图22: Run AutoGrid对话框

点击 **Launch** 按钮，运行程序进行计算，同时弹出 **AutoDock Process Manager**（AutoDock 进程管理器，图23），显示进程编号、运行时间及状态，还可以随时 Kill 该 autogrid4 进程，运行完毕后该对话框会自动关闭。

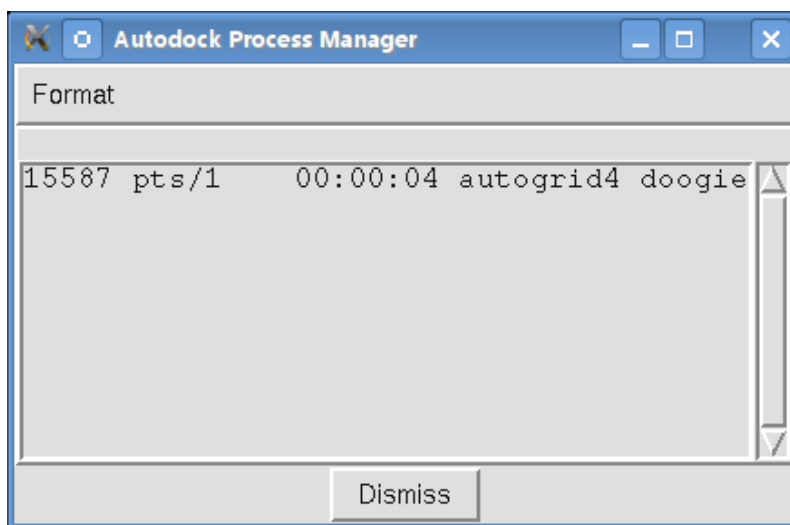


图23: Autodock Process Manager对话框

此外，还可以在控制台中输入命令以运行 autogrid4:

```
autogrid4 -p hsg1.gpf -l hsg1.glg & #注意文件名及路径
```

AutoGrid4程序运行完毕后，除了生成一个 hsg1.glg 记录文件外，最主要的是生成一系列针对不同原子探针的范德华作用力、静电力以及去溶剂化作用力的 Map 文件:

```
hsg1_rigid.A.map # atom-specific affinity map
hsg1_rigid.C.map # atom-specific affinity map
hsg1_rigid.NA.map # atom-specific affinity map
hsg1_rigid.OA.map # atom-specific affinity map
hsg1_rigid.N.map # atom-specific affinity map
hsg1_rigid.HD.map # atom-specific affinity map
hsg1_rigid.e.map # electrostatic potential map
hsg1_rigid.d.map # desolvation potential map
```

9. 准备 Dock 参数文件

ADT 菜单: **Docking** → **Macromolecule** → **Set Rigid Filename...** 设置对接中的刚性分子

“hsg1_rigid.pdbqt”;

ADT 菜单: **Docking** → **Ligand** → **Choose...** 设置对接中的 Ligand 分子为 “ind”;

ADT 菜单: **Docking** → **Macromolecule** → **Set Flexible Residues Filename...** 设置对接中的柔性残基为 “hsg1_flex.pdbqt”;

ADT 菜单: **Docking** → **Search Parameters...** → **Genetic Algorithm...** 打开 **Genetic Algorithm**

Parameters对话框, 设置对接遗传算法搜索参数(图24)。作为练习, 为了缩短计算时间, 将**Maximum Number of evals**改为**short: 250000**, 其他参数均使用系统默认参数, 点击**Accept**确认, 详细参数的设定请参考AutoDock相关文档。

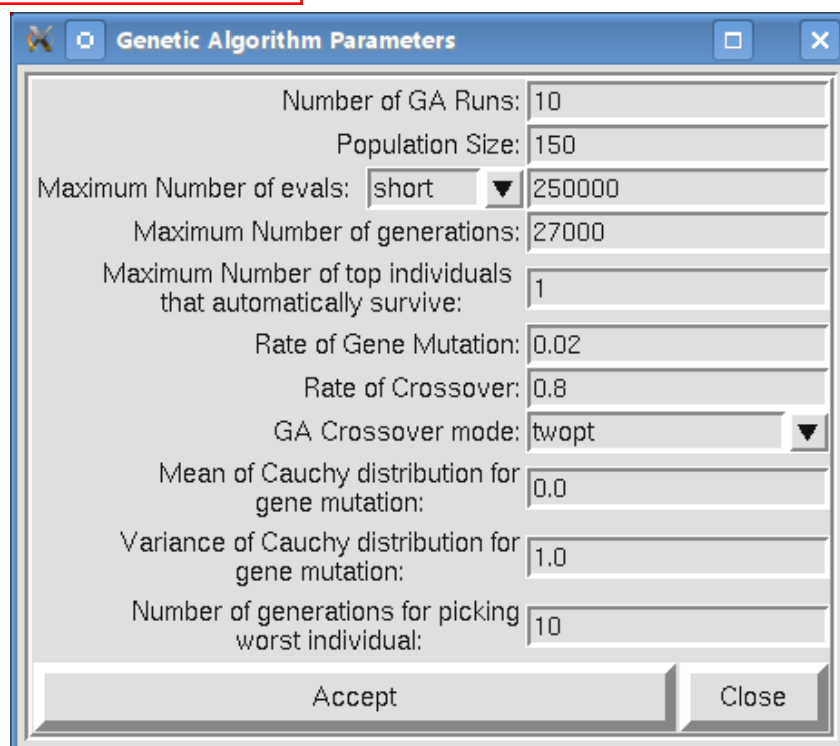


图24: **Genetic Algorithm Parameters**对话框

另外AutoDock还可以使用另外两种计算方法:

1. ADT菜单: **Docking** → **Search Parameters...** → **Simulated Annealing Parameters...** 设置模拟退火算法参数 (图25):

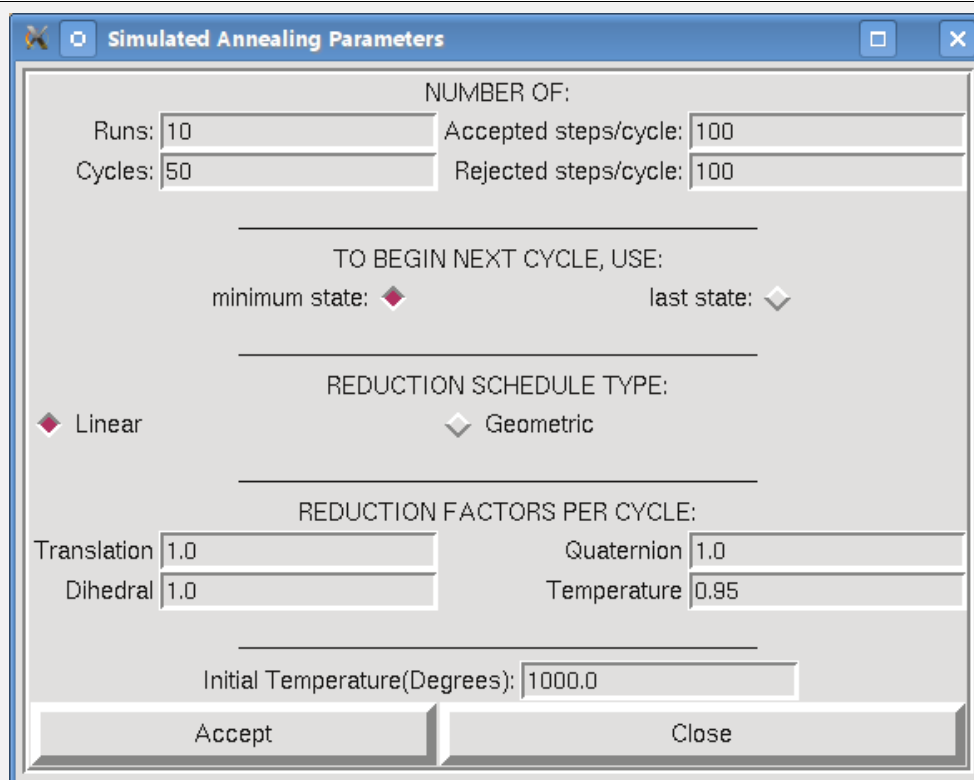


图25: **Simulated Annealing Parameters**对话框

2. ADT菜单: **Docking** → **Search Parameters...** → **Local Search Parameters...** 则可设置局部搜索算法参数 (图26) :

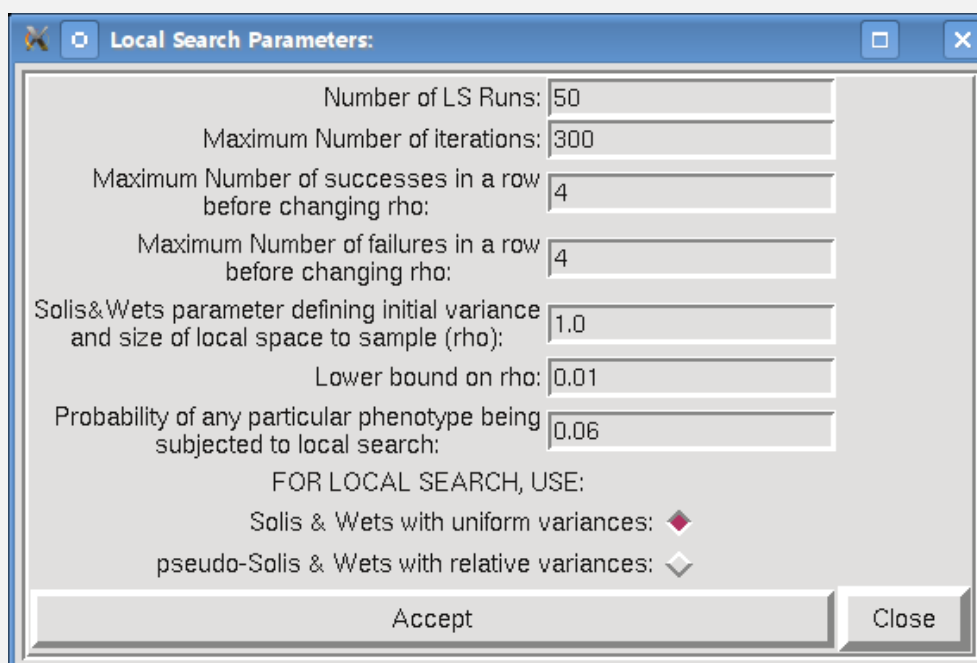


图26: **Local Search Parameters**对话框

ADT菜单: **Docking** → **Docking Parameters...** 打开**Set Docking Run Options**对话框, 设置

对接运行参数。在此，我么全部采用系统默认值（图27），点击Accept保存参数。

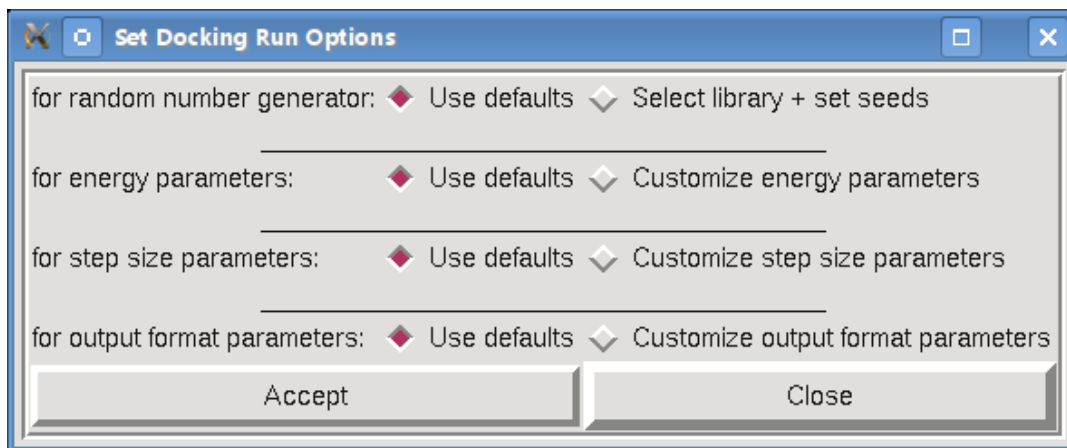


图27: Set Docking Run Options对话框

ADT 菜单: **Docking** → **Output** → **Lamarckian GA...** 输出拉马克遗传算法对接参数文件“ind.dpf”（如果要使用之前提到的其他算法进行计算，那么在设置好相应的参数后，需要在Output菜单中选择输出成相应算法的参数文件才能进行计算）。

ADT 菜单: **Docking** → **Edit DPF...** 打开**Edit dgf**对话框, 可以查看并手动编辑修改刚才生成的DPF文件（图28）。

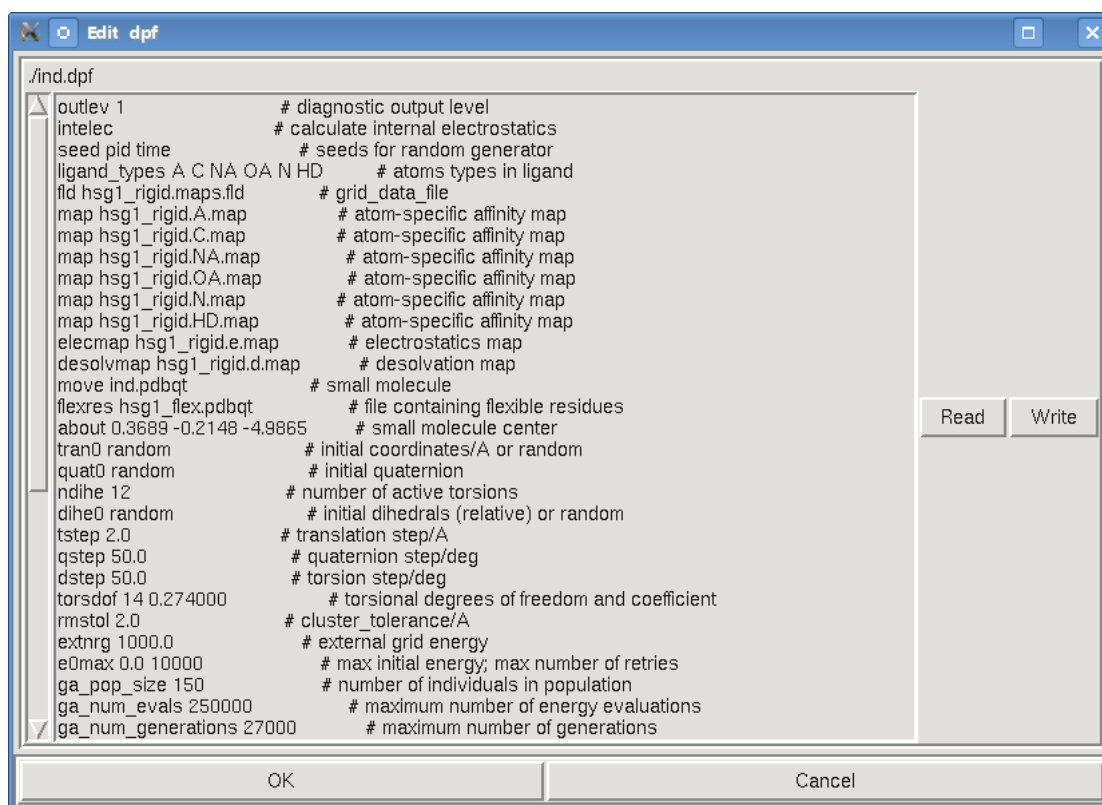


图28: Edit dgf对话框

10. 运行 AutoDock4

与运行AutoGrid4 非常相似，ADT菜单：**Run** → **Run AutoDock...** 启动运行AutoDock图形界面。点击**Program Pathname**后的**Browse**按钮，选择之前放置在1HSG目录中的**autodock4**程序，以替换界面中默认的autodock3，**Parameter File**以及**Log File**程序一般情况都能自动设置好，如须修改点击相应的**Browse**按钮选择正确的文件即可（图29）。

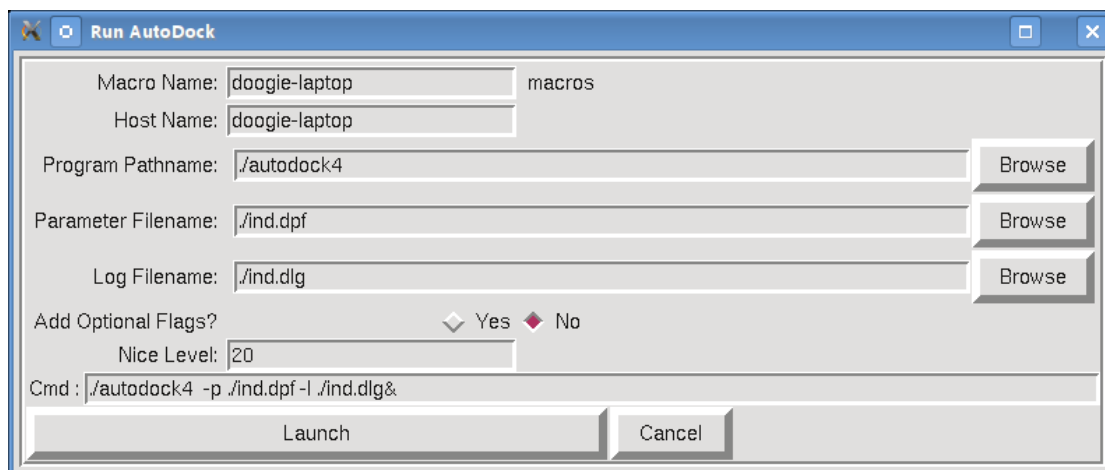


图29: Run AutoDock对话框

点击**Launch**按钮，与运行AutoGrid相似，运行程序进行计算，同时弹出过程管理器，显示进程编号、运行时间及状态，还可以随时Kill该autodock4进程，运行完毕后该对话框会自动关闭。

此外，还可以在控制台中输入命令以运行autodock4:

```
autodock4 -p ind.dpf -l ind.dlg & #注意文件名及路径
```

AutoDock4运行完毕后，程序运行记录以及最终的结果都被保存在“ind.dlg”文件中。

三、结果分析

1. 读取对接记录文件（.DLG）

ADT 菜单：**Analyze** → **Dockings** → **Open ...** 打开对接记录文件“ind.dlg”，弹出信息窗口，显示此日志文件中包含 10 个对接结果的分子构象以及数据，这与我们之前设置的对接参数是一致的（图 30）。

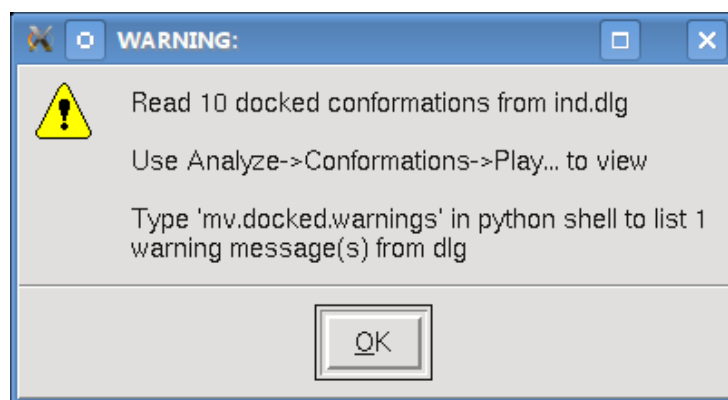


图 30: 打开 **ind.dlg** 文件后弹出的信息窗口

2. 观察对接好的分子构象

ADT 菜单: **Analyze** → **Conformations** → **Load ...** 将对接结果及分子构象载入到图形窗口中, 并且在弹出 **ind Conformation Chooser** 对话框中单击列表中的相应分子构象编号后, 上部显示窗口即可显示此分子构象的对接数据。而如果双击, 则可以将该分子构象载入到分子显示窗口中, 以便观察分析 (图 31)。

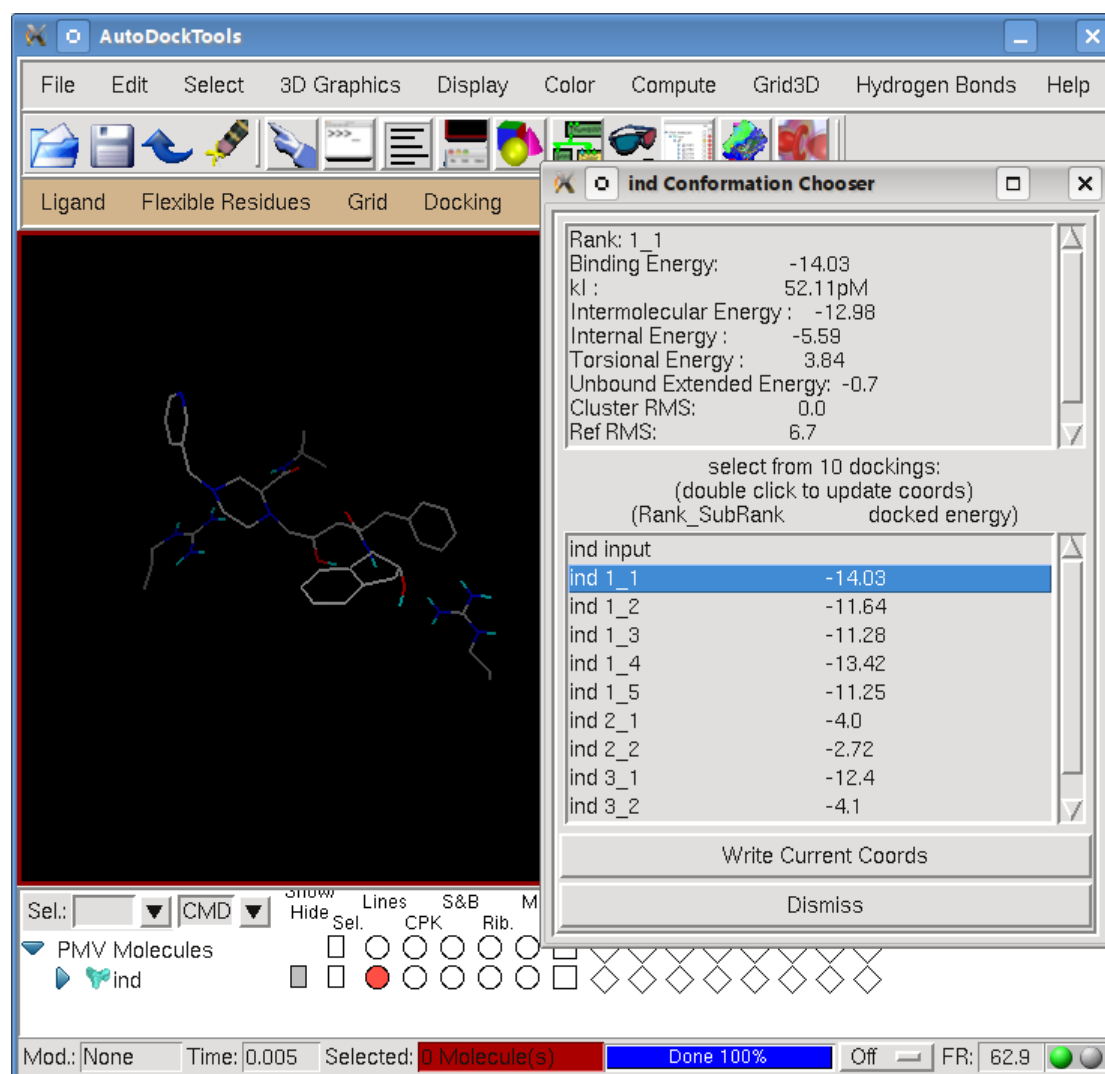


图31: ind Conformation Chooser对话框及显示出来的构象及其对接信息

ADT 菜单: **Analyze** → **Conformations** → **Play ...**弹出 ind 播放控制对话框 (图 32), 显示并播放对接分子构象, 这样可以通过不同的前进/后退按钮选择不同的分子构象, 还可以将所有对接构象按动画方式播放。

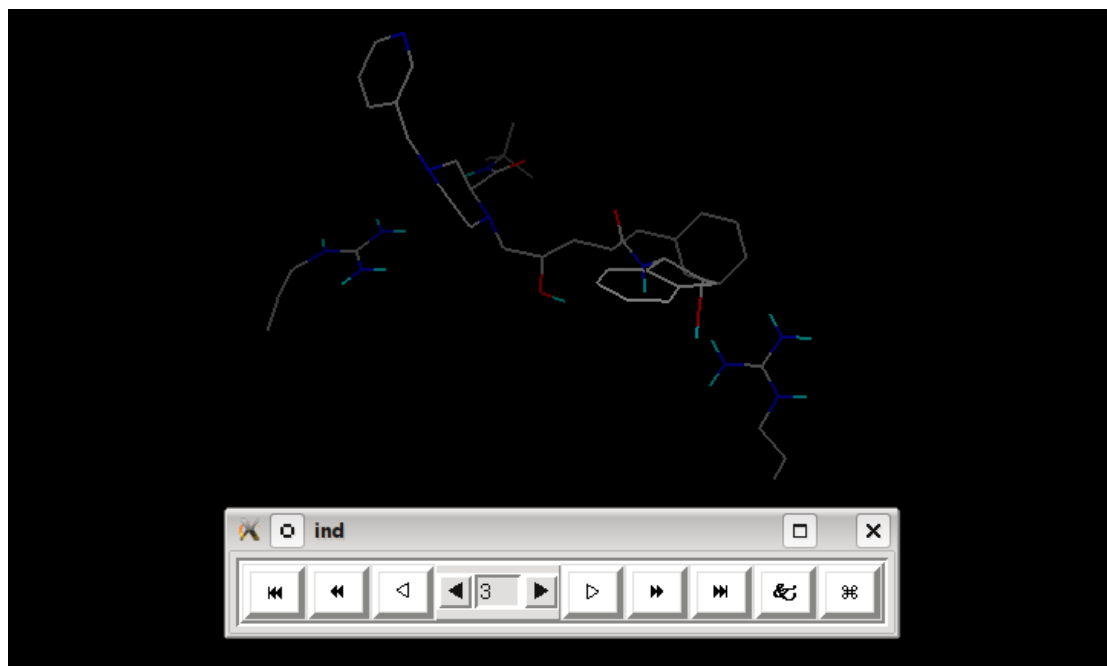


图 32: ind 播放控制对话框

单击 ind 播放控制对话框上的 **Info** 按钮, 打开 **Set Play Options** 对话框, 勾选 **Show Info** 可显示当前构象的相关信息 (图 33)。

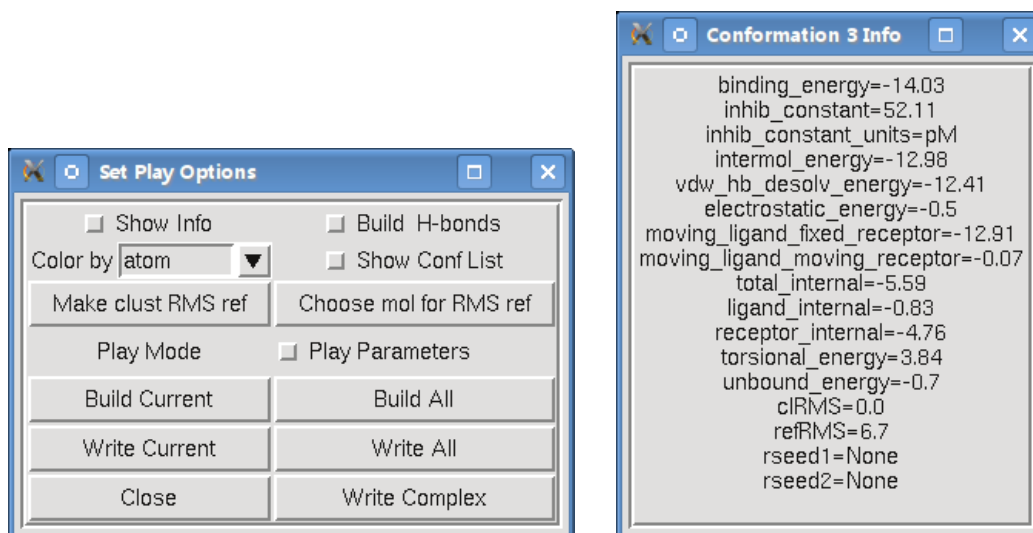


图 33: Set Play Options 及 Info 窗口

将 Color by 下拉菜单选为 vdw, 当前构象则按照范德华作用力的大小来进行着色 (图 34)。

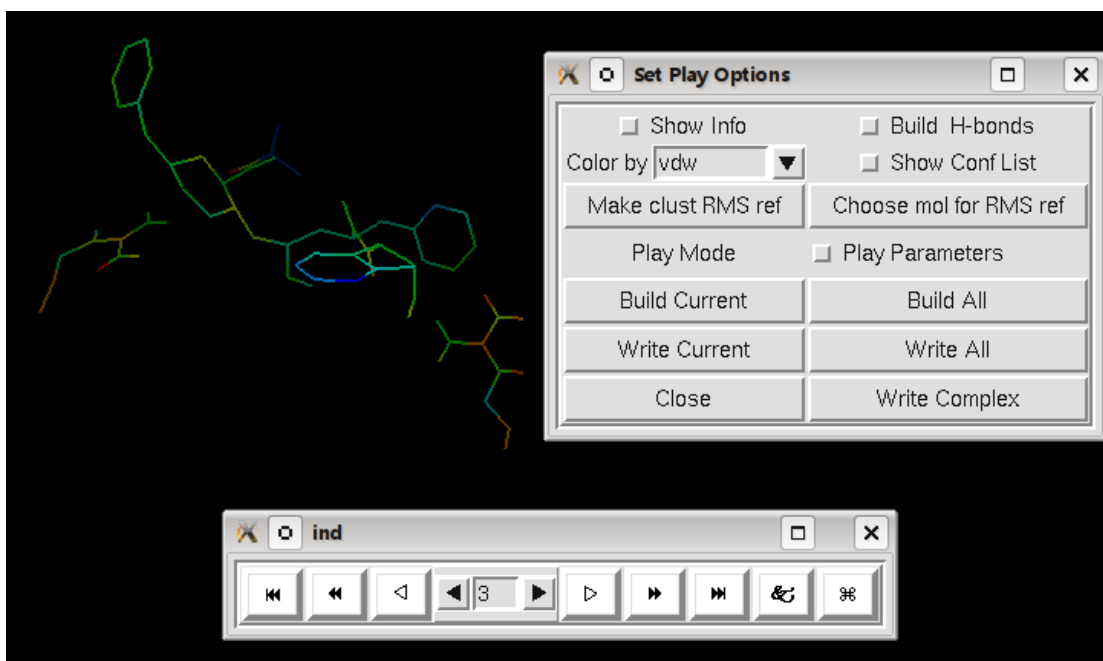


图 34: 使用 **Set Play Options** 对话框更改构象着色方式

点击 **Build All** 按钮可将所有构象重叠在一起显示 (图 35); 如果点击 **Build Current** 按钮则将当前显示的构象添加到显示窗口中, 方便比较不同的对接构象。

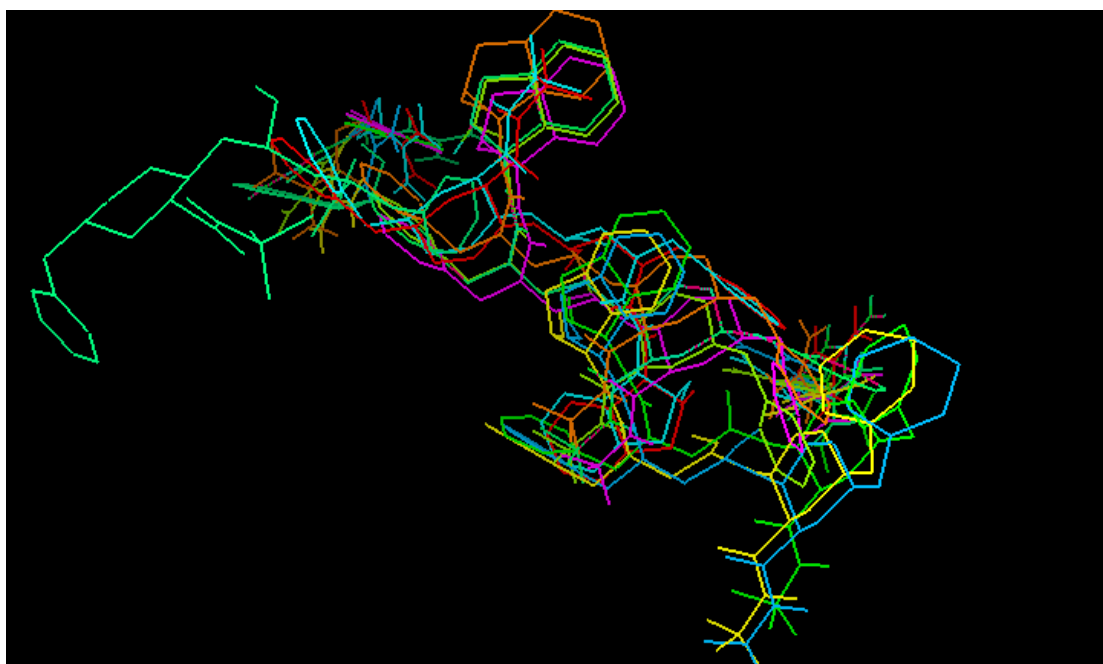


图 35: 所有构象重叠显示效果

点击 **Play Parameters**, 则可设置动画播放选项, 包括: 帧速率、开始帧、结束帧以及步长 (图 36)。

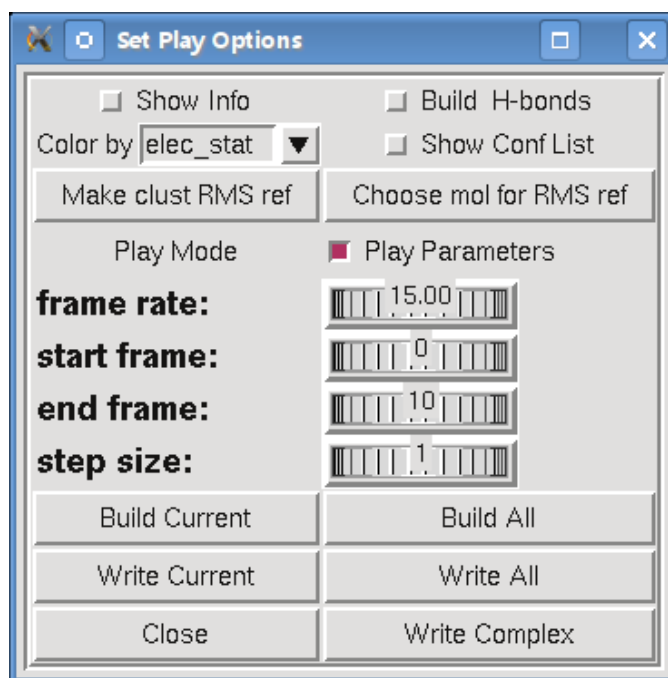


图 36: 播放参数设置面板

3. 聚类构象

ADT可以将对接结果相似的分子构象进行聚类，这样会极大的方便对不同对接结果的分析 and 比较。

ADT菜单: **Analyze** → **Clusterings** → **Show ...** 显示2.0 clustering交互式柱状图 (图37)，单击柱状图上相应的条带，分子显示窗口中将显示相应的分子构象，ADT默认只对tolerance (RMS值公差) 2.0进行一次聚类。图中横坐标为结合能 (Energy)，纵坐标为分子构象数量 (各条带构象数量总合为10)。

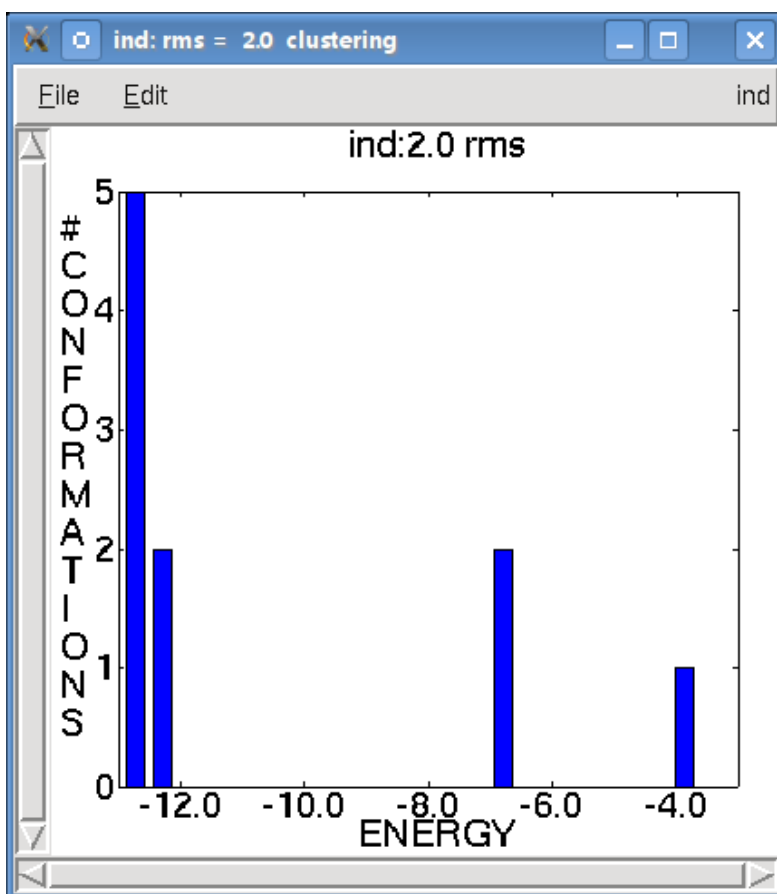


图37: 2.0 rms聚类结果

以上仅按照2.0 rms进行聚类显然还是不容易比较分析对接所产生的10个构象，所以我们分别按 rms: 1.0, 2.0和3.0对对接结果进行重新聚类。

ADT 菜单: **Analyze** → **Clusterings** → **Recluster ...**重新聚类，将 tolerance (RMS 值公差) 设为 1.0, 2.0 和 3.0, 输入输出文件名称, 点击 **OK**重新聚类。

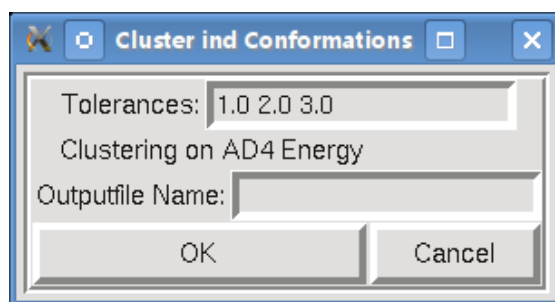


图38: **Cluster ind Conformations**对话框

ADT 菜单: **Analyze** → **Clusterings** → **Show ...** 选择 RMS 值公差后显示聚类图表 (图 38):

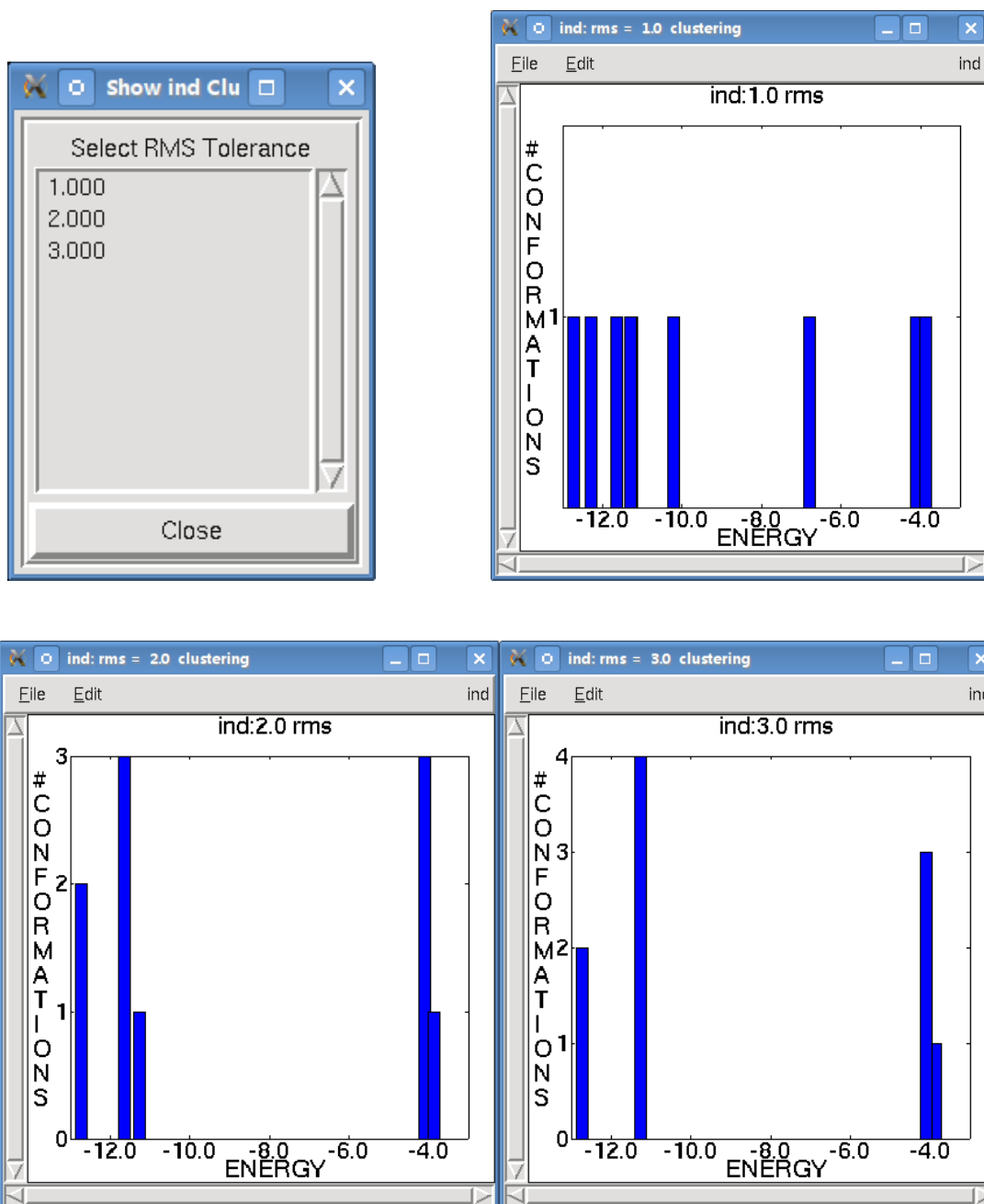


图38: 选择不同rms进行聚类的结果

以ind: 2.0 rms为例，点击柱状图中最左边条带（纵坐标为2包含两个分子构象），分子构象载入分子显示窗口，同时出现ind对话框，通过点击左/右键可以很方便的切换这两个分子（图39）。

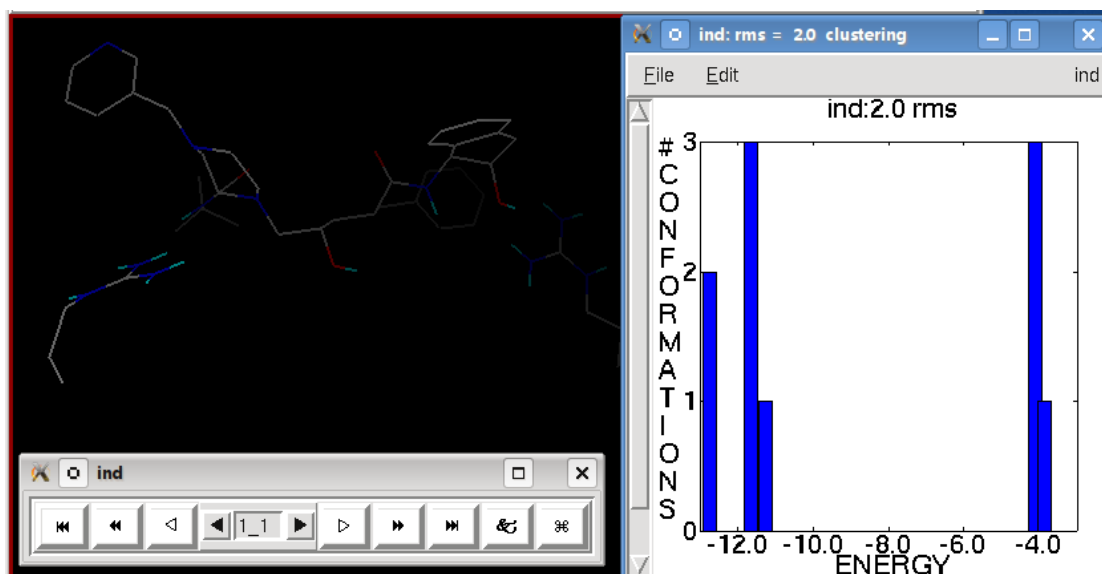


图39: 观察聚类后的分子构象

4. 在 Receptor 环境中观察对接构象

ADT 菜单: **Analyze** → **Macromolecule** → **Open ...** 载入 Receptor 刚性部分分子 “**hsg1_rigid.pdbqt**”, 这样就能看到 Ligand 分子在 Receptor 分子中的情况 (图 40)。

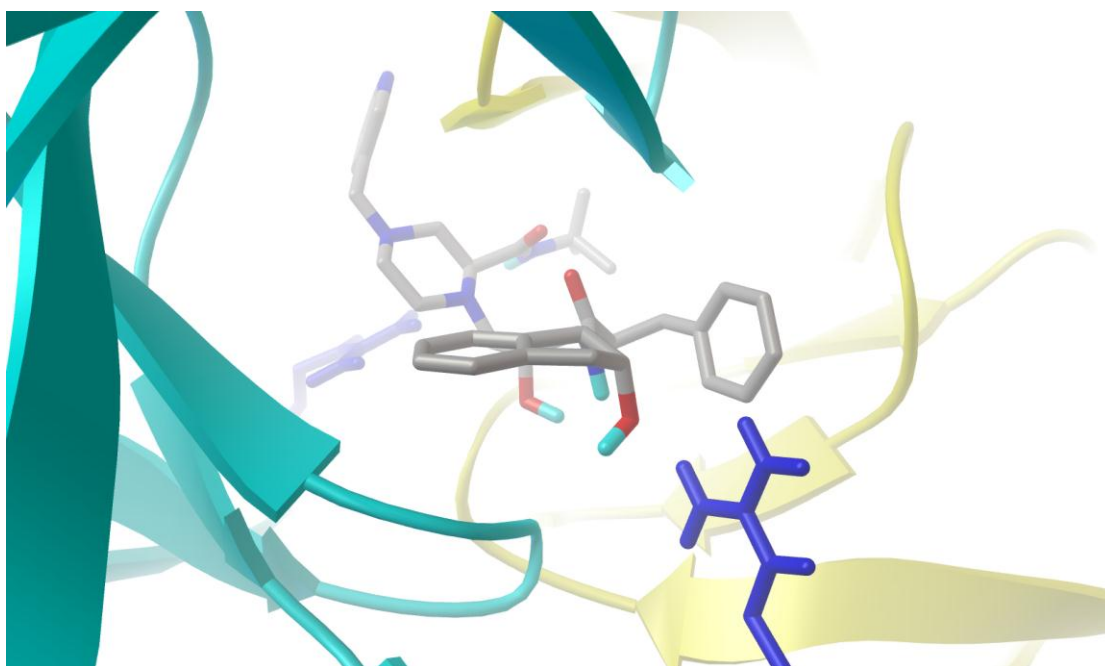


图 40: 载入 Receptor 分子刚性部分后的效果 (为了方便观察, 将 **hsg1_rigid** 以 Ribbon 模型显示, 两种颜色表示两个亚基)

ADT 菜单: **Analyze** → **Grids** → **Open ...** 打开 O 原子的 AutoGrid Map 文件 “**hsg1.OA.map**” (图 41, 42, 43)。

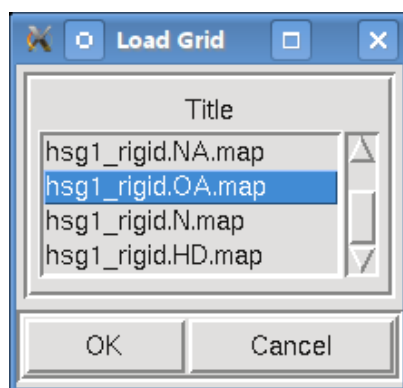


图 41: 选择不同原子类型 Map 文件

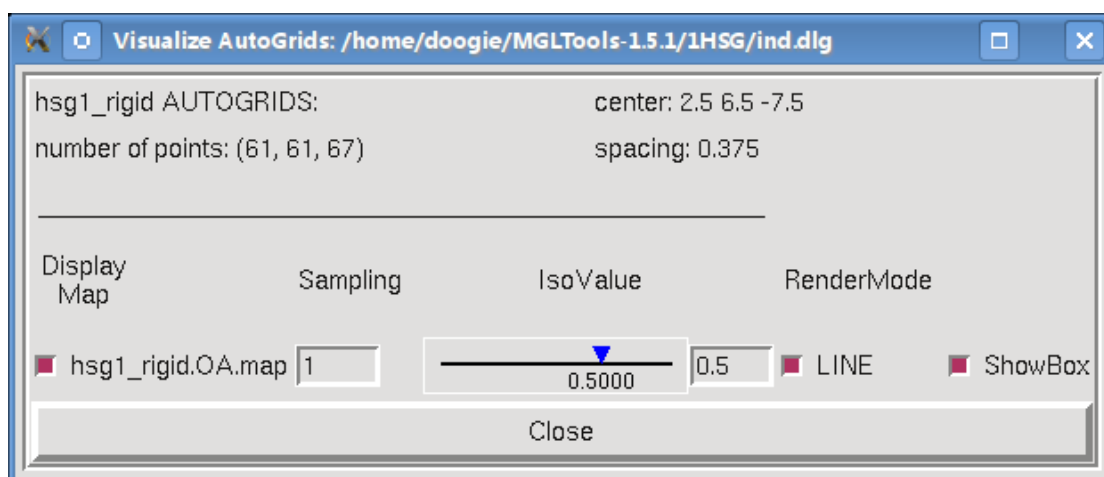


图 42: 设置 isocountour 的 IsoValue 值 (Kcal/Mol)

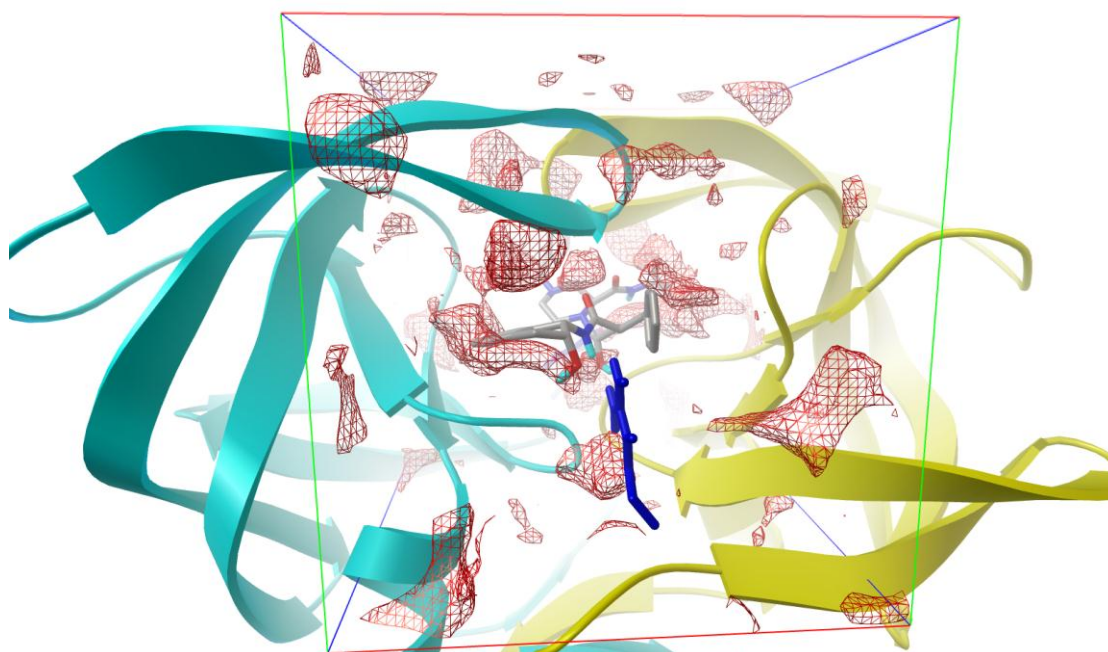


图 43: 打开 “hsg1.OA.map”，IsoValue=0.5 的显示效果

通过上面的操作虽然能显示不同类型原子的操作，但是确不能很好的将 Ligand 分子与周围

Receptor 分子氨基酸残基的相互作用表现出来。现在，按照前面介绍过的方法将 Receptor 上两条链中的两个 ASP25 氨基酸残基显示为球棍模型，调整观察角度及位置。从下图（图 44）中不难看出，抑制剂中（Ligand）O2 原子，被两个 ASP25 残基上的 O 所形成的氧亲和性口袋所浸没。而且如果你载入其他能量较低的分子构象，你会发现不同对接构象中的这个氧原子同样被浸没在这个口袋中，可见这个 O 在这个抑制剂中起了至关重要的作用。

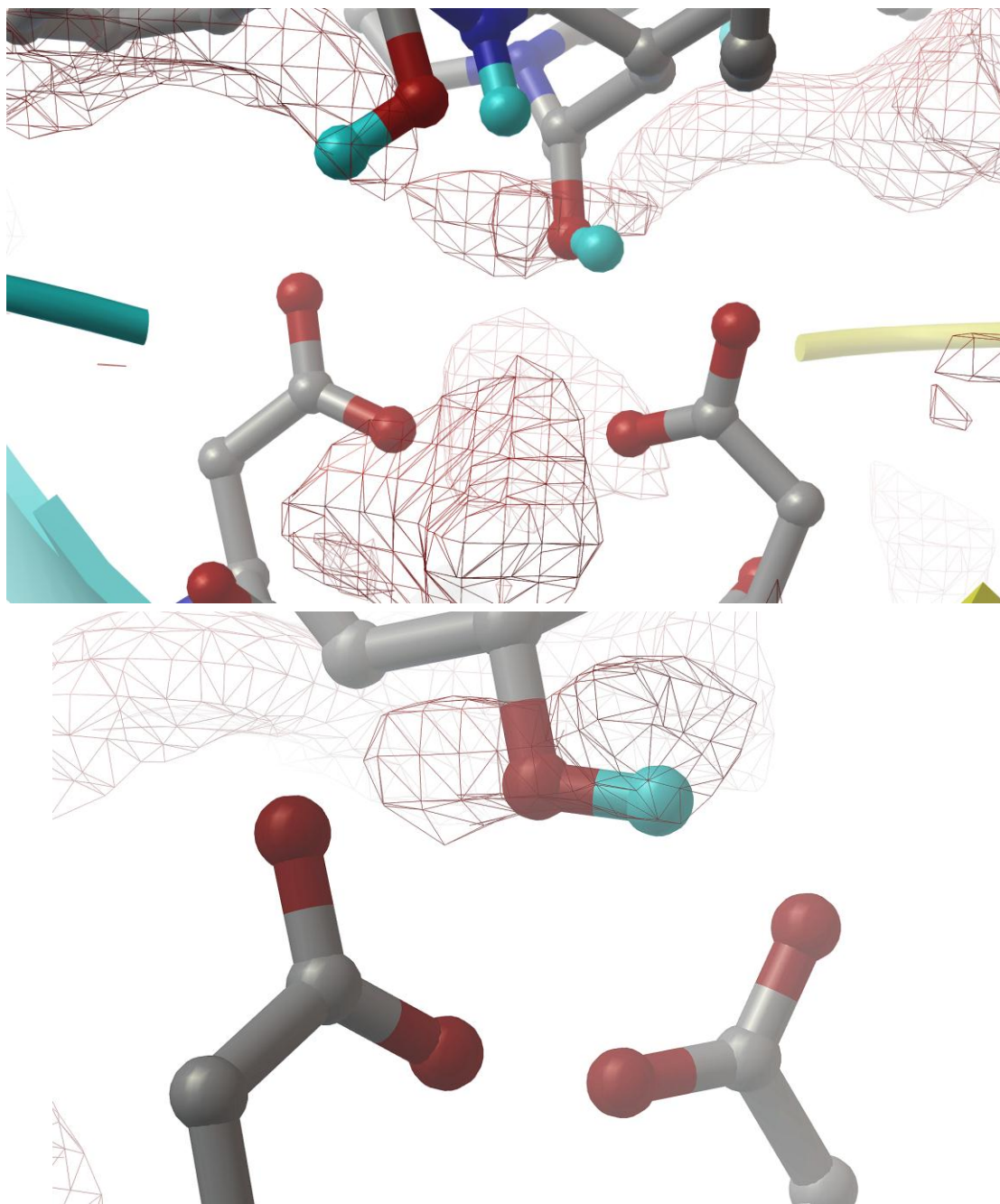


图 44: Receptor 的两个 ASP25 残基上的 O 与 Ligand 上 O 的相互作用

ADT 菜单: **Analyze** → **Dockings** → **Show as Spheres ...** 将对接得到的所有分子构象结果都以小球的形式显示。下图（图 45）中每一个小球表示该构象的几何中心，以方便不同构象之间的观察比较。

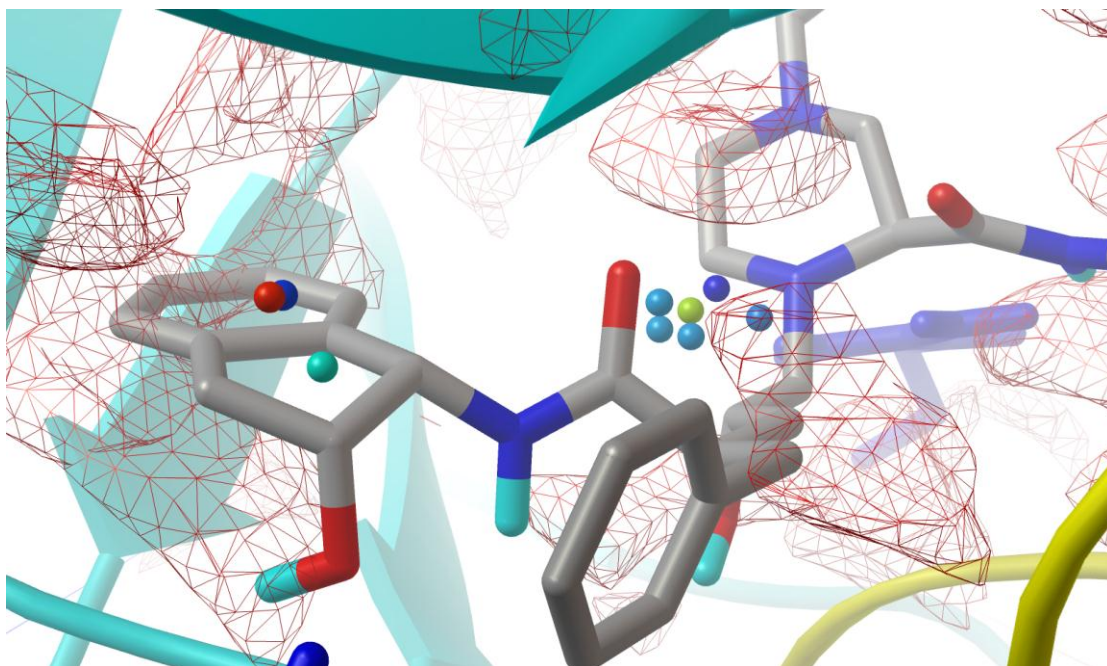
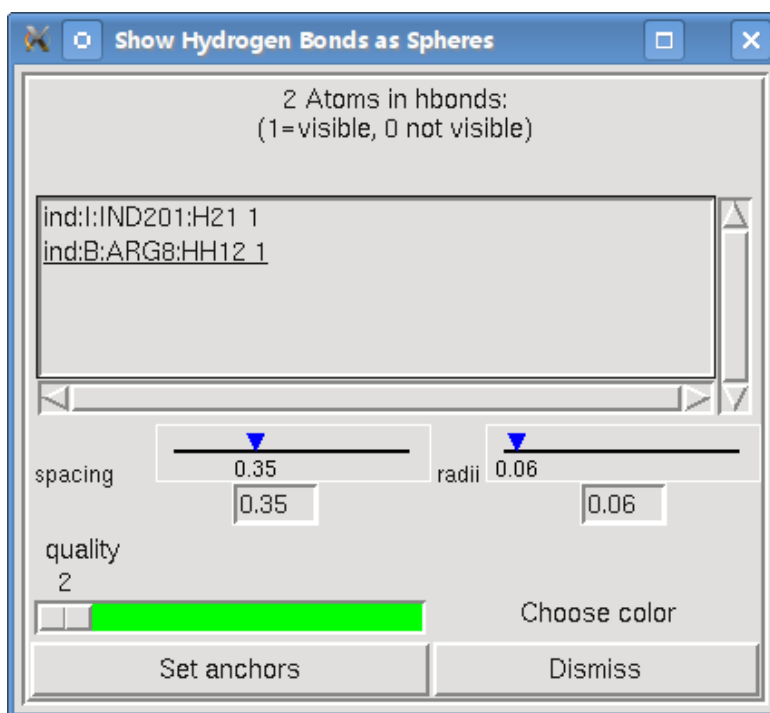


图 45: 一小球显示不同分子构象的显示效果

ADT 菜单: **Analyze** → **Dockings** → **Show Interactions** ADT 将自动计算并显示 Ligand 分子在当前构象下与周围 Receptor 残基之间的相互作用 (图 46, 47)。



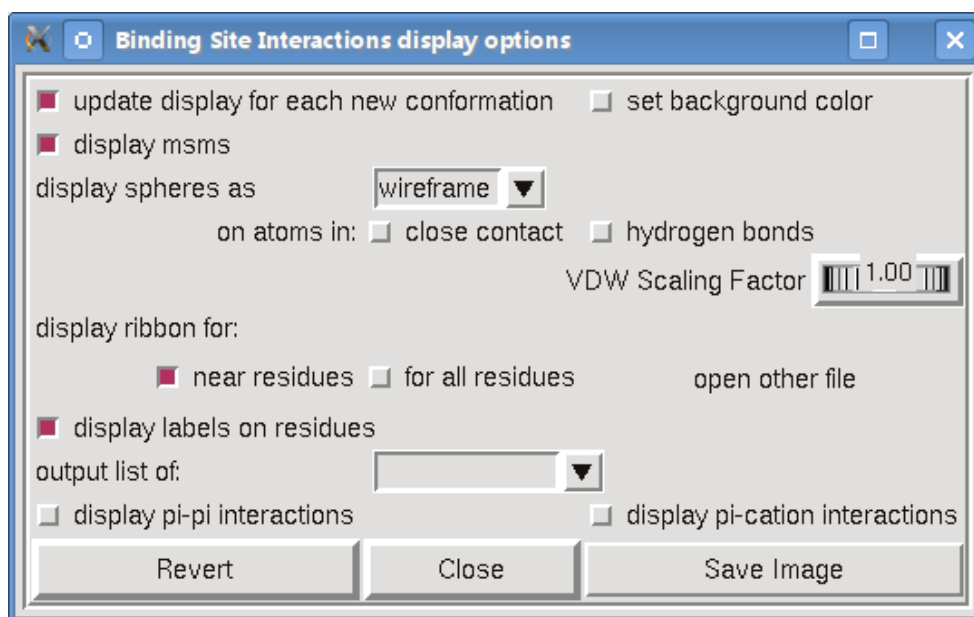
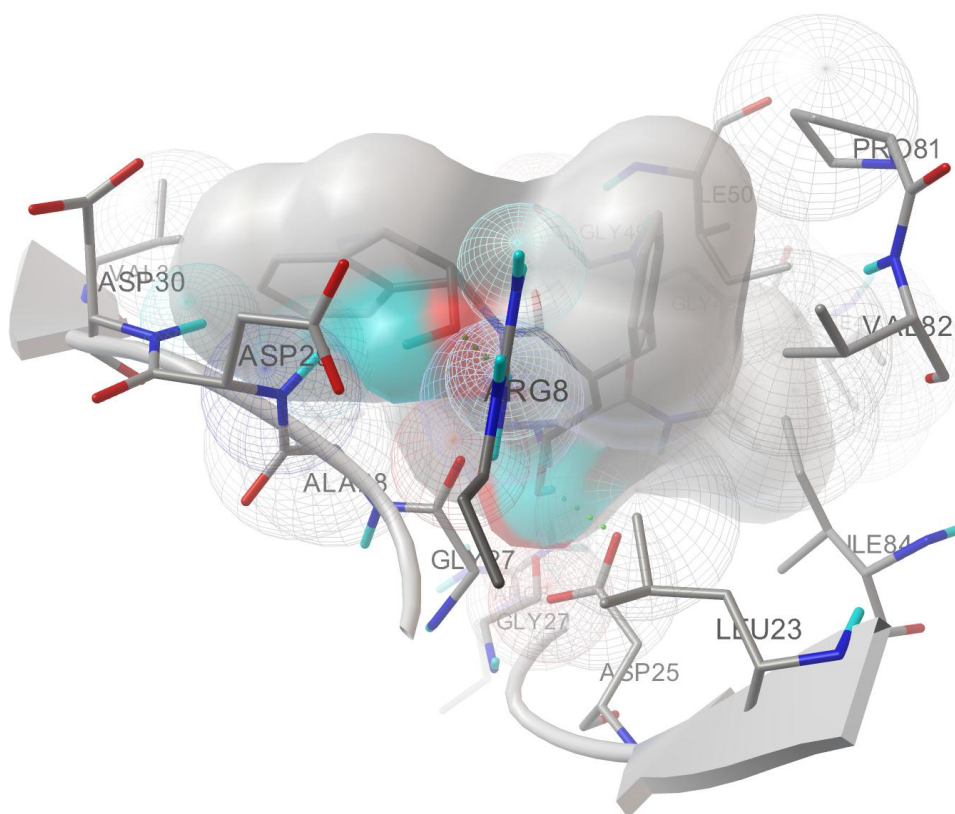


图 46: **Show Interactions** 设置对话框: **Show Hydrogen Bonds as Spheres** 以及 **Binding Site Interactions display options** 对话框



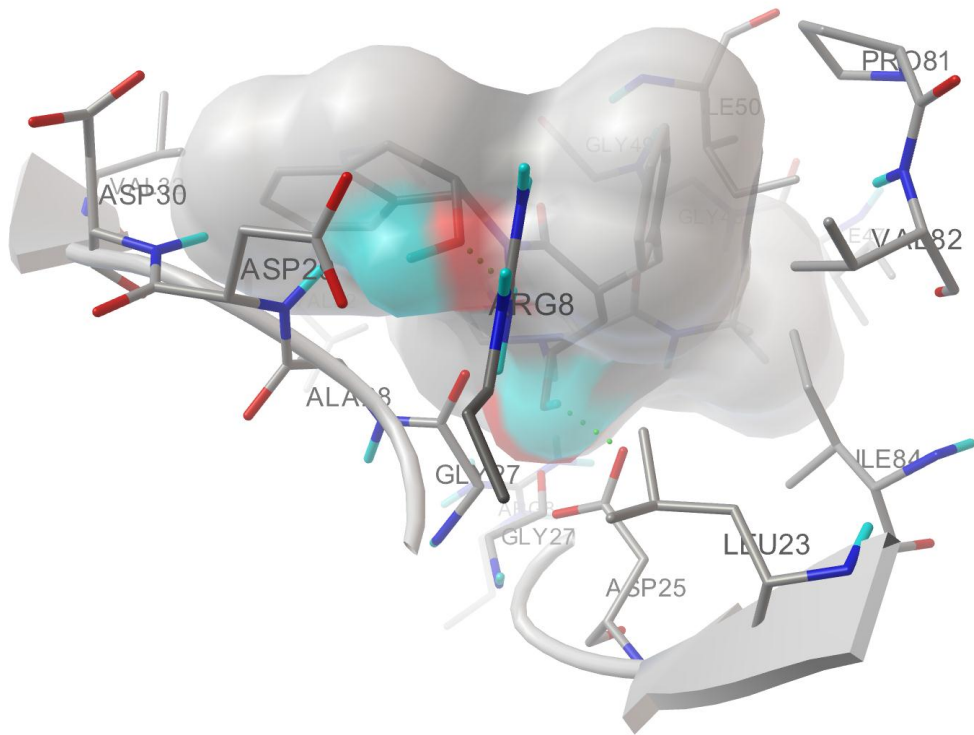


图 47: ADT 自动显示 Ligand 和 Receptor 相互作用的效果, 下图为关闭 wireframe spheres contact 效果