VLS_AMMP

User's Manual

What is VLS_AMMP?

The VLS_AMMP package is developed by Dr. Tania Pencheva, David Lagorce and Dr. Maria Miteva in Dr. Bruno Villoutreix's group at the *Laboratoire de Pharmacochimie Moléculaire et Cellulaire*, *INSERM U648*, *Paris*.

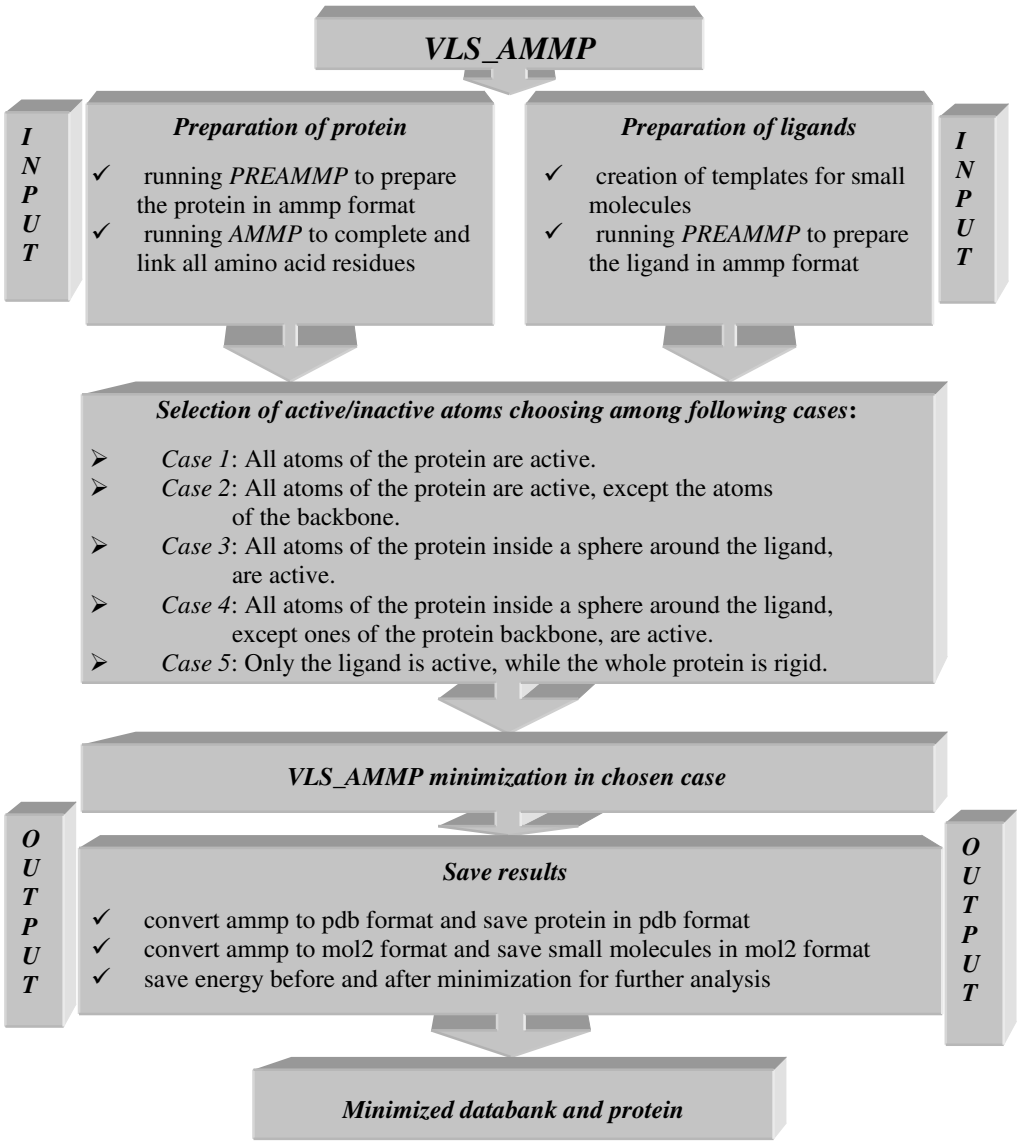
You may direct questions related to this package to the corresponding author(s) at:
***Maria Miteva**, PhD*
Laboratoire de Pharmacochimie Moléculaire et Cellulaire
INSERM U648, 45 Rue des Saints Peres, 75270 PARIS
Phone: 0033 0142864080
E-mail: maria.miteva@univ-paris5.fr

The correct way to refer this package is VLS_AMMP.

VLS_AMMP performs a whole automatic procedure for minimization of protein-ligand interactions for a huge amount of small ligands predocked to a binding site. The package is

developed based on the modern full-featured molecular mechanics program AMMP [1-5]. Since VLS_AMMP does not perform molecular docking by itself, it is typically applied in combination with a molecular docking program, such as DOCK, Flex etc., in structure-base drug design studies. The molecular docking program is used to provide the binding models of the molecules of interests to their target protein. Then, VLS_AMMP can be applied to minimize them and to ensure relatively high enrichment of the proceeded database of small molecules. In fact, VLS_AMMP package could be used separately or as a final step of a hierarchical VLS protocol.

The entire procedure of the minimization of protein-ligand interactions using AMMP, from input of protein and ligands databank to the final minimized databank, is shown in the following figure:



The package VLS_AMMP is developed in two forms: for a minimization of protein-ligands interactions, as well as for a minimization of ligands independently of any protein target. Both packages are supplied in different directories, respectively:

..*VLS_AMMP_MinProteinLigands*\ for a minimization of protein-ligands interactions

and

..*VLS_AMMP_MinMolecules*\ for a minimization of ligands independently of any protein target.

The package offers a possibility the user to choose between two force fields of AMMP: the available in the current version of AMMP *sp4*, and the developed based on Bagossi [6] *sp5*. The choice should be made when the package is going to be started (see How to use VLS_AMMP).

The following reference is supposed to be cited in any resulting publication by applying VLS_AMMP:
Pencheva T., D. Lagorce, I. Pajeva, V. Villoutreix, M. Miteva, *VLS_AMMP*: A New Molecular Mechanics Calculations Tool for Virtual Screening, in preparation.

Back to the top

Where to get VLS_AMMP?

VLS_AMMP is freely distributed to the academic users. It could be received after a request to the corresponding author.

Back to the top

How to install VLS_AMMP?

The VLS_AMMP package is written in *C* and *Python* languages and has been tested on UNIX and LINUX platforms. After downloading the program package, please move it to the directory where you would like the program to be installed. Then, install the program through the following three-step procedure.

Step 1: Uncompress the package
You can do this in a Unix / Linux shell as follows:
gunzip VLS_AMMP_MinProteinLigands.tar.gz (or respectively VLS_AMMP_MinMolecules.tar.gz)
tar -xvf VLS_AMMP_MinProteinLigands.tar (or respectively VLS_AMMP_MinMolecules.tar)

Under your working directory you will get two directories, respectively named:
..*VLS_AMMP_MinProteinLigands*\ for a minimization of protein-ligands interactions
and
..*VLS_AMMP_MinMolecules*\ for a minimization of small molecules independently of any protein target.

Under each of both directories, there are several subdirectories:

- "bin/": executable binary code

- "doc/": user manual in htm and in pdf format
- "example/": example for practicing VLS_AMMP
- "install/": Makefile for the package VLS_AMMP - optimization of Makefile
- "progs/": contains the following programs in the corresponding subdirectories:
  - "ammp/": program AMMP
  - "preammp/": program PREAMMP
  - "Python/": Python scripts
  - "vls_min/": C code and executable ammp files

Step 2: Compile the source codes
Go into the "install" subdirectory and run the compiling script "Makefile" by typing:
make

The script will compile the source codes automatically and generate executable files for programs AMMP, PREAMMP, as well as for the minimization package VLS_AMMP. This script will also automatically copy all executable files to the directory "bin/" in order to be used thereafter.

Please notice that the default C compiler assigned in the "Makefile" script is the SGI MIPS "CC" compiler. If you are using a different C compiler on your computer, e.g. the GNU compiler "g++", you are supposed to change the line in the "Makefile" script started with "CC = ". You may also want to enable certain flags of the compiler to make the executable code work more efficiently on your computer. What kinds of flags should be used depends on the type of your compiler and also your computer. Please consult with an expert if you are not sure about what to do.

Step 3:
User have to edit his ".bash_profile" file and to add the path of VLS_AMMP_MinProteinLigands/bin (respectively VLS_AMMP_MinMolecules/bin, or both) in PATH environment variable. After adding, save your ".bash_profile" and type source ~\.bash_profile.

Back to the top

How to use VLS_AMMP?

The basic function of VLS_AMMP is to minimize the interaction of a given small molecule as a part of huge ligands database to a target protein. It is users' choice where to collect the data from calculation. At the beginning in the chosen directory user should supply the protein, the database of ligands, as well as the *input_parameter_file*. Namely this *input_parameter_file* should consists all of the parameters needed to run VLS_AMMP (see example below). You are supposed to edit this file to meet your own purpose.

Example of input parameter file:
path_of_VLS_AMMP= *absolute path of VLS_AMMP after uncompress, where is all the staff of the package*
protein= *user's name of the desired protein target*
bank= *user's name of the desired ligands database*

case_choice= *user's choice of the preferred case of protein flexibility*
radius= *user's choice for the sphere radius*

The first parameter in the input parameter file is *absolute path of VLS_AMMP*. **Note: If the total amount of symbols in the created from package *VLS_AMMP* files preammp_ligang.txt or preammp_protein.txt (in the directory VLS_AMMP_*/progs/vls_min) is more than 80 symbols, the program PREAMMP crashes.** The next both parameters specifies the input files correspondingly for the protein and for the ligands database. In the next row user has to make a choice for a desired case of protein flexibility. At the last row, the size (in Å) of the sphere around the ligand should be specified. This parameter will be used only in *case 3* and *case 4*, where the partial flexibility of the protein in the sphere with specified size is ensured. You can find an example input file in the "example/" directory.

To run VLS_AMMP, simply use the input file as follows:

- for a minimization of protein-ligand interactions:
  MinProteinLigands_sp*.py *input_parameter_file*

- for a minimization of small molecules:
  MinMolecules_sp*.py *input_parameter_file*

The * denotes your choice for the applied force field.

VLS_AMMP requires as input a protein in pdb format and small molecules in mol2 format. After minimization, the results for the minimized compounds are saved in the same formats.

VLS_AMMP starts with the preparation of the protein and small molecules in the specific ammp format. This step is performed by using program *PREAMMP*. The corresponding templates for proteins are available in the program *PREAMMP* itself, namely in subdirectory *pdb*. During the preparation of the protein some problems could appear, as the most common are availability of incomplete residues and non-correspondence between the atoms in the proteins and templates in *PREAMMP* program. The atoms in the proteins should be with the same names as in templates. All non-correspondences between proteins atoms and templates should be overcome by the user before starting the whole automatic procedure. **Each user should take care to prepare himself the terminal residue**, which is very simple procedure when compare, i.e. LEN with LEU in templates (respectively MEN with MET in templates and ETR with GLU in templates). **Important**: The new prepared residues should be added in the place where the rest of residues are, namely subdirectory *pdb* of the program *PREAMMP*. The program *PREAMMP* should run silently without warnings or error messages [http://www.cs.gsu.edu/~cscrwh/ammphelp/Index.htm]. You may receive a message about "*something.OXT*" not being in the geometry. This means the c-terminal residue has both atoms of the terminal acid. Proposed in the *VLS_AMMP* package script *prepare_protein.ammp* ensures building of all incomplete residues in the protein targets.

In order to prepare small molecules in the specific ammp format, a specific procedure, called *mol2_to_pdb_sp** has to be performed. It ensures the creation of a pdb and a template file for small ligands, based on the initial mol2 file. The pdb and the template files are needed to run the program *PREAMMP*, which creates the input files in ammp format. The *mol2_to_pdb_sp** has been created separately for the force fields *sp4* and *sp5* ("*" denotes chosen force field 4

or 5). This procedure could finish with some warning, possible to appear during the creation of templates and giving information i.e. for a new atom type or wrong atom bonds.

When the protein and the small molecule are ready for a minimization, before to proceed the minimization itself user has to make a choice of a degree of the protein flexibility. The preparation of active/inactive atoms for the protein is ensured by the procedure *active_case**. Five different cases have been elaborated for choice of active/inactive atoms in the protein, while ligands are always active. The "*" corresponds to the case number, defined by the user, who can choose one of the following possibilities:

- *Case 1*: All atoms of the protein are active.
- *Case 2*: All atoms of the protein are active, except the atoms of the backbone.
- *Case 3*: All atoms of the protein inside a sphere around the ligand are active.
- *Case 4*: All atoms of the protein inside a sphere around the ligand, except ones of the protein backbone, are active.
- *Case 5*: Only the ligand is active, while the whole protein is rigid.

There is no *active_case1*, because in this case all atoms are active and it is managed directly at the minimization step.

After that the real minimization can start. The input file for AMMP, that will execute the minimization, is *min_case*.ammp*. The "*" again corresponds to the case number. These scripts contain the selection of optimization method (in the presented version *Conjugate gradient*), and the number of iteration steps (in this version 2x500). Each experienced user can select by itself own optimization method, as well as number of iteration steps, making changes in the corresponding *min_case*.ammp*.

After finishing the minimization, the following possibilities are ensured:

1) keeping of new coordinates of ligands after minimization in the initial mol2 format. This operation is performed by *ammp_to_mol2*, which ensures the opposite conversion from ammp format back to the initial mol2 format.
2) keeping of new coordinates of the protein after minimization in the initial pdb format. This operation is performed by *ammp_to_pdb_case**, which ensures the opposite conversion from ammp format back to the initial pdb format. The "*" corresponds to the case number. There is no *ammp_to_pdb_case5*, because in *case 5* the protein is rigid and there is no change in the protein initial coordinates.
3) keeping the external, internal and total energy between the protein and each ligand before and after minimization. This operation is performed by *save_energy*.
4) keeping any warnings appeared during the minimization.

The automatization of the procedure for many small molecules is accomplished with a *python* script.

Back to the top

Examples

Two examples are supplied at users' disposal in order the working process of VLS_AMMP to be presented:

1. for a minimization of protein-ligand interactions
2. for a minimization of small molecules

In the case of a minimization of protein-ligand interactions, the user has to supply the desired protein target in pdb format, the desired ligands database in mol2 format, and the input parameter file, edited by the user in order to meet his purpose (see common explanation above, also see the concrete ammp.param in the VLS_AMMP_MinProteinLigands\example\ directory). In this case user has simply to run
MinProteinLigands_sp*.py ammp.param

At the end of the minimization, user will have four important files with saved results after the minimization:

- *ligand_databank_case3_minimized.mol2* - contains the coordinates of the small molecules after the minimization (see explanation above)
- *ligand_databank_case3_output_pdb.pdb* - contains the coordinates of the protein after the minimization (see explanation above)
- *ligand_databank_case3_energy.txt* - contains the external, internal and total energy between the protein and each ligand before and after the minimization (see explanation above)
- *ligand_databank_case3_total_warnings.txt* - contains any warnings appeared during the minimization (see explanation above)

These four important files are saved in the subdirectory OUTPUT in the working directory, where the user supply the protein, databank of small molecules and input_parameter_file.

In the case of a minimization of small molecules independently of any protein target, the user has to supply only the desired ligands database in mol2 format, and the input parameter file, edited by himself in order to meet his purpose (see common explanation above, also see the concrete ammp.param in the VLS_AMMP_MinMolecules\example\ directory). In this case user has simply to run
MinMolecules_sp*.py ammp.param

At the end of the minimization, user will have three important files with saved results after the minimization, namely:

- *small_molecules_databank_minimized.mol2*,
- *small_molecules_databank_energy.txt*
- *small_molecules_databank_total_warnings.txt*

contained the same information as described in the case of a minimization of protein-ligand interactions.

These three important files are saved in the subdirectory OUTPUT in the working directory, where the user supply the protein, databank of small molecules and input_parameter_file.

Back to the top

Trouble shooting

The VLS_AMMP package is developed as a user-friendly program. If you have experienced any problem when using this package, you may contact the author at the address mentioned above. We will try our best to get back to you. *However, before you contact us, please make sure you have gone through this manual and it does not have the answer to your question. For*

general questions about how to run a program on Unix/Linux platform, please consult with a computer expert instead of us.

We are in the process of compiling more complete FAQs for VLS_AMMP. Any suggestion or comment on the program is highly appreciated. We have benefited so much by communicating with the VLS_AMMP users.

Back to the top

References

1. 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.
2. Weber I., R. Harrison, Molecular Mechanics Analysis of Drug-resistant mutants of HIV Protease, *Protein Engineering*, 1999, 12(6), 469-474.
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. Weber I.; R. Harrison, Molecular Mechanics Calculations on HIV-1 Protease with Peptide Substrates Correlate with Experimental Data, *Protein Engineering*, 1996, 8, 679-690.
5. Weber I., R.Harrison, Molecular Mechanics Calculations on Rous sarcoma Virus Protease with Peptide Substrates, *Protein Science*, 1997, 6, 2365-2374.
6. 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