AutoDock4Zn - Tutorial

This document explains how to perform docking simulations with the AutoDock4Zn forcefield described in the paper:

> *AutoDock4$_{Zn}$: An Improved AutoDock Force Field for Small-Molecule Docking to Zinc Metalloproteins*
> Santos-Martins, D., Forli, S., João Ramos, M., Olson, A., J. J.Chem.Info.Mod. **2014**.
> http://dx.doi.org/10.1021/ci500209e

In the last section of this document there is a short checklist of steps which may suffice for advanced users.
NOTE: since version 4.2.6, the extra keywords used for the Zn forcefield are supported by the official AutoDock binaries.

Software requirements
The following software must be already installed:

AutoDock version 4.2.6 or later:
http://autodock.scripps.edu/downloads

MGLTools version 1.5.7 or later
http://mgltools.scripps.edu/downloads

Instructions refers to terminal commands, therefore basic knowledge of UNIX or Windows command line usage is advisable.
The variable $MGLROOT refers to your MGLTools installation directory, e.g.:

```
/usr/local/mgltools-1.5.7/                  (Linux/Mac)
"C:\Program Files (x86)\MGLTools-1.5.7\"    (Windows)
```

WINDOWS USERS: Use the “python.exe” command instead of “pythonsh” for all commands. The path separator in Windows is the backslash (“\”), although Unix paths below can be used by using quotes around them (i.e., cd “AutoDock4Zn_toolbox/”).

Required files
For convenience, have the following files in the working directory:

Data files
protein.pdb and ligand.mol2 (AutoDock4Zn_toolbox/Tutorial)
AD4Zn.dat (AutoDock4Zn_toolbox)

Scripts

prepare_gpf4zn.py	(AutoDock4Zn_toolbox/)
prepare_dpf42.py	($MGLROOT/MGLToolsPckgs/AutoDockTools/Utilities24)
zinc_pseudo.py	(AutoDock4Zn_toolbox/)
prepare_receptor4.py	($MGLROOT/MGLToolsPckgs/AutoDockTools/Utilities24)
prepare_ligand4.py	($MGLROOT/MGLToolsPckgs/AutoDockTools/Utilities24)

Binaries

autodock4	(version: 4.2.6 or newer)
autogrid4	(version: 4.2.6 or newer)
pythonsh	($MGLROOT/bin) (Linux/Mac)
python.exe	(“C:\Program Files (x86)\MGLTools-1.5.7\”) (Windows)

Step 1 - Prepare Receptor and Ligand (as in the standard AutoDock protocol)
Add polar hydrogens, gasteiger charges and set atom types:

```
$MGLROOT/bin/pythonsh prepare_receptor4.py -r protein.pdb -o protein.pdbqt
$MGLROOT/bin/pythonsh prepare_ligand4.py -l ligand.mol2 -o ligand.pdbqt -A  hydrogens
```

It is common to fine tune the receptor by modeling missing atoms, or testing different combinations of conformations and protonation states. Such tasks are not discussed in this tutorial.

Step 2 - Add Tetrahedral Zinc Pseudo Atoms (TZ) to the receptor
TZ atoms represent the preferred position for tetrahedral coordination by the ligand.

```
$MGLROOT/bin/pythonsh zinc_pseudo.py -r protein.pdbqt -o protein_tz.pdbqt
```

Step 3 - Use the modified forcefield (AD4Zn.dat)
The AutoDock4$_{Zn}$ forcefield is mostly defined by non bonded pairwise potentials which are written to the grid parameter file (*.gpf) in the form of `npb_r_eps` keywords. The file AD4Zn.dat includes the definition of the TZ atom type for the AutoDock forcefield.The keyword `parameter_file` in the gpf specifies AD4Zn.dat as the forcefield to be used, so AutoGrid requires a local copy of it in the working directory. Alternatively, the keyword `parameter_file` in the .gpf can point to the full or relative path where AD4Zn.dat is located.

Step 4 - Generate the Grid Parameter File (.gpf)
The preparation script will be executed to generate the GPF to configure the grid calculation:

```
$MGLROOT/bin/pythonsh prepare_gpf4zn.py -l ligand.pdbqt -r protein_tz.pdbqt \
-o protein_tz.gpf  -p npts=40,30,50 -p gridcenter=18,134,-1 \
—p parameter_file=AD4Zn.dat
```

The -p flag is used to set the box center (gridcenter) and size (npts) along with the parameter_file specific for this case.

Step 5 - Run AutoGrid4
The code that supports user defined pairwise potentials (`npb_r_eps` keyword) from the .gpf file was restored from an old version of the software and added to the modified AutoGrid4 binary provided in this tutorial. This changes will be included in next release of the standard AutoDock binaries.

```
autogrid4 -p protein_tz.gpf -o protein_tz.glg
```

At this stage, all forcefield information has been encoded in the maps (*.map), and the remaining steps are the same as in the standard AutoDock protocol.

Step 6 - Generate the Docking Parameter File (.dpf)
The preparation script will be executed to generate the DPF to configure the actual docking calculation:

```
$MGLROOT/bin/pythonsh prepare_dpf42.py -l ligand.pdb -r protein_tz.pdbqt \
    -o ligand_protein_tz.dpf
```

Step 7 - Run AutoDock

```
autodock4 -p ligand_protein_tz.dpf -o ligand_protein_tz.dlg
```

Summary checklist
The following steps differ from the typical AutoDock protocol:
Run `zinc_pseudo.py`
Generate receptor PDBQT file with `prepare_gpf4zn.py`
Use of `AD4Zn.dat` in the parameter files
Use of `autogrid4.2.5.x.20131125`