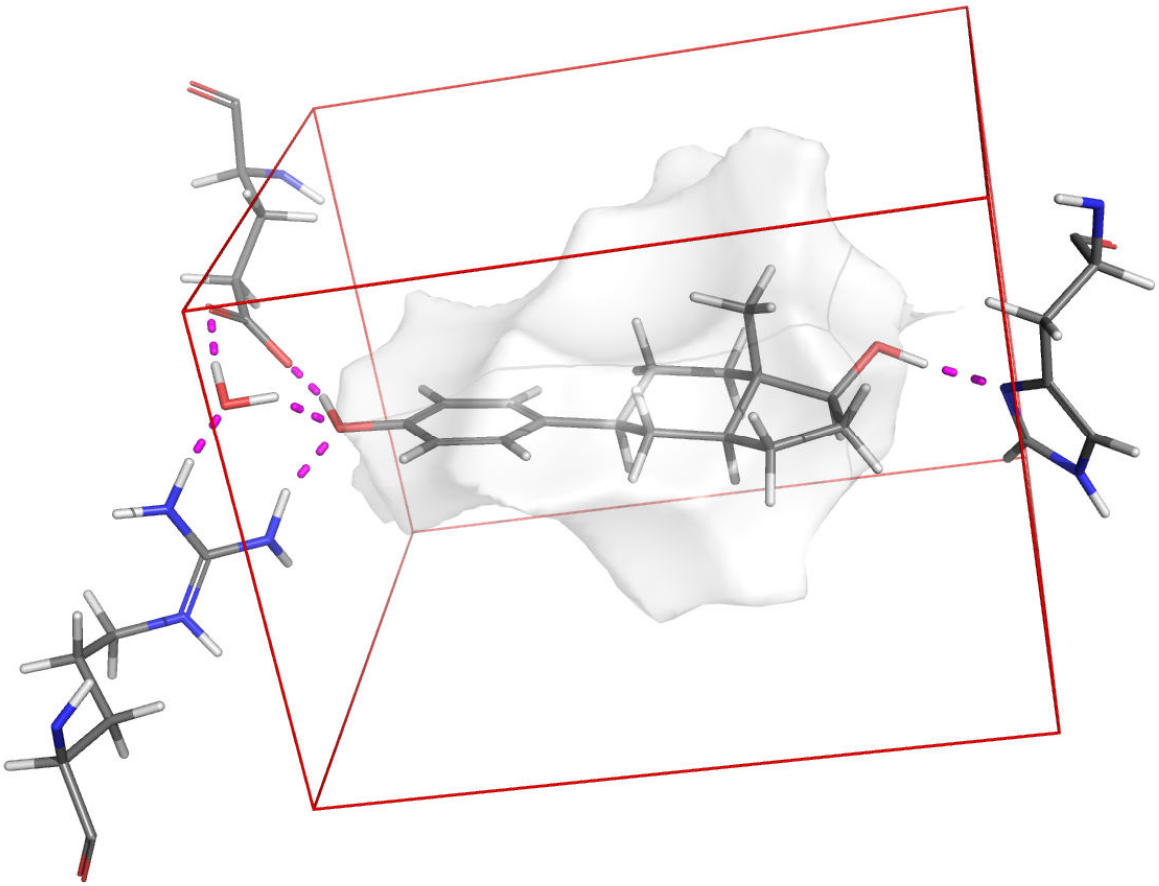


uDock 2.5

flexible docking program for MOE/smp clusters

User Manual

H.Shadnia Sep 2009

Contents

Part A

Introduction

uDock is a flexible ligand-receptor docking program designed for MOE (www.compchem.com). This program is available for download from www.shadnia.com. The main idea behind uDock is to reduce the number of commonly used and potentially invalid approximations and cosmetic procedures. In return, the program requires considerable processing power.

The main workflow in uDock is shown in appendices I,II. In order to run a docking job, three pieces of data are required:

a- *The structure of receptor*
This is prepared from the raw crystal coordinates using the **PDB-Thaw** program and is saved as a *.moe file. (see section 2)

b- *The coordinates of the docking box*
This is prepared using the crystal coordinates and the '**DockingBox**' program. It is saved as a *.mdb file. (see section 3)

c- *The ligand database*
This includes all plausible conformers (and sometime even rotamers) of the ligands to be docked into the active site. It can be prepared in different ways and is saved as a *.mdb file. It also has to be properly prepared or 'tagged' using **H_DB_Prepare** program. (see section 4)

After the three files were prepared, a job 'launcher' job named **H_Dock.svl** is edited to reflect the correct filenames, file paths, and some other user settings. Then the docking will be run on a single machine or a windows / Linux cluster.

The full flowchart of docking is shown in Appendix II. Briefly, random conformations of the each ligand database entry (individual conformers) are placed inside the docking box, the geometry is optimized and the energies are recorded before, during and at the end of energy minimization. This is done for a user defined number of time (usually 10~20 times) which is named the 'training' phase. After that, the program finds typical and minimum energy values for the given complex and tightens the acceptable energy thresholds for the rest of calculations on the given ligand conformer. The energy calculations on ligand 'poses' which are too high energy will be aborted to save the processing time. The program will keep generating a user-defined number of 'good' solutions for each ligand-conformer and proceeds to the next entry when done.

The energy minimization steps and the energy criteria used to abort or accept a given pose can be easily reconfigured for a wide range of applications. For most purposes however, the dynamical 'Autopilot' system with the default values is efficient. uDock emphasizes in achieving convergence for each complex.

The docking engine produces a raw output database which includes many conformations of complexes of ligand-target, with detailed energetics recorded. uDock does not perform any cosmetic 'cleanup' on this database.

A set of 'post processing' programs then help to find the best complex, and read the energetics and perform quantitative calibration or prediction of activities (Section 6 and appendix iii).

Theory

After running extensive tests on different targets, the following rules were used to design uDock:

i – The initial energy of a randomly generated complex has NO meaningful correlation to the final (energy minimized) value. This is simply because even one single VdW clash between two hydrogens can raise the energy to hundreds of thousands of kcals/mol. But after a few iterations of energy minimization, these clashes will relieve and the energy becomes more stable, and comparable to the final geometry.

ii- The receptor active site can not be treated as rigid (due to the same reason).

iii- Explicit water molecules have to be maintained and treated properly (esp those which make a 'water bridge' between two functional groups. In general they need to be re-arranged or re-docked for each individual complex. uDock requires the explicit waters to be present but it doesn't 're-dock' them by default.

iv- Due to the 'gearing effect' [ref] number of possible conformations or 'poses' for the simplest ligand-receptor or even host-guest complexes can easy exceed thousands. For this reason uDock tries at least 2~5000 'poses' for each ligand. This is crucial to ensure reaching convergence, i.e. subsequent runs of the program will not result in a lower energy complex.

Some programs apply 'smoothing functions' to VdW terms in order to bypass this effect. Such measures artificially reduce the complexity of conformational space, but they often diminish the uniqueness of the active ligands which perfectly fit the grooves of the target active site.

vi- Each conformer of the ligand is a unique chemical entity which may or may not fit to a target. For this reason all (room temperature) conformers of each ligand need to be individually docked into the active site. In rare cases where a particular moiety of the receptor is extraordinarily complex, even different rotamers need to be tried. When docking macromolecular structures as peptides, since it isn't possible to generate all plausible conformers, a reasonable number of such conformations generated by MD may be used.

vii- In theory, the entire conformational space of the target protein need to be explored for each individual 'pose', meaning that the rotatable side chains of the active site residues need to be actively rotated. The current version of uDock does not perform this, but the program is designed to allow future implementation of this feature. For many simple active sites (sometimes called 'rigid active sites' as those of nuclear receptors) this is not crucial.

viii- The raw crystal coordinates need to be treated properly to generate an energy minimized structure that is as faithful as possible to the raw coordinates. Three issues are remarkable in this topic:

1 – A regular energy minimization on raw crystal geometry causes unacceptable amount of perturbation in the geometry, esp. that of buried active sites. For this reason, the 'PDB Thawing' program (H_PDB_THAW see www.shadnia.com/H_Thaw) should be used to energy minimize the initial PDB file.

2- The limited precision of Cartesian system causes small random perturbations of the macromolecular geometry independent of ligand binding. For this reason, the energy read outs in uDock are by default limited to the active site residues (the 1st shell or S1) which are kept fully flexible. The amino acid residues surrounding the active site (labeled as the 2nd shell or S2) are kept semi-flexible and their energies are excluded from the final energy calculations. The Atoms or residues surrounding S2 are kept rigid and can be deleted to accelerate the calculations. The PDB Thawing program defines these shells, but they can be manually modified too. The definition of these shells are saved within the 'receptor' file that is generated by PDB-Thaw.

3- When comparing different instances of raw crystal structures of supposedly identical complexes, conformational difference is observed which leads to variation in docking results. A calibration test can be used to identify and filter out the 'ill-modeled' residues in each instance of crystal structure. This results in dramatic improvement of the docking results [H. Shadnia, J.S. Wright to be published]

PART B

1 – Program configuration

Purpose: installation and configuration of MOE and uDock.

1-a: Program files

The following is the description of the individual programs:

Filename	description
H_Dock.svl	launcher: configure and launch the docking jobs
H_Docker.svl	Host program
H_Docker_Client.svl	Client program
H_DockingBox.svl	Graphical interface to define the docking box
H_Host_Tools.svl	Host utilities
H_Toolbox.svl	Main utilities
H_PDB_Thaw.svl	Graphical PDB-Thaw program
H_DB_Prepare.svl	Graphical program to prepare (tag) the input database
H_DB_PickMin	utility to pick the lowest energy conformers
H_DB_PicknMin	utility to pick *n* lowest energy conformers
H_Post_Process	post processing tools
H_IF-E.svl	Program to analyze interaction forces and energies

1-b: Setup the environment variable:

For 'bash' edit the file `.bashrc` and add the following lines:

```
uDock="~/udock/svl"
export uDock
```

For other types of shells refer to the operating system manual.

1-c: MOE should be installed and configured to run in SMP mode.

Please refer to MOE manual on '***Installing and Running MOE/smp***'.

1-d: (optional) configure the grid engine (load-balancing system)

1-d:-I: Create a sun-grid-engine file name 'moe.sge' that contains the job creation details:

```
#!/bin/bash
#
# CHOOSE THE NUMBER OF PROCESSOR WHEN YOU RUN THE JOB.
# FOR EXAMPLE TO USE 8 PROCESSORS, TYPE:
#     qsub -pe  moe 8 moe.sge
#
#
```

```
# Tell Gridengine what shell to use:
#$ -S /bin/bash
#
# Start this script from the current working directory:
#$ -cwd
#
# Combine output and error messages into one output file:
#$ -j y
#
# Name of the output file:
#$ -o moe.$JOB_ID.err
#
echo -n "I am running: "
echo $INPUT
/opt/moe/bin/moebatch  -mpu $TMPDIR/moemachines -mpulog moe.$JOB_ID.log -run $INPUT
```

1-d-II: create a job-generation script named 'runMoe' as follows:

```
#!/bin/bash
#  usage: runMoe numberOfProcessors inputFile
qsub -v NS=$1,INPUT=$2 -pe moe $1 /opt/moe/bin/moe.sge
```

2- Preparing the target structure

Purpose: to prepare an energy minimized model of the full or truncated macromolecular target.

2-a: Provide the 3D coordinates

Download the coordinates file from the protein databank (RCSB.ORG) or use the homology modeling to build it.

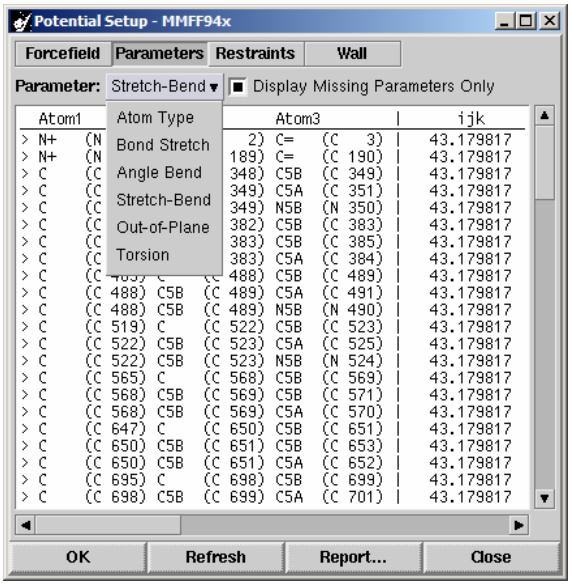
2-b: Add hydrogens

In case the hydrogen atoms are missing, add them using MOE>Edit>Hydrogens>Add Hydrogens or use MOE>Compute>Protonate 3D

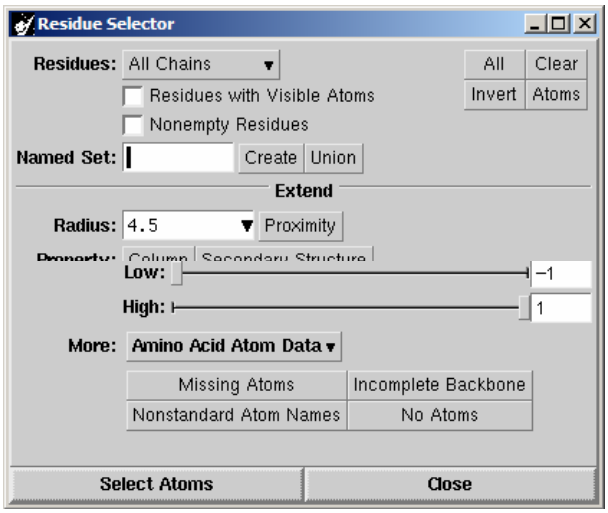
2-c: 'Clean up' the structure:

3-c-I: Inspect atom types

After selecting the appropriate force-field (MOE>Window>Potential Setup >Force Field), check the for missing parameters to ensure there are no illegal atom types or undefined force-field parameters. To do this, use MOE>Window>Potential Setup> Parameters, check 'Display Missing Parameters Only' and under 'Parameter' see each individual parameter type, most importantly 'Atom Type, Bond-Stretch, Angle Bend,….' (Fig 1)



2-c-II: Inspect missing atoms
Load the 'sequence editor' (Main menu > SEQ)
Enable synchronized selection (Sequence Editor > Selection > Synchronize)
Load the 'Residue Selector' (Sequence Editor>Selection>Residue Selector)
Under 'More' select 'Amino Acid Atom Data'
Click on each of four classes of missing data to find the missing atoms (see below). Correct the missing atoms if any.

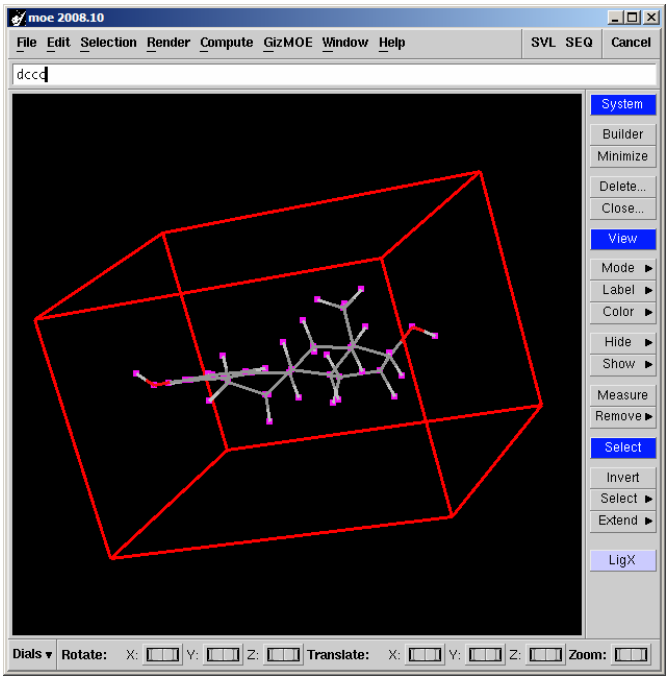


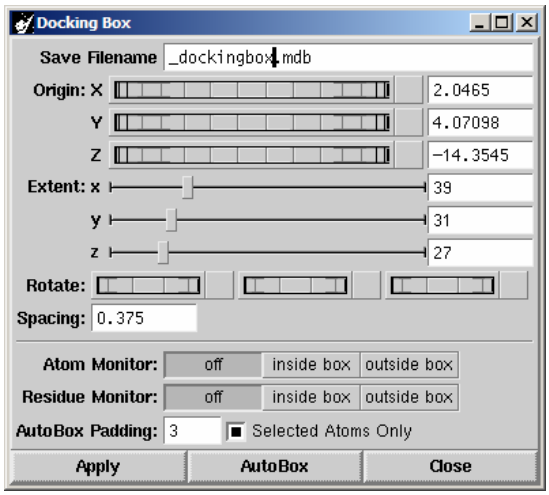
2-c-III: Delete unrelated chains
Many PDB files include crystallographic or biological dimmers, trimers, or polymers. They might include co-crystallized small organic molecules or metal ions. Delete all unrelated structures.
2-d: run the Thawing program.
Select the ligand on the screen and run the 'H_PDB_Thaw.svl'. Select the appropriate protocol and click 'OK'. Refer to http://www.shadnia.com/H_Thaw for more information
3-e: open the thawed file (*_thawed.moe) and delete the ligand. Save this file as '_receptor.moe' or any other name. This is the main 'target' file that will be used for docking.

3- Preparing the docking box

Purpose: to define a virtual box that contains the active site, random conformers of the ligands will be generated inside this box during the docking.

3-a. Load the 'thawed' protein file (*_thawed.moe)
3-b. Render the active site boundaries [See tutorial A for an example]
3-c. Run the docking box program 'H_DockingBox.svl'. Modify the dimensions if necessary and save the final box. (See below)





4- Preparing the ligand database

Purpose: to prepare a 'tagged' database of all conformers/rotamers of all ligands to be docked into the receptor.

4-a: build conformers of each ligand

3-a-I. Use the molecular builder module or other ways to draw / import the structure of the ligand.

3-a-II. Explicit list of conformers/rotamers for each ligand should be prepared. Use the systematic conformer search, stochastic conformer search, or even MD simulation to build an MDB containing the conformers for each ligand. [See tutorial B for an example]

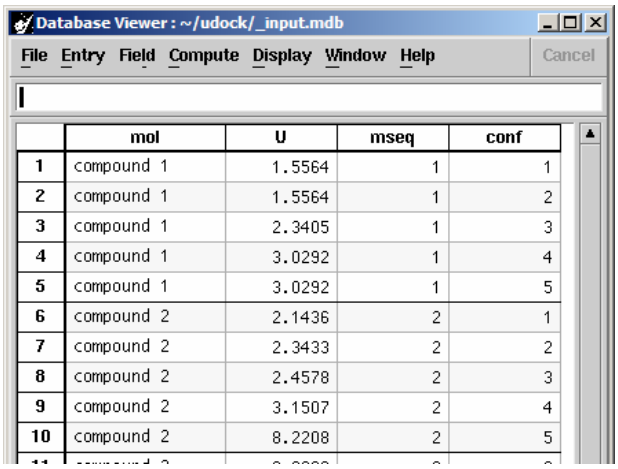
4-b: merge multiple ligands into one database

After preparing individual conformers, use the database merge tool [Database Viewer > File > Merge] to merge all individual files into one large database.

4-c: prepare (or 'tag') the database

Use the H_DB_Prepare.svl program to 'tag' the large database. Tagging assigns ligand numbers (specified in the 'mseq' column) and conformer numbers (specified in 'conf' column) to the

database. uDock uses these tags to monitor different procedures. An example of prepared and tagged input database is shown here.



	mol	U	mseq	conf
1	compound 1	1.5564	1	1
2	compound 1	1.5564	1	2
3	compound 1	2.3405	1	3
4	compound 1	3.0292	1	4
5	compound 1	3.0292	1	5
6	compound 2	2.1436	2	1
7	compound 2	2.3433	2	2
8	compound 2	2.4578	2	3
9	compound 2	3.1507	2	4
10	compound 2	8.2208	2	5

5- Job setup and run

5-a: Open the launch program 'H_Dock.svl' using the text editor or any other text editor.
5-b: edit the 'options' to reflect the correct filenames and desired options. Save the svl file.
5-c: run the launcher program in MOE/smp mode. See MOE's manual on 'Installing and Running MOE/smp'.

5-d how to run the launcher program for HPCVL grid-engine:
5-d-I: To run the program on 24 processors use the following command:

```
runMoe      24    H_Dock.svl
```

5-d-II: To monitor the job use 'qstat'

The following shows an example of the docking job running on 24 processors.
The job has been assigned a job-ID of 5285. It can be killed using 'qdel 5285'.

```
job-ID  prior   name       user         state submit/start at     queue                          slots ja-task-ID
-----------------------------------------------------------------------------------------------------------------
5285 0.55500 moe.sge    hpc1764      r     09/11/2009 13:23:48 all.q@compute-0-21.local          24
```

5-d-III.To view the docking 'log'
When docking is running, it creates two files which are named by job-ID as explained above. the error file contains all debugging messages that would be visible on MOE's SVL window in graphical mode. This file would be named 'moe.5285.err' for the previous example. You can monitor the progress of docking by viewing the contents of this file using 'cat' or 'more' commands or any other method. The following is an example: (comments are added in italics):

```
running uDock using the following options:        (input parameters)
[ _program_path:'~/udock/svl/', _project_path:'~/udock/', lig_dbname:'_input.mdb',
output_dbname:'_output.mdb', target_filename:'_receptor_ERa.moe',
docking_box_filename:'_dockingbox_ERa.mdb', _target_good_poses:40, _target_bad_poses:3000,
lig_max_retries:2000, _auto_pilot:'ON', _training_iterations:10, _Threshold_Percent:60,
_forcefield:'$MOE/lib/mmff94s.ff' ]

file transfer adapted to OS: unix                 (automatically assigned)

========================= START  ==================================
```

```
running on 64 clients:                              (number and list of processing nodes)
compute-1-15
…
compute-1-12.local
================== starting the job spool ========================
        running the remote SVL loader               (loading program on processing node)
(v)     OK loading "~/udock/svl/H_Docker_Client.svl" on Clients: .....
(v)     OK loading "~/udock/svl/H_Toolbox.svl" on Clients: .....
        Loading Codes on all Clients successful!

 >>  Scanning the input database: entry 1 of 5     (first ligand/conformer)
 !   Training mode: ON                            (using 10000kcal/mol limit for energies)
(c)     try #0 to make a good initial pose, max3000 (generate the ligand inside the box)
(c)     rotating .........
(c)     centring .........
(c)     discarding........                          (did not fit in the box)
(c)     try #1 to make a good initial pose, max3000
(c)     rotating .........
(c)     centring .........
(c)     good pose found...                          (did fit in the box)
 :|  compute-1-29.local => training (total:1)
.....
 :|  compute-1-18.local => training (total:9)
 :|  compute-0-18.local => training (total:10)
 [*] Activating AutoPilot                           (calculating tight energy thresholds)
          == > updating the threshold for task#7 idx=3,value=100000 to 21613.4
          == > updating the threshold for task#10 idx=3,value=100000 to 19.2877
          == > updating the threshold for task#13 idx=3,value=100000 to -1.13228
 training finished, killing all current tasks       (start over with tight thresholds)
 :)  compute-1-13.local => good pose (total:1)      (good results to be saved in output)
 :)  compute-0-3.local => good pose (total:2)
 :)  compute-0-18.local => good pose (total:3)
:(  compute-1-30.local => bad pose (total:7)        (bad results will be discarded)
....
 :)  compute-1-5.local => good pose (total:39)
 :)  compute-0-13.local => good pose (total:40)
 *  finished entry #1 in [197] secs, estimated remaining:[788] secs(elapsed\estimated time)
    good:  40 [39.2%] , bad: 62 [60.8%], total= 102                (overall statistics)
 >>  Scanning the input database: entry 2 of 5     (staring up with the next ligand)
....
++   DONE! finished scanning the input database    (the last ligand is sent to the cluster)
   >    scanning the input db finished - job spool done
================== Job Spool : Done ========================
        Search finished! Aborting further calcs on entry #6

================================================
=  Computations complete                        (all calculations are done)
================================================
 total calculation time: 1172.0 seconds                          (total time)
 good: 200 [30.9%] , bad: 448 [69.1%], wasted:0 , total:648      (performance stats)
```

The second file contains MOE's OS level messages and can be used for troubleshooting of MOE/smp network. This file would be named 'moe.5285.log' for the previous example.

6- Post processing

6-a: Understanding the output database

To allow for method development, uDock saves a raw database of the docked poses. The database contains the following fields:

mseq	ligand number	same as the input database

Conf	conformer number //
str_1	receptor coordinates at the complex configuration
str_2	ligand coordinates at the complex configuration

Thus the most important energies are the ones at the end of the energy minimization (those of step 4 and 5):

R_4_sel	The internal energy of the ligand at the bound conformation ($E_{L'}$)
R_4_int	The interaction energy of the ligand vs. target (E_{int})
R_5_sel	The internal energy of the (pseudo) complex. See below for details.

6-b: Quantitative prediction

To allow for fundamental analysis of the results, the raw system energies are recorded in uDock. This is done in 5 steps, step 1: before any energy minimization, step 2 after 20 iterations of energy minimization, step 2: after 20 additional iterations, step 3: after 40 more iterations (final energies). Steps 4,5 at the end of energy minimization. Each energy reading generates three sets of numbers designated by suffices: *_all*, *_sel*, and *_int* fields which correspond to the total system energy, the internal energy of the **selected atoms** and the interaction energy of the selected atoms vs. the rest of the system. Through step 1 to 4 the ligand structure is selected, and at step 5 the ligand plus the first shell of amino-acids (active site) are selected.

By definition, the interaction energy is the difference in energy of products (complex) from those of the reactants (ligand and receptor) **in the conformation they hold** upon binding:

$E_{int} = E_C - (E_{R'} + E_{L'})$ [1]

In comparison, the binding energy ($\Delta E_{binding}$) is the difference in energy of products (complex) from those of the **free** ligand and receptor:

$\Delta E_{binding} = E_C - (E_R + E_L)$ [2]

The free ligand and receptor (L, R) are deformed upon binding to form the deformed ligand and receptor (L', R'), thus the deformation energies (ΔE_L, ΔE_R) are defined as:

$\Delta E_L = E_{L'} - E_L$ [3]

$\Delta E_R = E_{R'} - E_R$ [4]

Thus, from eq. 1: $E_C = E_{int} + E_{R'} + E_{L'}$

From eq.3: $E_L = E_{L'} - \Delta E_L$

From eq.4: $E_R = E_{R'} - \Delta E_R$

Now using eq.2:

$\Delta E_{binding} = E_{int} + E_{R'} + E_{L'} - (E_{R'} - \Delta E_R) - (E_{L'} - \Delta E_L)$

$= E_{int} + E_{R'} + E_{L'} - E_{R'} + \Delta E_R - E_{L'} + \Delta E_L$

$= E_{int} + \Delta E_R + \Delta E_L$

Thus, three terms from the docking describe the binding energy

1- E_{int} is the ligand interaction energy designated as R_4_int in the raw database.

2- ΔE_R The receptor deformation energy: since the energy of free receptor (E_R) and is arguably difficult to calculate (since it requires explicit solvation of the active site and more importantly it's a constant number; from eq.3 the following approximation can be made:

$\Delta E_R = E_{R'} - const$ [5]

3- ΔE_L The ligand deformation energy: this can be calculated by subtracting the energy of ligand in complex by that of the free ligand according to eq.3 .

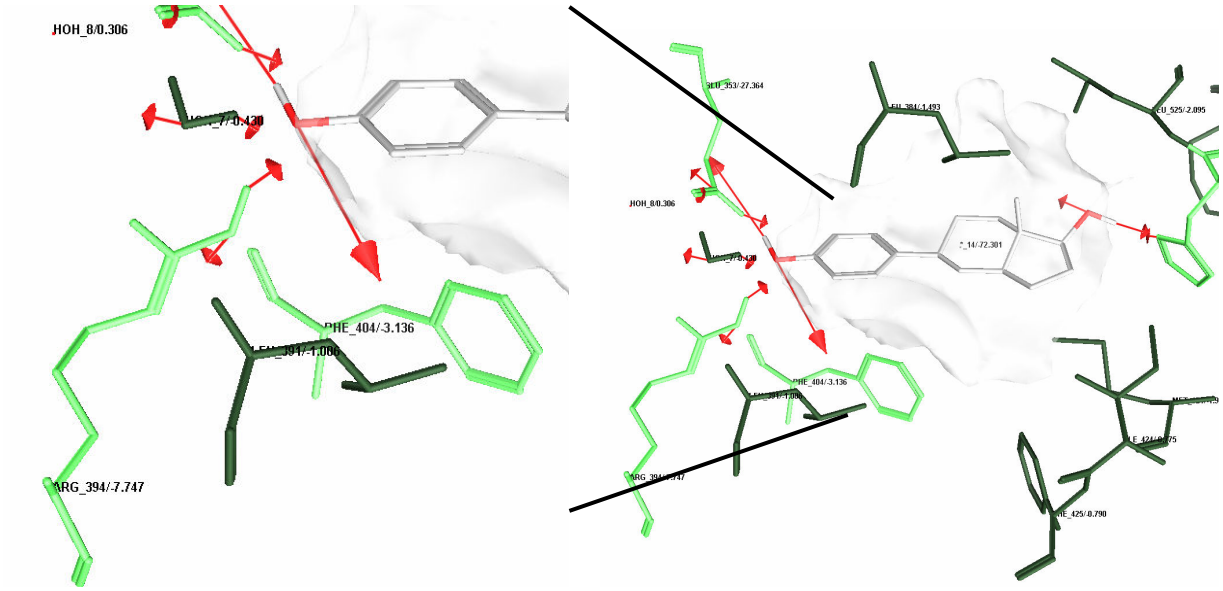
Two more parameters are required to describe the binding energies

4- ΔG_{solv} The ligand (de)solvation energy. This can be calculated using Gaussian etc.

5- $T\Delta S$ Currently there is no easy way to calculate this. When all ligands in a dataset cause extraction of same number of water molecules from the active site, this term can be relatively constant. In other cases, for example when a small molecule as phenol is compared to a larger molecule, this term cannot be ignored. Our group in collaboration with CCG is developing a method to estimate this term (Oct 2009).

6-c: The detailed analysis of interaction energies

We developed a program to analyze the interaction energy of individual amino acids with the active site residues. The program calculates and visualizes the interaction energies as well as forces which allow for detailed analysis and understanding of molecular events for a given complex. See www.shadnia.com/H_IFE for more details and download.



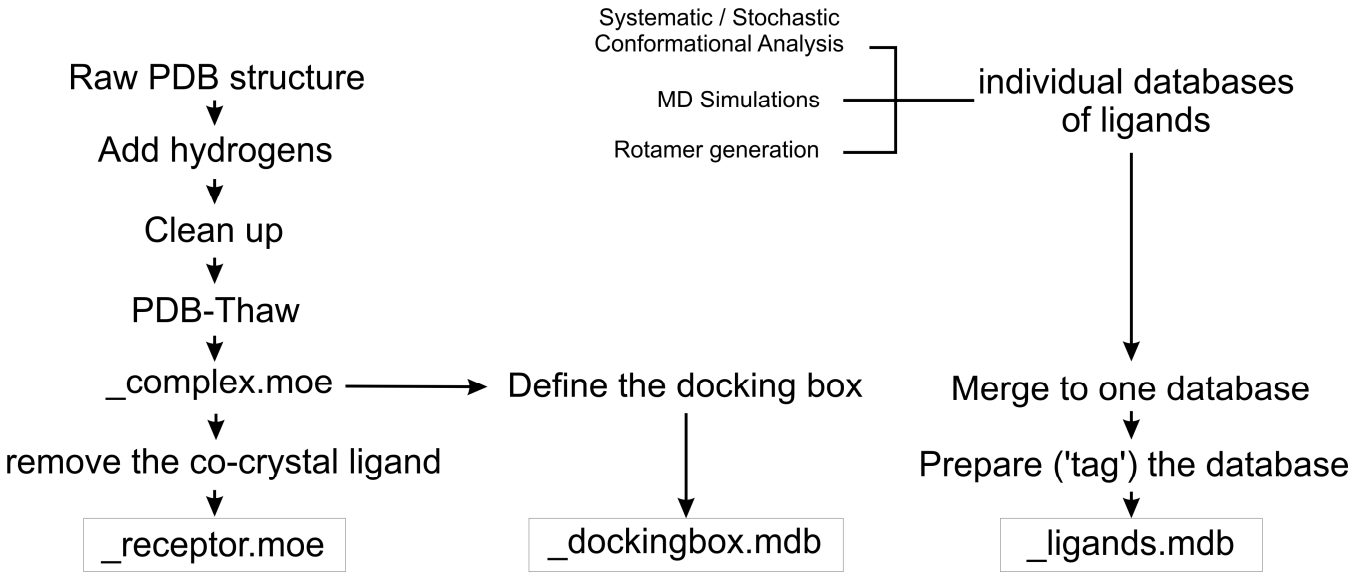
The analysis of the interaction energies and forces

The red arrows show the atomic forces. The favorable interactions are shown in dark green (0 to -4 kcal/mol) and light green (stronger than -4 kcal/mol). The magnitudes of interactions are displayed as labels.

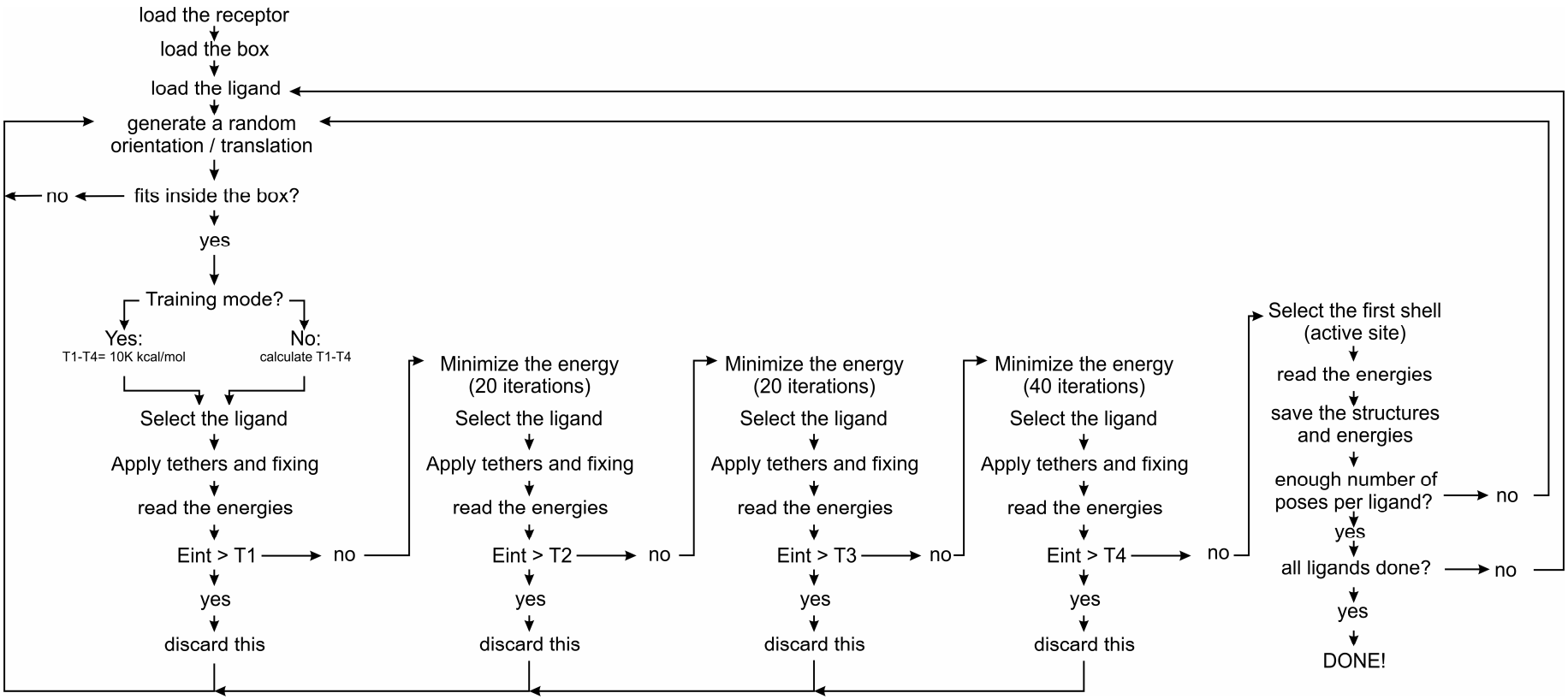
We also prepared a comprehensive set of programs also allows automated comparison and analysis of such forces and energies for an entire set of complexes generated via docking. Please email the author to obtain this particular package.

PART C

Appendix I: Preparation flowchart



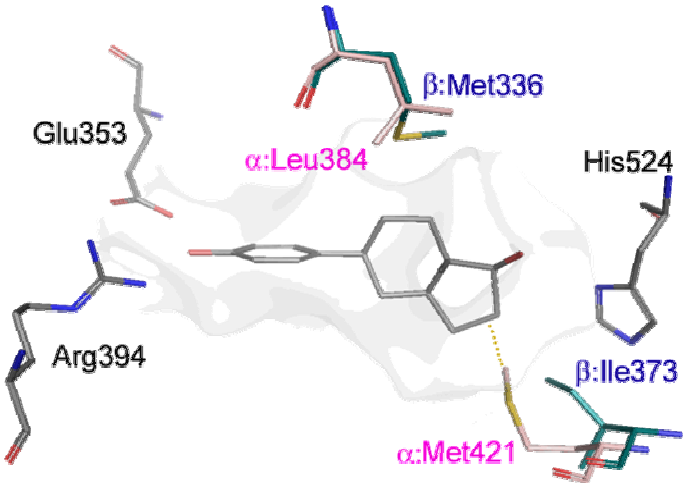
Appendix II: Docking flowchart



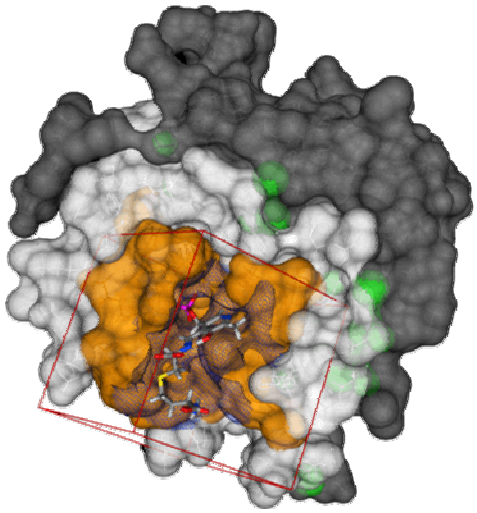
PART D

Selected uDock citations

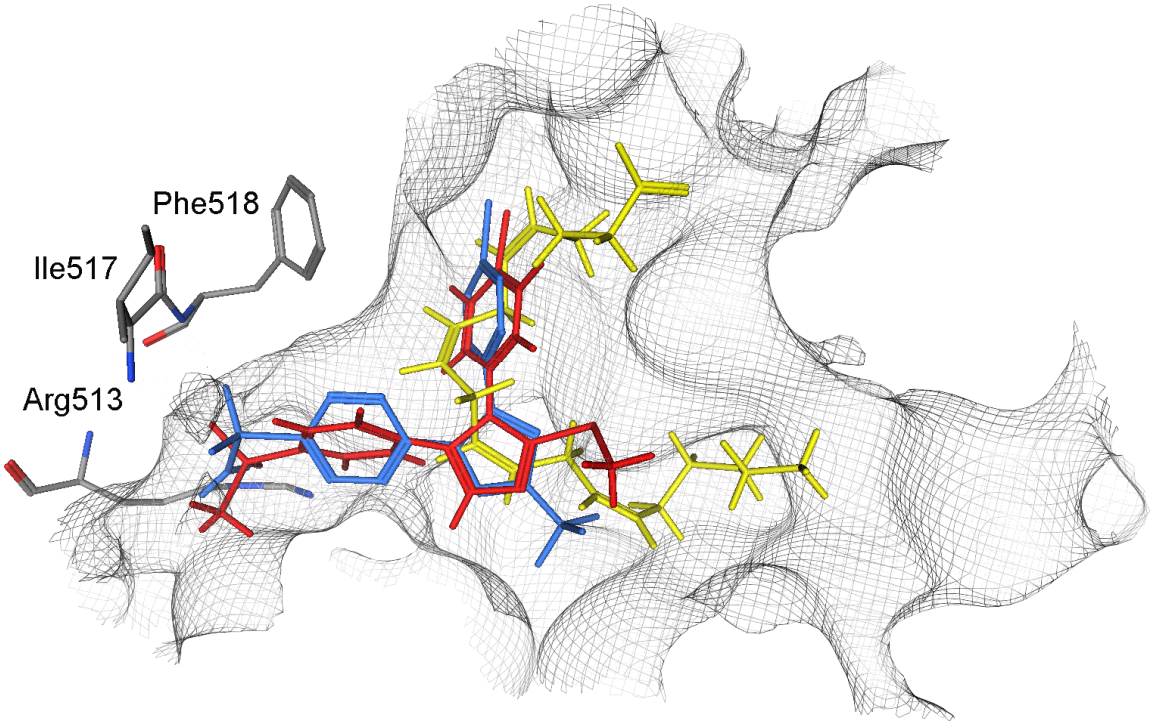
Deconstructing the estradiol ABCD ring structure: A new family of A-CD compounds which are potent and selective estrogen receptor agonists [Link]
Tony Durst*, Mohammed Asim, James S. Wright, Hooman Shadnia, Christine Pratt, John Katzenellenbogen, Kathryn Carlsen. *Bioorganic and Medicinal Biology*, **2009**



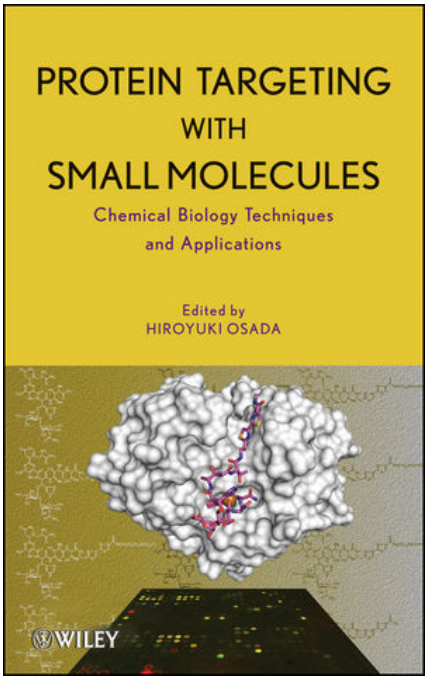
Investigation of Residues K112, E136, H138, G247, Y248, and D249 in the Active Site of YeastCystathionine β-Synthase [Link]
Pratik Lodha, Hooman Shadnia, Colleen M. Helferty, James Wright, and Susan M. Aitken*. *Biochemistry and Cell Biology,* **2009**



Design, synthesis, and biological evaluation of substituted 2-alkylthio-1,5-diarylimidazoles as selectiveCOX-2 inhibitors [Link]
Latifeh Navidpour*, Hooman Shadnia, Hamed Shafaroodi, Mohsen Amini, Ahmad Reza Dehpourd and Abbas Shafiee.*Bioorganic & Medicinal Chemistry*, *15*, 1976–1982,**2007**



Protein Targeting with Small Molecules: Chemical Biology Techniques and Applications
Hiroyuki Osada , Wiley, ISBN: 978-0-470-12053-8, Hardcover



PART E

Tutorials
[To be added soon!]

Tutorial A: how to render the active site cavity
Tutorial B: how to generate a database of ligand conformers
Tutorial C: Docking of estrogenic ligands into ERa

PART F

Datasets
[To be added soon!]

PART G

FAQ
[To be added soon!]