MOLS v2.0.3

Software package for Protein-Ligand Docking and Peptide Modelling

---

User Manual

Authors:

D. Sam Paul, Ph.D.

S. Nehru Viji, Ph.D.

P. Arun Prasad, Ph.D.

K. Vengadesan, Ph.D.

N. Gautham, Ph.D., FNASc

Centre of Advanced Study in Crystallography and Biophysics

University of Madras, India

CONTENTS

1. INTRODUCTION

MOLS 2.0.3 is a Java-based Graphical User Interface for iMOLSDOCK [1–3], a 'flexible ligand - rigid protein' docking algorithm. In addition, MOLS 2.0.3 has MOLS, a conformation search tool [4] and the recently upgraded version of MOLSDOCK with 'induced fit' receptor flexibility for docking peptide ligands. MOLS 2.0.3 uses *Jmol: an open source Java viewer for chemical structures in 3D. (http:www.jmol.org)*.

We welcome your suggestions to enable us to improve MOLS 2.0.3 Any difficulty in installing MOLS 2.0.3 or bug reports please contact:

Dr. N. Gautham
Professor (retd.),
Centre of Advanced Study in Crystallography and Biophysics,
University of Madras, Guindy, Chennai – 600025.
email id: *n_gautham@hotmail.com*

Acknowledgements

The development of MOLS 2.0.3 is the result of a team effort and in particular contributions from **Vengadesan**, **Arun Prasad** and **Nehru Viji** are acknowledged.

If you use MOLS 2.0.3 for publication, please cite:

Paul DS, Gautham N (2016) MOLS 2.0: software package for peptide modeling and protein–ligand docking. J Mol Model 22:1–9. doi: 10.1007/s00894-016-3106-x

Publications related to MOLS 2.0.3

Sam Paul, D., Gautham, N. (2018). Protein–small molecule docking with receptor flexibility in iMOLSDOCK. J Comput Aided Mol Des. doi: 10.1007/s10822-018-0152-8

Paul, D.S., and Gautham, N. (2017). iMOLSDOCK: Induced-fit docking using mutually orthogonal Latin squares (MOLS). J. Mol. Graph. Model. *74*, 89–99.

Viji, S.N., Balaji, N., and Gautham, N. (2012). Molecular docking studies of protein-nucleotide complexes using MOLSDOCK (mutually orthogonal Latin squares DOCK). J. Mol. Model. 1–18.

Viji, S.N., Prasad, P.A., and Gautham, N. (2009). Protein- Ligand Docking Using Mutually Orthogonal Latin Squares (MOLSDOCK). J. Chem. Inf. Model. *49*, 2687–2694.

Arun Prasad, P., and Gautham, N. (2008). A new peptide docking strategy using a mean field technique with mutually orthogonal Latin square sampling. J. Comput. Aided Mol. Des. *22*, 815–829.

Vengadesan, K., and Gautham, N. (2003). Enhanced sampling of the molecular potential energy surface using mutually orthogonal Latin squares: Application to peptide structures. Biophys. J. *84*, 2897.

2. INSTALLATION

MOLS 2.0.3 is available only for Linux (64 bit).

2.1 Prerequisites:

1. Java 7 or its later versions
2. C++ compiler

Fpocket 2.0 [6] and Open Babel 2.4.1 [7] are shipped along with MOLS 2.0.3. Fpocket 2.0 and Open Babel are installed along with the installation of MOLS 2.0.3.

2.2 How to install MOLS 2.0.3

To install MOLS 2.0.3, download MOLS2.0.3.tar.gz from https://sourceforge.net/projects/mols2-0/. To complete the installation, **type** the following series of commands in a command line.

> Note :
>
> 1. The below installation commands and environment settings are written assuming MOLS2.0.3 is installed in the home directory(/home/user-name/).

Installation steps in RedHat/CentOS/Fedora/Ubuntu/mint

Step 1: $ tar -zxvf MOLS2.0.3.tar.gz

Step 2: $ cd MOLS2.0/

Step 3: $ sh pre-install.sh

Step 4: Set environment for 'zlib' and 'cmake'

After step 3, type the following lines in ~/.bashrc

```
export PATH=~/MOLS2.0/cmake-3.5.1/bin:$PATH
export LD_LIBRARY_PATH=~/MOLS2.0/mols_zlib/lib:$LD_LIBRARY_PATH
```

Step 5: $ source ~/.bashrc

Step 6: $ sh install.sh

Step 7: Set environment for 'fpocket2.0', 'openbabel' and 'intel-libraries'

Type the following lines in ~/.bashrc

```
export PATH=~/MOLS2.0/fpocket2/bin:$PATH
export PATH=~/MOLS2.0/open-babel-tools/bin:$PATH
export LD_LIBRARY_PATH=~/MOLS2.0/open-babel-tools/lib:$LD_LIBRARY_PATH
export LD_LIBRARY_PATH=~/MOLS2.0/intel64_lin:$LD_LIBRARY_PATH
```

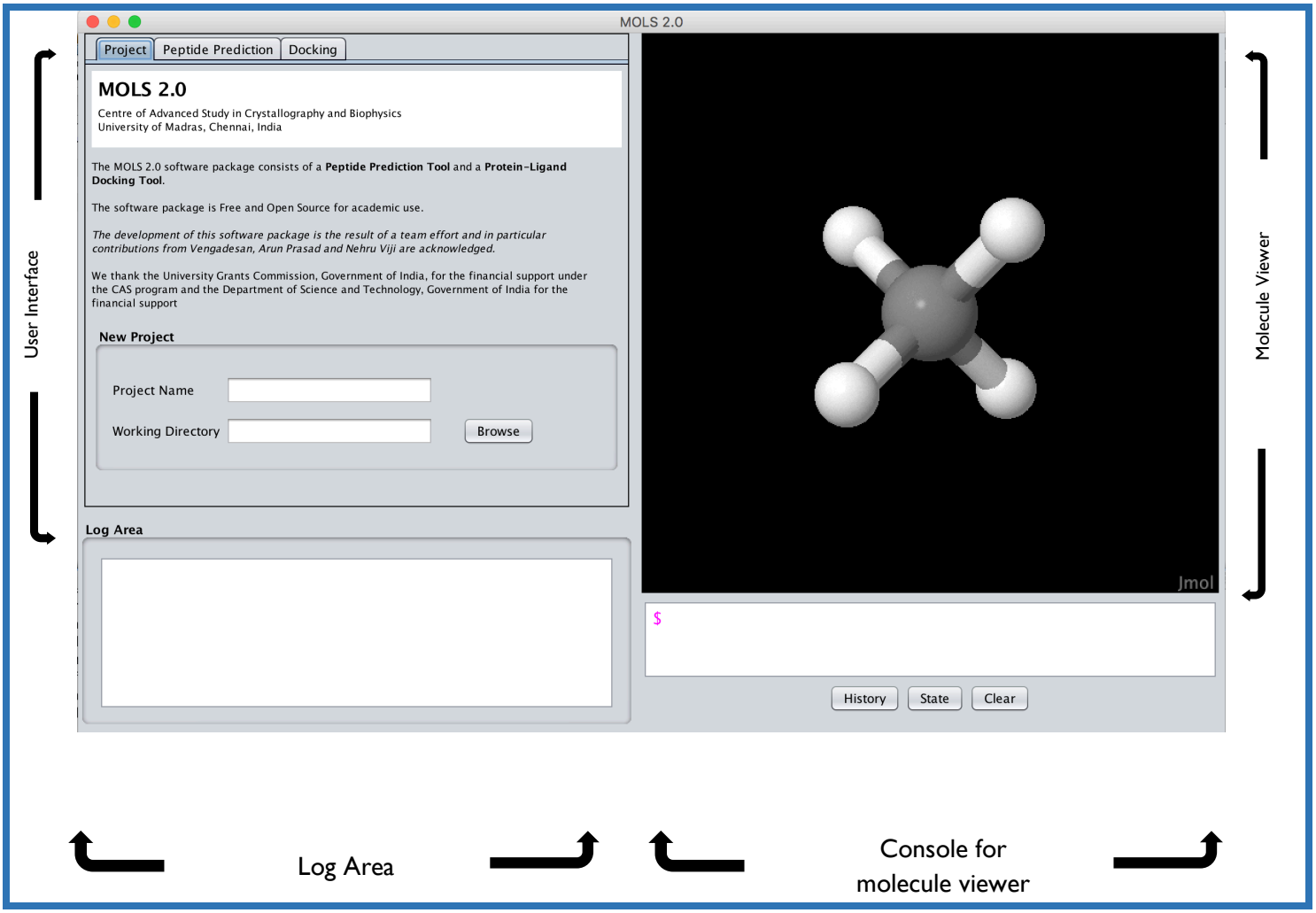
Step 8:sh post-install.sh

Step 9: $ source ~/.bashrc

3. To launch the MOLS2.0.3 user interface, type the following commands

```
$ cd ~/MOLS2.0

$ java -jar mols.jar
```



Tutorial could be downloaded from https://sourceforge.net/projects/mols2-0/files/. More details are given under section 5 in this user manual.

4. To run iMOLSDOCK using command line, details required for docking have to written in the following files before executing *imolsdock*

1. **user.inp**

Here is a sample *user.inp*. Information about each line is given below.

```
 1 0
 2 /home/sam/MOLS2.0/result
 3 case_1w1p
 4 2
 5 /home/sam/dataset/1w1p/ligand_1w1p.mol
 6 2
 7 43.079 75.602 51.839
 8 /home/sam/bitbucket/dataset/1w1p/protein_1w1p.pdb
 9 2 30.0
10 0
11 0 4.0
12 1.0 5.0 1.0
13 0
14 1000
```

Line No.	Input	
1.	0 - Disable comment ; 1 - Enable comment	
2.	Path where results have to be saved	
3.	Project Name	
4.	1 - Peptide-Protein docking ; 2 - Small molecule-Protein docking	
5.	Peptide sequence / Small molecule structure (.mol)	
6.	Force field to calculate intraligand energy Peptide-protein docking: 1 – AMBER ; 2 – ECEPP Small molecule-protein docking; 1 – MMFF94 ; 2 - GAFF	
7.	Grid Centre (x ,y , z of active site)	
8.	Receptor protein structure	
9.	1 – Rigid receptor docking 2 – Flexible receptor docking	0° - 360° (Range of flexibility in flexible residues of receptor protein)
10.	0 – side-chain flexibility in receptor protein.	

11.	0 – Manual Flexible residues 1 – Auto Find Flexible residues	4.0 Å – Scan radius for finding flexible residues around the active site.	
12.	Optimal weight for Scoring Function		
	1.0 – intraligand energy	5.0 – Protein-ligand interaction energy	1.0 – intraprotein energy
13.	0 - Default		
14.	1000 (default) – Number of iterations in Minimization		

2. **input** file (number of structures to be generated)

Format: <starting structure number> <ending structure number>

For example, to generate structures from 1 to 10

```
1 10
```

1 (starting structure number) 10 (ending structure number)

3. **residues.txt** (List of flexible residues in the active of the receptor protein)

Format: <Residue Name> <Chain ID> <Residue Number>

```
LEU A 2
ASN A 6
LEU A 19
```

 # After feeding the inputs through the above files, type the following to execute protein-ligand docking

```
$ ./imolsdock
```

5. iMOLSDOCK: The Docking Tool

5.1 Inputs for Protein-Ligand Docking

Project Name:	A name for the current docking simulation has to be mentioned in the *project name* text box. This project name will be prefixed to the output result files.
Working Directory:	Path of the directory where the output result files have to be stored has to be selected.
Ligand:	The ligand may be any organic chemical compound. If the structure of the ligand to be docked is available in MDL Molfile (.mol) format then that may be selected. If the structure of the ligand is not available then the desired ligand may be built using the built-in Molecule Builder of MOLS2.0.3.
Receptor:	The receptor, usually a protein, in .pdb format has to be selected.
Grid centre:	Grid centre is the centroid of the ligand binding site in the receptor protein.
manual	This option can be used if the ligand binding site in the receptor protein is known. The binding site can be defined by the coordinates of the centre of the box and its dimensions after clicking the *manual* button.
auto	This option can be used if the ligand binding site in the receptor protein is not known. When the *auto* button is clicked, the binding site in the receptor protein is automatically predicted using the Fpocket algorithm [6]. Coordinates of the binding site centroid will be displayed in the x, y, z text boxes respectively.
Flexible Residues:	If receptor flexibility is enabled then side chain flexibility will be allowed for the selected flexible residues.

Manually specify flexible residues	If the flexible residues are known then click the 'manual' option and specify the flexible residues in the Text Area. The Residue name, Chain ID and the Residue number of the flexible residues have be given in the following format: ASP A 129,LEU A 130,ILE A 131
Auto find flexible residues	When the flexible residues are not known, the auto option may be used. Before selecting this option, the protein structure and the binding site must be specified. The 'Auto find flexible residues' option will scan for protein residues(neighbouring residues) that are within the specified cut-off distance from each atom of the ligand. The neighbouring residues will be show in the Text Area. The flexible residues have to be *saved* by clicking 'Save Flexible Residues' Flexible residues may be added or removed in the Text Area. **Note**: If any flexible residue is added or removed in the Text Area then before proceeding to the next step flexible residues have to be saved by clicking 'Save Flexible Residues'
No. of structures:	The required number of optimal conformations has to be mentioned. The starting conformation number has to be mentioned in *start* textbox and the ending conformation number has to be mentioned in the *end* textbox. [Note that for technical reasons, the *start* and the *end* are cardinal numbers.] The '*no. of structures'* cannot exceed 1500.

5.2 Inputs for Protein – Peptide Docking

Project Name:	A name for the current docking simulation has to be given in the *project name* text box. This project name will be prefixed to the output result files.
Working Directory:	Path of the directory where the output result files have to be stored has to be selected.
Peptide:	The amino acid sequence of the peptide ligand has to be given by single letter code or in capital letter.
Receptor:	The receptor, usually a protein, in .pdb format has to be selected.
Grid centre:	Grid centre is the centroid of the ligand binding site in the receptor protein.
manual	This option can be used if the ligand binding site in the receptor protein is known. The binding site can be defined by the coordinates of the centre of the box and its dimensions after clicking the *manual* button.
auto	This option can be used if the ligand binding site in the receptor protein is not known. When the *auto* button is clicked, the binding site in the receptor protein is automatically predicted using the Fpocket algorithm [6]. Coordinates of the binding site centroid will be displayed in the x,y,z text boxes respectively.
Flexible Residues:	If receptor flexibility is enabled then side chain flexibility will be allowed for the selected flexible residues.
Manually specify flexible residues	If the flexible residues are known then click the 'manual' option and specify the flexible residues in the Text Area. The Residue name,

	Chain ID and the Residue number of the flexible residues have be given in the following format: ASP A 129,LEU A 130,ILE A 131
Auto find flexible residues	When the flexible residues are not known, the auto option may be used. Before selecting this option, the protein structure and the binding site must be specified. The 'Auto find flexible residues' option will scan for protein residues(neighbouring residues) that are within the specified cut-off distance from each atom of the ligand. The neighbouring residues will be show in the Text Area. The flexible residues have to be *saved* by clicking 'Save Flexible Residues' Flexible residues may be added or removed in the Text Area. **Note**: If any flexible residue is added or removed in the Text Area then before proceeding to the next step flexible residues have to be saved by clicking 'Save Flexible Residues'
No. of structures:	The required number of optimal conformations has to be mentioned. The starting conformation number has to be mentioned in *start* textbox and the ending conformation number has to be mentioned in the *end* textbox. [Note that for technical reasons, the *start* and the *end* are cardinal numbers.] The '*no. of structures'* cannot exceed 1500.

5.3 How to run docking?

Click *run* button to start iMOLSDOCK docking simulation. After starting the docking simulation, the 'Docking Status' button may be clicked to check the running status of the project in an external Text Editor.

3.4 Output files

Successful execution and completion of iMOLSDOCK produces the following output files.

mols.pdb	File with co-ordinates of all optimal MOLS peptide conformations in PDB format
mini.pdb	File with co-ordinates of all minimized ligand conformations in PDB format
mols_complex.pdb	File contains all the optimal MOLS ligand-protein complex structures.
mini_complex.pdb	File contains all the conjugate gradient minimized ligand-protein complex structures.
mols.log	Log file with optimal MOLS energy of all the structures.
mini.log	Log file with energy of all minimized structures.

4. MOLS: Peptide Prediction Tool

4.1 Inputs for Peptide Prediction Tool

Molecule	The name of the molecule. This name will be prefixed to the output files.
Set output directory	The directory on to which the results will be stored has to be selected.

Sequence The sequence of peptide by single letter code either small or caps.

No of cycles Number of optimal conformations.

Choose search option

(i) ***Backbone*** This option will take only the backbone for MOLS search and side chains are fixed at amino acid (insight II) library value.

(ii) ***Backbone + Side chain*** This option will take both backbone and side chains for MOLS conformational search.

(iii) Rotamer This option will take backbone + side chain for MOLS search. However side chain conformations are chosen from a rotamer library [8], [instead of the 10° grid used in option (ii)]

Choose run option

(i) ***PDBgen*** *PDBgen* will build an initial model for the given amino acid sequence, with an extended conformation for the backbone, and maximally extended side chain conformations, as in the BIOSYM (Biosym/MSI, 1995) package.

(ii) ***MOLS structures*** This option generates the specified number of low energy structures using the MOLS technique.

(iii) ***Minimised structures*** This option will move each structure from the discrete grid on which it is generated to the nearest local minimum using conjugate gradient minimization (Press, *et al*., 1992).

(iv) ***Clustering*** This option will cluster together similar conformations among those generated and minimized.

Choose force field

(i) AMBER [9]

(ii) ECEPP/3 [10]

Choose clustering algorithm

(i) K-means [11]

(ii) Hierarchical [12]

If *Hierarchical* clustering algorithm is selected then the user has to specify ***RMSD cutoff*** value and select the mode of superimposition (***all atom, Backbone atoms only*** and $\boldsymbol{C_\alpha}$ ***trace atoms only***) during RMSD calculation. Otherwise ***RMSD cutoff*** and mode of superimposition can be ignored.

Interval length Interval length has to be mentioned for drawing 'cluster plot'. This plot gives the relationship between the numbers of new clusters discovered versus the structure number.

After supplying the required inputs to the MOLS Peptide Prediction Tool, click *save* button to save the input.

4.2 Outputs

After successful execution and completion of MOLS conformational search, the following output files will be produced.

inp.pdb	File with co-ordinates of the given peptide sequence in extended conformation (this pdb file can be used to view the initial molecule conformation).
mols.pdb	File with co-ordinates of all optimal conformations in the PDB format.
mini.pdb	File with co-ordinates of all optimal conformations minimized by conjugate gradient minimization method.
cent.pdb	File with cluster properties and co-ordinates of each cluster centroids.
cplt.pdb	File containing plot with structure number versus number of new clusters.
mini.out	File with dihedral angles and energy of all minimized conformations.
mols.out	File with dihedral angles and energy of all MOLS optimal conformations.
inf.log	Log file with status of program completion. This file also has the computation time and system time.

Brief description of MOLS Subroutines used for Peptide Prediction

Subroutine PDBGEN (file: pdbgen.f)

The sequence of peptide is input to this subroutine. If the sequence is in small letter it converts them into capital letters. If there is any unusual amino acid it gives alert to the user and terminates the program. It finds the length of the given amino acid sequence (number of residues). The coordinates of all amino acids have been placed into the library files 'ALLAMINOMOLS_1.lib' and 'ALLAMINOMOLS_2.lib'. The library 'ALLAMINOMOLS_2.lib' has co-ordinates of 'ALLAMINOMOLS_1.lib', but rotated 180°. These two libraries are used to build the extended conformation for the input peptide sequence. The co-ordinates of 'ALLAMINOMOLS_1.lib' and 'ALLAMINOMOLS_2.lib' are taken from the insight II library. Amino and Carboxyl terminals are added automatically at the terminals. After the input peptide sequence is checked and confirmed to be correct, the routine searches the co-ordinate library and identifies the corresponding co-ordinates of each residue in the peptide sequence and joins them sequentially. The co-ordinates of residues in the odd positions are taken from the first library ('ALLAMINOMOLS_1.lib') and the co-ordinates of residues in the even positions are taken from the second library ('ALLAMINOMOLS_2.lib'). Then 'molgen1' routine is used to rotate 'φ' angle (For proline 'φ' is fixed at 109° to maintain the geometry). The co-ordinates of the built molecule will be written in 'mols.pdb'. In this routine (PDBGEN) the total number of atoms as stored as 'natom', length of the input peptide sequence is stored as 'lres' and input single letter code peptide sequence are stored as 'fres(lres)'.

Subroutine AMPPAR (file: amppar.f)

The co-ordinate file 'mols.pdb' generated by PDBGEN routine is the input to the routine. It reads the atoms names, atoms numbers, residue names and residue numbers from the file 'mols.pdb' and uses these information to generate inter atomic non-bonded interaction pairs (1-4 pairs, 1-5 and more pairs for electrostatic energy, van der Waals energy and hydrogen bonding energy terms) by using connectivity library 'ALLCONN.lib'. Atom pairs which are not affected during molecular rotations will be excluded (for example, atom pairs in the rings and about peptide bond). In this routine the parameters for AMBER force field energy calculation are extracted from 'ATOMTYPE.lib' and 'ENERGYPARAM.lib' libraries. The rotatable in the

molecule and their corresponding moving atoms using the library 'DIHEDS.lib' will be identified. All these information will be written in the file 'mols.inp'. The output from subroutine AMPPAR is the number of rotatable bonds (number of conformational parameters in variable 'ntor' and number of hydrogen bond pairs in 'nhb').

Subroutine ECPPAR (file: ecppar.f)

Input to this routine is 'mols.pdb'. As mentioned in subroutine 'AMPPAR', the parameters and other information for ECEPP/3 force field energy calculation are extracted from 'mols.pdb' using 'ALLCONN.lib', 'ATOMTYPE.lib', 'ENERGYPARAM.lib' and 'DIHEDS.lib'. The extracted information is written into 'mols.inp'. Here some constants are taken from the header file 'ecepp.h'.

Subroutine MAIN (file: drive.f)

This is the main routine which controls all the subroutines in the MOLS package. In this routine 'user.inp', the file which stores all the inputs (peptide sequence, search option, run option, force field option, clustering option), is read. According to the inputs other main subroutines like PDBGEN, AMPPAR or ECPPAR, VARINIT, CONFORMATION, MOLS, MINIMIZ, CLUSTER or MCLUST will be called from this MAIN subroutine. This MAIN routine gives total computation time, system time and job status information in the log file.

Subroutine VARINIT (file: varinit.f)

The inputs for this routine are 'mols.pdb' and 'mols.inp'. Routine 'VARINIT' uses function 'dihedr' to calculate the initial values of the conformational parameter and initializes them using library 'VAR.lib'. Routine 'VARINIT' supplies the co-ordinates for the given peptide sequence, parameters needed for energy calculations and rotation parameters in the common variable 'x (maxatm, 8)'.

Subroutine CONFORMATION (file: conformation.f)

This routine generates the co-ordinates of the molecule making use of the conformational parameter (torsion angle) through the common variable 'x (maxatm, 8)'. If key option 'kk' is '0', then the routine generates co-ordinates of initial molecule. If key option 'kk' is '1', then the

routine generates co-ordinates for MOLS structures. If key option 'kk' is '2', then the routine generates co-ordinates for minimized MOLS structures. The co-ordinates are written in 'molsx.pdb', 'molsi.pdb', 'molso.pdb' respectively.

Subroutine MOLS (file: mols.f)

This routine is the core routine of the MOLS package. Small subroutines namely '*anggen*', '*scinp*', '*rand1*', '*write_par*', '*pargen1*', '*subpar*', '*molgen*', '*ampene*' or '*ecpene*', '*average*', '*sort_and_set_rank*', '*rank_sort*', '*best*' and '*output*' are used in routine MOLS. This routine gets the co-ordinates of the initial molecule and the parameters for energy calculation via the common variable 'x (maxatm, 8)'.

Subroutine '***anggen***' generates the values for the conformational variable (torsion angle). By default, the range for torsion angle is chosen to be 0° to 360° with step/grid size 10°. After that '***anggen***' checks whether the total number of variables (torsion angles) is lesser than the order of Latin square. In the current package the order of mutually orthogonal Latin squares (mols) is set to 37. If the number of variables is greater than 37 then the order of mols will be set to the next higher prime number and appropriate grid size will be identified to generate values for the conformational variables. Finally the conformation variable/torsion angle values are stored in the variable 'ang(maxpar,maxord)'.

If 'rotamer' search option is selected then subroutines **'*scinp*'** and '***subpar***' will be used to identify the position of side chains to insert Tuffery side chain rotamers using 'SC.lib' library. The number of conformational variables will be changed accordingly.

'***rand1***' subroutine generates random seed values to supply them to system random number generator for every cycle of each run. The four-digit random seeds are stored in variable 'jseed (5000)'.

'***write_par***' subroutine randomizes the values of conformational variables using system random number generator 'rand'. Initial values are received from variable 'ang (maxpar,maxord)' and outputs are sent to variable 'angle(maxpar,maxord)'.

'***pargen1***' constructs the mutually orthogonal Latin squares using the conformational variable values using 'angle (maxpar, maxord)' as symbols and stores the elements of MOLS in variable 'e (maxord,maxord,maxpar)'.

'***molgen***' subroutine rotates the molecule using small subroutines 'elemen' and 'rotor' using the conformational values in each subsquares of MOLS from variable 'e(maxord,maxord,maxpar)'. The initial co-ordinates are taken from 'x (maxatm, 8)' and the rotated co-ordinates are stored in variable 'y (maxatm, 8)'.

Functions ***ampene*** and ***ecpene*** are used to calculate the AMBER and ECEPP/3 energy of generated conformations. These functions take co-ordinates and parameters from common variable 'y (maxatm, 8)'. Based on the user's choice either AMBER or ECEPP/3 energy function will be used for energy calculations.

Subroutine '***average***' is used to calculate Boltzmann weighted average for each conformational variable.

Subroutine '***sort_and_set_rank***', '***rank_sort***', '***best***' are used to analyse the average values obtained from subroutine '***average***' and the optimum values for each conformational variable will be selected.

Subroutine '***output***' gives the output conformational variable values, energy of the structures through common variable opmo (maxstr,maxpar) and emo(maxstr). A file 'mols.out' will also be generated.

Subroutine MINIMIZ (file: minimiz.f)

Subroutine MINIMIZ minimizes the MOLS conformations by Conjugate Gradient method. This routine gets variable values and energy of MOLS conformations from routine '***mols***' through common variables 'opmo(maxstr,maxpar)' and 'emo(maxstr)'. After minimization this routine puts the minimized values of the variables and energy in 'opmi(maxstr,maxpar)' and 'emi(maxstr)'. The minimized values are recorded in 'minimiz.out' file. User chosen energy (AMBER or ECEPP/3) will be used in this routine.

Subroutine CLUSTER (file: cluster.f)

Subroutine CLUSTER uses co-ordinates of optimized conformations from 'molso.pdb' file and clusters them by K-means algorithm as described in the SCAR clustering algorithm except the last two steps which involve substructure clustering and centroid minimization. Initial global cluster cutoff will be calculated from random segments of non-homologous protein chains with length varying from five to fifteen residue length. The global cluster cutoff will be stored in variable 'pgcut(20)'. The user will be cautioned when the peptide sequence length is greater than 15 residues. This subroutine used kabsch algorithm for superimposition and calculates rmsd between pair of conformations. Subroutine '***clusterplot***' is used to draw the cluster plot. The number of clusters, its members, their properties and number of structures vs number of new clusters plot are given in the following files 'cluster.out', 'centroids.pdb', 'plot.out' respectively.

Subroutine MCLUST (file: mclust.f)

This subroutine gets the co-ordinates of the optimized conformations from 'molso.pdb' file and clusters them using hierarchical algorithm described by Kriz et al. Number of clusters, its members, their properties and number of structures vs number of new clusters plot will be written in 'mcluster.out', 'cprop.out','plot.out' files respectively.

Brief description of iMOLSDOCK Subroutines used for Protein-Ligand Docking

Program MAIN (file: smdrive.f): This is the main program in iMOLSDOCK package. The user's input like project name, ligand structure, receptor protein structure, number of required ligand conformations, centroid of the ligand binding site in the receptor protein will be read from input file 'user.inp'. Shell script '*fileform.sh* 'converts the ligand from .mol file format to Simplified Molecular Input Line Entry System (SMILES) format using OpenBabel 2.4.1. Here the receptor protein is read and the total number of atoms (excluding the HETATMs) in the receptor protein is stored in the common variable '*ipatom*'. Subroutines *pdbgen, varinit, conformation, mols , minimiz* will be called from this program MAIN.

Subroutine PDBGEN (file: smpdbgen.f)

This subroutine has three subroutines '*prep_coord*', '*coorgen*' and '*store_index*'. In Subroutine *'prep_coord'* shell script '*script1'* generates the three-dimensional structure of the ligand from the SMILES string and then the rotatable bonds in the ligand structure are found using 'findrotatable.pl'. In subroutine *'lchange'* the total number of atoms in the ligand is found and stored in the common variable '*natom*'. Shell script '*script01'* calculates the intramolecular ligand energy using Open Babel 2.4.1. Since the ligand is a small organic chemical compound the MMFF94 force field is used. *Subroutine 'coorgen'* extracts the co-ordinates from the ligand file. The co-ordinates of the ligand are then stored in the common variable '*rx (100,3)*'. Half the length of the initial conformation of the ligand is stored in common variable '*big*'. Subroutine *'store_index'* stores the indices of the rotatable bonds and its neighbours.

Subroutine VARINIT (file: smvarinit.f)

Subroutine '*dockinit*' reads the receptor protein file and assign atom type to the atoms in the binding site. Protein atoms at a distance less than the sum of half the length of the initial conformation of the ligand and the maximum interaction range of the PLP function (6 Å) are only included in the search space. For which, the cutoff distance is stored in variable 'cutdist'. Number of protein atoms within '*cutdist*' is stored in variable '*ic*'. Atom type of protein atoms (in the search space) and the ligand atoms are stored in common variable '*pat(mnatp)*' and common variable '*lat(maxatm)*'. The atom type of the protein and the ligand will be later used in the PLP scoring function.

Subroutine MOLS (file: smmols.f)

The conformational space of the ligand is searched using the MOLS algorithm. Therefore this subroutine has all the small subroutines in subroutine *MOLS* of the MOLS tool with few additional small subroutines added for iMOLSDOCK. The added few are '*precal*', '*eplp*', '*rotate*' and '*translate*'. The search space in iMOLSDOCK is M+6, where 'M' is the number of rotatable bonds in the ligand and the six additional parameters to describe the position and orientation of the ligand. In the initialisation part of subroutine '*MOLS*', 'M+6' is stored in variable '*npar*'. The Piecewise Linear Potential (PLP) potential function is used for ligand-protein interaction energy which is done in subroutine '*eplp*'. Prior to that, subroutine '*precal*' identifies pairwise atomic

interaction types between the ligand and the protein. The parameters for the best optimal MOLS structure are stored in common variables '*opmo(maxstr,maxpar)*', '*emo(maxstr)*'.

Subroutine CONFORMATION (file: smconformation.f)

In this subroutine the conformation of the MOLS optimal ligand structure is generated making use of the common variables '*opmo(maxstr,maxpar)*', '*emo(maxstr)*' generated from subroutine '*MOLS*'. Then the MOLS optimal structure of the ligand is written in file '*mols.mol2*'.

Subroutine MINIMIZ (file: smminimiz.f)

Shell script '*minimize.sh*' will minimize the MOLS generated ligand structure in file '*mols.mol2* ' using Conjugate Gradient Minimization method. The minimized structure is then written into '*min.pdb*' file which is converted to .mol format as '*min.mol2*'. The intramolecular energy of the minimized ligand is found using MMFF94 force field and the energy values are written into file '*output2*'. The intramolecular energy of the minimized ligand, which is expressed in units of Kcal/mol, is added with the ligand-protein intermolecular PLP energy, which is expressed as a dimensionless quantity. The total energy is expressed as a dimensionless quantity. The final minimized structure of the ligand is the written in '*mini.mol2*' output file.

References

1. Viji SN, Prasad PA, Gautham N (2009) Protein- Ligand Docking Using Mutually Orthogonal Latin Squares (MOLSDOCK). J Chem Inf Model 49:2687–2694

2. Paul DS, Gautham N (2017) iMOLSDOCK: Induced-fit docking using mutually orthogonal Latin squares (MOLS). J Mol Graph Model 74:89–99 . doi: 10.1016/j.jmgm.2017.03.008

3. Arun Prasad P, Gautham N (2008) A new peptide docking strategy using a mean field technique with mutually orthogonal Latin square sampling. J Comput Aided Mol Des 22:815–829

4. Vengadesan K, Gautham N (2003) Enhanced sampling of the molecular potential energy surface using mutually orthogonal Latin squares: Application to peptide structures. Biophys J 84:2897

5. Sam Paul D, Gautham N (2018) Protein–small molecule docking with receptor flexibility in iMOLSDOCK. J Comput Aided Mol Des. doi: 10.1007/s10822-018-0152-8

6. Le Guilloux V, Schmidtke P, Tuffery P (2009) Fpocket: an open source platform for ligand pocket detection. BMC Bioinformatics 10:168

7. O'Boyle NM, Banck M, James CA, et al (2011) Open Babel: An open chemical toolbox. J Cheminformatics 3:33 . doi: 10.1186/1758-2946-3-33

8. Tuffery P, Etchebest C, Hazout S (1997) Prediction of protein side chain conformations: a study on the influence of backbone accuracy on conformation stability in the rotamer space. Protein Eng 10:361–372

9. Cornell WD, Cieplak P, Bayly CI, et al (1995) A second generation force field for the simulation of proteins, nucleic acids, and organic molecules. J Am Chem Soc 117:5179–5197

10. Nemethy G, Gibson KD, Palmer KA, et al (1992) Energy parameters in polypeptides. 10. Improved geometrical parameters and nonbonded interactions for use in the ECEPP/3 algorithm, with application to proline-containing peptides. J Phys Chem 96:6472–6484

11. Betancourt MR, Skolnick J (2001) Finding the needle in a haystack: educing native folds from ambiguous ab initio protein structure predictions. J Comput Chem 22:339–353

12. Kříž Z, Carlsen PHJ, Koča J (2001) Conformational features of linear and cyclic enkephalins. A computational study. J Mol Struct THEOCHEM 540:231–250 . doi: http://dx.doi.org/10.1016/S0166-1280(00)00728-4