AMMOS_SmallMol

User's Manual

What is AMMOS

AMMOS (**A**utomatic **M**olecular **M**echanics **O**ptimization for *in silico* **S**creening) [1] employs an automatic procedure for energy minimization of protein-ligands complexes (package AMMOS_ProtLig) or of small chemical compounds present in a library (package AMMOS_SmallMol). As such, the software offers valuable solutions to assist structure-based *in silico* screening experiments or ligand-based projects. The package makes use of molecular mechanics concepts and is based on the program AMMP [2-3], available under GNU license (http://www.cs.gsu.edu/~cscrwh/ammp/ammp.html). AMMOS has been developed in the INSERM – University Paris-Diderot, lab. Bioinformatics-MTI (http://www.vls3d.com/). The package AMMOS is written in C and Python and is available for Linux and Mac OS X systems under the GNU General Public license.

Please, cite:

Tania Pencheva, David Lagorce, Ilza Pajeva, Bruno O Villoutreix and Maria A Miteva.
AMMOS: Automated Molecular Mechanics Optimization tool for *in silico* Screening

You may direct questions related to this package to the corresponding author:
Maria Miteva, PhD
Senior Scientist
INSERM Bioinformatics-MTI University Paris-Diderot
5 rue Marie-Andrée Lagroua

75205 Paris Cedex 13
E-mail: maria.miteva@univ-paris-diderot.fr

What is AMMOS_SmallMol

AMMOS_SmallMol performs energy minimization of small molecules in chemical libraries and can be applied on a huge number of 3D conformations pre-generated with different programs [4-6]. The molecular mechanics minimization of AMMOS_SmallMol is based on two force fields of AMMP: *sp4*, and *sp5* [7]. The entire procedure of AMMOS_SmallMol, from the input of small molecules to the final minimized compounds is shown in the following scheme:

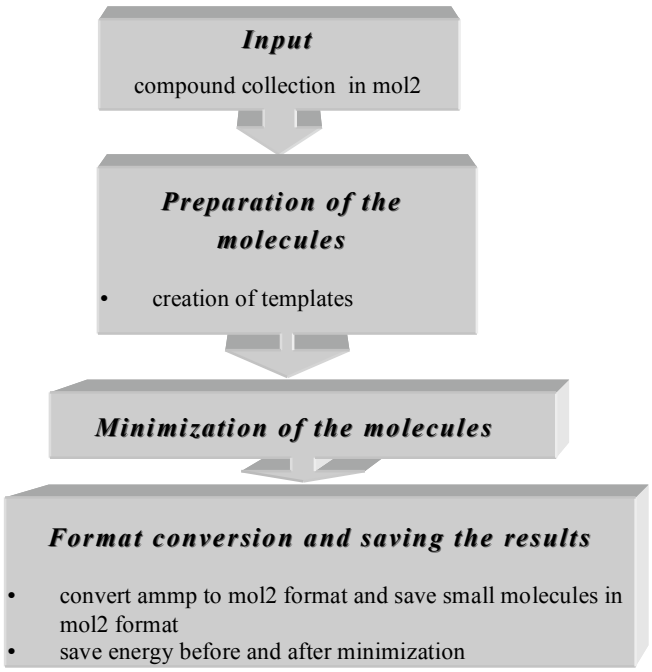

Figure 1. Schematic flow of the AMMOS algorithm;
the arrows show the cycle for the automated procedure

Back to the top

How to install AMMOS_SmallMol

Step 1: Uncompress the package in a Linux shell as follows:

> *tar -zxvf AMMOS_SmallMol.tar.gz*

The directory *~AMMOS_SmallMol* contains the subdirectories:

- "*doc/*": user manual
- "*install*/"

- "*bin/*": executable binary files
- "*progs/*":
  - "*ammp/*": the program AMMP
  - "*preammp/*": the program PREAMMP
  - "*Python_scripts/*": the Python scripts of AMMOS_SmallMol
  - "*vls_min/*": the C programs source and input files for the AMMOS_SmallMol energy minimization protocols
- "*example/*": an example for running AMMOS_SmallMol

Step 2: Compile the source codes
In the *~AMMOS_SmallMol/install* subdirectory run the compiling script "*Makefile*" by typing:

> *> make*

(you need Python 2.4 or above)
The script will compile the source codes automatically and generate executable files for the programs AMMP, PREAMMP, and AMMOS_SmallMol. This script will automatically install all executable files into the directory *~AMMOS_SmallMol/bin/*.

Step 3:
User should edit the ".cshrc" (or ".bashrc") file to add the path of *~AMMOS_SmallMol/bin* in *PATH* environment variable. For example, in c-shell:

> *> set path=($path ~/AMMOS_SmallMol/bin)*

WARNING: this present version should run on both, Linux and Mac versions (eg Mac OS X, 10.4 and 10.5, PPC and Intel). On Mac 10.5 for example, assuming you have the developer tools & shell is bash. You need to generate a .bashrc file and add something like this if you want to install AMMOS only for your account:

PATH=$PATH:"/Users/bruno/Desktop/data.dir/AMMOS_onVLS3D_last_Nov7/ AMMOS_SmallMol /bin:"

export PATH

but then when you start a new terminal or login again…this file .bashrc is eventually not read by the system. So you also have to create a: .bash_profile file, and there you can add:

source ~/.bashrc

Back to the top

How to run AMMOS_SmallMol

In the working directory where AMMOS_SmallMol computations will be ran, the following files should be present: the compound collection in mol2 format and the *input parameter file* (see for example the input.param file in the *~AMMOS_SmallMol/example* directory). After the minimization, the results will be saved in mol2 format. The user should edit this file to give the correct paths and chosen parameters:

The input parameter file:

path_of_*AMMOS_SmallMol = path of the package AMMOS_SmallMol*[1]
bank= *name of the small molecules databank in mol2*

Note:
[1]The total number of symbols of the path_of_*AMMOS_SmallMol* should not exceed 80.

To run AMMOS_SmallMol for energy minimization of small molecules:

> *AMMOS_SmallMol_sp4.py input_parameter_file*

You can apply *sp4* or *sp5* force field.

AMMOS_SmallMol Procedure:
The entire automatic procedure for energy minimization of small molecules with AMMOS_SmallMol, from the input file (the small compound collection) to the output (the final databank of the minimized compounds) is shown in figure 1. The automatic procedure for many small molecules is accomplished via a Python script. The procedure *mol2_to_templ_sp4 (or sp5) (available in ~AMMOS_SmallMol/progs/vls_min)* creates templates file for the small molecules, based on the initial mol2 file. Some warning could appear if unknown atom types or bonds are present.

The input files required by AMMP for the minimization procedures are *min_ligand.ammp* (available in *~AMMOS_SmallMol/progs/vls_min*). They allow for the selection of optimization method (in the presented version *Conjugate gradient*), and the number of iteration steps (in this version 2x500). Experienced users can select other optimization method available in AMMP, as well as number of iteration steps.

After the minimization stage, AMMOS_SmallMol ensures:
1) keeping the new coordinates of the molecules after minimization in mol2 format.
2) keeping the energy of the molecules before and after minimization.
3) keeping any warning that may appear during the AMMOS_SmallMol run.

Back to the top

Example

An example of energy minimization of small molecules with AMMOS_SmallMol showing the procedure is available in the directory *~AMMOS_SmallMol/example*. The initial 3D molecule conformations in mol2 format and the input parameter file (input.param) (see explanations above) are given. Users can simply run:

> *AMMOS_SmallMol_sp4.py input.param*

At the end of the minimization, final results can be found in the directory *~AMMOS_SmallMol/example*:

- *mol_databank_minimized.mol2* contains the coordinates of all minimized molecules (in multiple conformers if available) (see explanations above)
- *mol_databank_energy.txt* contains the energy of molecules before and after the minimization (see explanations above)
- *mol_databank_total_warnings.txt* contains any warning that can appear during the minimization run (see explanations above)

Back to the top

References

1. Pencheva T., D. Lagorce, I. Pajeva, B.O. Villoutreix, M.A. Miteva. AMMOS: Automated Molecular Mechanics Optimization tool for *in silico* Screening.
2. Harrison R., C. Reed, I. Weber. Analysis of Comparative Modeling Predictions for CASP2 Targets 1, 3, 9, and 17. *Proteins: Structure, Function, and Genetics*, 1997, Suppl. 1, 68-73
3. Weber I., R. Harrison. Molecular Mechanics Calculations on Protein–Ligand Complexes. *Perspectives in Drug Discovery and Design*, 1998, 9/10/11**,** 115-127
4. Leite T.B., D. Gomes, M.A. Miteva, J. Chomilier, B.O. Villoutreix, P. Tufféry. Frog: a FRee Online druG 3D conformation generator. *Nucleic Acids Research*. 2006, 34, W738-744.
5. Kirchmair J., G. Wolber, C. Laggner, T. Langer. Comparative performance assessment of the conformational model generators omega and catalyst: a large-scale survey on the retrieval of protein-bound ligand conformations. *J Chem Inf Model,* 2006, 46,1848-1861
6. Sauton N., D. Lagorce, B.O. Villoutreix, M.A. Miteva. MS-DOCK: Accurate multiple conformation generator and rigid docking protocol for multi-step virtual ligand screening. *BMC Bioinformatics*, 2008, 9:184
7. Bagossi P., G. Zahuczky, J. Tözsér, I. Weber, R. Harrison. Improved Parameters for Generating Partial Charges: Correlation with Observed Dipole Moments, *Journal of Molecular Modeling*, 1999, 5, 143-152

Back to the top