

A utility script for automating and integrating AutoDock and other associated programs for virtual screening.

Vikas Sharma¹, Kiran Kumar Pattanaik¹, Venkatesan Jayprakash², Arijit Basu^{2*}, Nibha Mishra²

¹Department of Computer Science and Engineering, ²Department of Pharmaceutical Sciences, Birla Institute of Technology, Mesra, Ranchi 835215, India; Arijit Basu – Email: abasu@bitmesra.ac.in, arijit4uin@gmail.com; Phone: +919430353750; *Corresponding author

Received April 26, 2009; Revised June 19, 2009; Accepted July 18, 2009; Published September 06, 2009

Abstract:

AutoDock is one of the most widely used docking programs but, is under utilized for the purpose of virtual screening. In order to overcome the difficulties that arise during virtual screening with AutoDock especially for GUI users an attempt is taken to automate the entire docking process, starting from grid map generation through ligand preparation and finally docking. Additional features from software like FILTER, OMEGA, ROCS and EON are also incorporated to create a complete virtual screening platform. The program completely reduced user intervention so that the program can be left unattended for longer period of time and there by improving the computational efficacy.

Availability: The script is available upon request from the authors

Keywords: AutoDock; FILTER; OMEGA; ROCS; EON; *Mycobacterium tuberculosis* thymidylate kinase; virtual screening.

Background:

This availability of protein structures in recent years led to the development of docking of ligands into the target binding site. Docking [1] which was pioneered during the early 1980's, models and estimates the recognition process between the protein and a ligand since then a significant development has been observed both in docking algorithm and scoring functions [2]. In modern drug discovery process High Throughput virtual screening (HTVS) or High Throughput Docking (HTD) plays an important role and often serves as a starting point to drug discovery process. Among the most widely used HTVS programs are Gold [3], Dock [4], Glide [5], FlexX [6], Fred [7], and LigandFit [8]. Most of the software that are specifically designed for virtual screening (VS) are commercial packages. Dock and Fred are exceptions which are freely available for academic purposes.

AutoDock is also free academic purpose, but it is under utilized for virtual screening as compared to other software, although it is one of the most frequently used docking tool for other purposes like blind-docking and binding mode analysis. The lack of popularity of AutoDock for HTVS is attributed to many reasons like: AutoDock consumes relatively more docking time as compared to other docking programs probably because of its relatively accurate docking algorithm and scoring function. Some practical difficulties also arise especially for GUI users, for example: each and every time the user has to repeat the protocol for ligand preparation and docking parameter file generation. Therefore the docking process cannot be left unattended by the user for any virtual screening experiment with the existing features of AutoDock. In the present work all these procedures are fully automated by a Unix script. Additionally features from software like FILTER, OMEGA, ROCS, EON and BABEL (<http://www.eyesopen.com/products/>) are also incorporated. The aims are mainly to minimize user intervention during the docking process so that the process can run unattended, as well as find a reasonable solution to the VS problem generally associated with AutoDock.

Features:

A dialog box within the script is created for providing a user interface. Some preinstalled softwares like (MGLTools1.5.0, AutoDock 4.0, AutoGrid 4.0, OPENBABEL-2.2.0, FILTER, OMEGA, ROCS and EON) are required. It is designed in such a way so that it provides a clear advantage over the limitations by AutoDock while performing virtual screening other associated programs were also incorporated along with AutoDock which screens the structures according to certain

similarity criteria and other user defined filters, there by reducing the redundancy and improving computational efficacy. However the user is free to skip any program which performs a similarity calculation or imposes any filter and can go directly for the docking which is again automated. The only input the script requires is the ligand database in sdf (or any other recognized file formats in BABEL) of the molecules and then it automatically processes the data in an interactive fashion. An interactive editing of the default parameters is provided. Before launching a fresh program and before loading new parameters it pauses for 10 seconds and asks the user for any alteration. For example while writing the docking parameter file/s (.dpf) it loads the default parameters and pause the program for 10 seconds enabling the user to edit any parameter value, once these values are loaded it is set default for every molecule.

The script provides an automated yet efficient mode for analyzing the results obtained after a virtual screening experiment. After the docking is over it converts all the docking conformers for every molecule into its corresponding pdb files it also appends all the dlg files and then summarizing and sorting them according to their scores. In order to validate the program a "lead-like" subset of 972608 compounds obtained from ZINC7 database was screened against *Mycobacterium tuberculosis* thymidylate kinase a potential target for anti tubercular drug development. A satisfactory performance is observed while performing the process of virtual screening locally on a single system with a dataset of 972608 compounds (**unpublished results**) however; its validation on a Grid platform is under development.

Methodology:

Software:

The script runs on a linux platform in bash shell and requires some preinstalled softwares like MGLTools1.5.0, AutoDock 4.0, AutoGrid 4.0, OPENBABEL-2.2.0, FILTER, OMEGA, ROCS and EON.

Scripting:

Unix based Shell and Perl scripting is used in the current work. Dialog command is used to create the various dialog boxes wherever required in order to make the utility more user friendly. Based on the choices in the MAIN DIALOG BOX some SUB DIALOG BOXES are created as given in **Figure 1**. The main dialog box is created as shown in supplementary material.

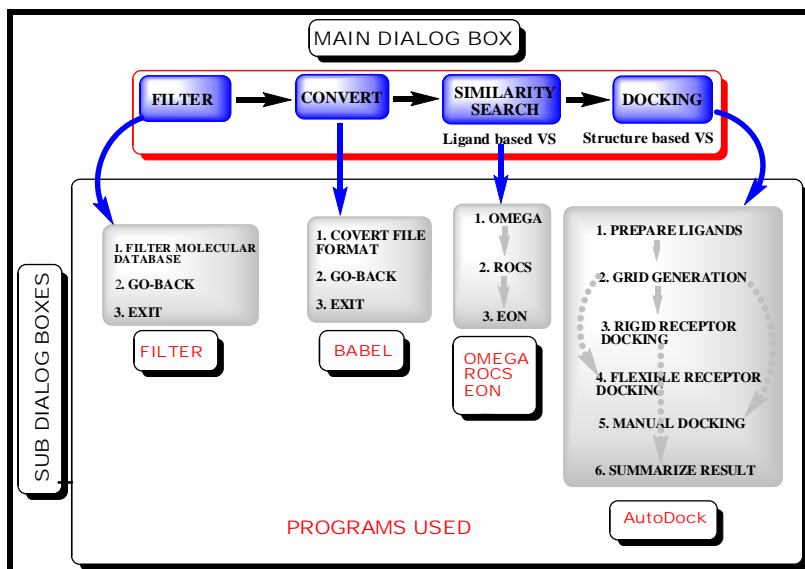


Figure1: Flowchart showing the various dialog boxes depicting the workflow and corresponding programs used to build the script. Four main protocols for VS are included which are (1) Filtering Molecular databases; (2) Conversion of file formats, (3) Ligand based similarity search, (4) Structure based docking.

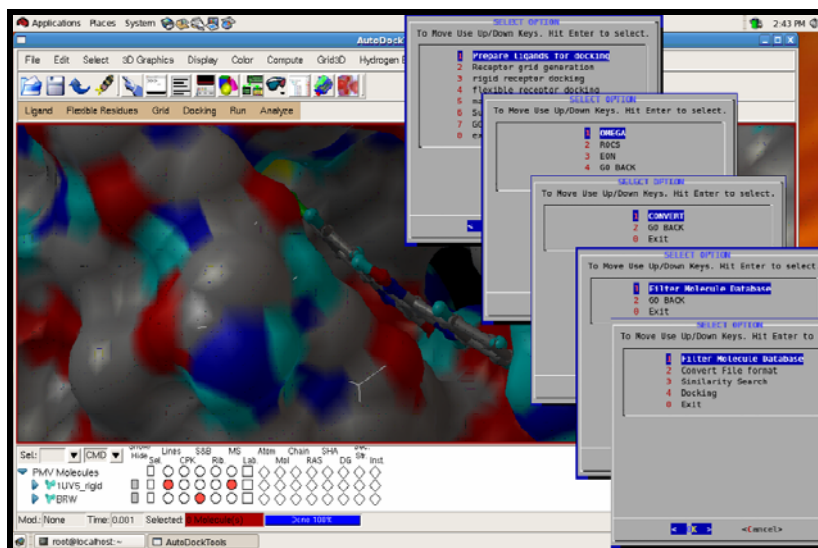


Figure2: A LAYOUT of how the script looks after execution. The various dialog boxes one after the other with MGL Tools window in the background.

Acknowledgments:

The authors are grateful to AICTE (All India Council for Technical Education) for the financial aid. The Scripps Research Institute and OpenEye Scientific Software are also acknowledged for providing the necessary software absolutely free to carry out the research.

References:

[1] ID Kuntz *et al.*, *J. Mol. Biol.* **161**:269 (1982).
 [2] N Moitessier *et al.*, *Brit. J. Pharmacol.* **153**:S7 (2008).

[3] G Jones *et al.*, *J. Mol. Biol.* **267**:727 (1997).
 [4] DT Moustakas *et al.*, *J. Comput. Aided. Mol. Des.* **20**: 601 (2006).
 [5] TA Halgren *et al.*, *J. Med. Chem.* **47**:1750 (2004).
 [6] B Kramer, *Proteins* **37**:228 (1999).
 [7] MR McGann *et al.*, *Biopolymers* **68**:76 (2003).
 [8] CM Venkatachalam *et al.*, *J. Mol. Graph. Model.* **21**:289 (2003).

Edited by P. Kanguane

Citation: Sharma *et al.*, *Bioinformatics* 4(2): 84-86 (2009)

License statement: This is an open-access article, which permits unrestricted use, distribution, and reproduction in any medium, for non-commercial purposes, provided the original author and source are credited.

Supplementary material

```
main_menu()
{
dialog --clear --backtitle "..DRUG DESIGN PLATFORM.." --title "SELECT OPTION" \
--menu "To Move Use Up/Down Keys. Hit Enter to select." 20 50 5 \
"1" "Filter Molecule Database" \
"2" "Convert File format" \
"3" "Similarity Search" \
"4" "Docking" \
"0" "Exit"          2>user.choice.tmp.$$
btnn=$?
case $btnn in
0)echo You Selected Choice \#`cat user.choice.tmp.$$`;
1)btnn=1;echo 0 >user.choice.tmp.$$; #cancel
255)btnn=1;echo 0 >user.choice.tmp.$$;#ESc
esac
}
```

Likewise other sub-boxes are also created depending on the choices. Here are some examples about how all the commands are compiled into a script:

```
Choice1 "Filter Molecule Database"
filter -in "$ifile"."$ifmt" -out "$ofile"."$ofmt" $option -table "$ofile"_table.txt -tableFlag
```

```
Choice2 "Convert File format"
babel -i"$ifmt" $ifile.$ifmt -o"$ofmt" $osuff.$ofmt
```

```
Choice3 "Similarity Search"
omega2-2.2.1 -in "$IPREFIX"_"$i"."$IFMT" -out "$OPREFIX"_"$i"."$OFMT" -prefix "$OPREFIX"_"$i"
```

```
rocs -dbase "$ROCSIDBN"_"$i"_omega."$ROCSIDBF" -query "$ROCSQFN"."$ROCSQFF" -outputquery "$ROCSTF" -out "$OFMT" -prefix "$PREFIX"_"$i"
```

```
eon -dbase "$EONIDBN"_"$i"_eon_input_1."$EONIDBF" -query "$EONQFN"."$EONQFF" -ofmt "$OFMT" -outputquery "$EONTF" -prefix "$PREFIX"_"$i" -besthits "$EONBH"
```

```
Choice4 "Docking"
prepare_ligand4.py -l $i -A bonds -A hydrogens -A bonds_hydrogens
autogrid4 -p "$r_stem".gpf -l "$r_stem".glg
autodock4 -p "$dpf" -l "$l_stem".dlg
```