MULTIDOCK V1.0

Richard M. Jackson
Biomolecular Modelling Laboratory
Imperial Cancer Research Fund
44 Lincoln's Inn Fields
London WC2A 3PX

1.0 Description

The program MULTIDOCK was developed to provide a method for refining the interface between two proteins at the atomic level given an initial docked complex *e.g.* generated by a docking algorithm or manual docking procedure. The motivation for this work was to provide a rapid energy refinement protocol for the large number of putative docked complexes produced by rigid-body docking programs such as FTDOCK or DOCK. The program models the effects of side-chain conformational change and the rigid-body movement of the interacting proteins during refinement. The protein is described at the atomic level by electrostatic and van der Waals interactions in which the sidechains are modelled by a multiple copy representation according to a rotamer library on a fixed peptide backbone. Rigid-body energy minimization is then performed to relax the interface. Thus refinement of the interface between the two proteins is based on a two step process:

1) A probability based conformational matrix of the protein sidechains is refined iteratively by a mean field method. In which a given sidechain interacts with the fixed backbone and the probability weighted average of the surrounding sidechains.

2) The protein backbone atoms and the highest probability sidechain conformations from step (1) undergo rigid-body energy minimisation to relax the protein interface.

Steps (1) and (2) are repeated until convergence of the interaction energy between the two proteins.

A description of the program and its application to refinement of protein-protein interrelations in protease-protein inhibitor and antibody-protein antigen interfaces is given in Jackson *et al.* (1998).

2.0 Using MULTIDOCK

MULTIDOCK reads the ATOM record of a PDB file for a protein complex consisting of standard amino acid types (non-standard amino acid types are not acceptable). Output coordinates are also in PDB format, and a `pdb_output.out` file is also generated consisting of details of the refinement procedure and an analysis of the contribution of individual amino acids to the interaction enthalpy. The program is run from the unix command line as follows:

```
> multidock pdb_input pdb_output control_parameters
```

The first file is the PDB coordinate file of the complex to be refined, with chains separated by TER cards the second file is for the output PDB coordinates and the third file is the `control_parameters` for the refinement (described in section 2.3 below). A second output file

(described in section 2.4) `pdb_output.out` is automatically generated (unless specified in the `conrtol_parameters` file).

2.1 `pdb_input`

Given that the program uses a molecular mechanics force field to describe atomic interactions it is important that the ATOM records of the `pdb_input` file do not contain non-standard records, errors or missing atoms. It is the users responsibility to make sure this is the case. Common errors will result in the program being stopped or the user issued a warning. In the case where atom records are missing the user has the choice of (manually) re-building the relevant atoms or changing the PDB file to a valid residue type and renaming the residue to reflect the correct atom records e.g. pruning back the residue to Ala or Gly. The program defines the twenty standard amino acids with default charges at pH 7 in solution (i.e. Asp, Glu and the C-terminus are negatively charged; Arg, Lys and the N-terminus are positively charged). Histidine is by default defined as neutral, with both the Nδ1 and Nε2 protonated rotamers present in the mean field simulation. However, if it is required that it is positively charged the residue name in the ATOM records must be changed from HIS to HIP. The residues SER, THR, CYS(h) have an additional degree of freedom not defined in the side chain rotamer library involving the χ2 rotamer defined as Cα-Cβ-O(S)γ-Hγ. Hence three rotamers representing the *gauche*$^-$, *trans* and *gauche*$^+$ conformations are generated for all such residue rotamers in the mean field simulation. Similarly, for TYR the two *cis* and *trans* conformations of the χ6 rotamer defined as Cε1-Cζ-Oη-Hη are generated for all such residue rotamers. Cysteine residues are treated separately. The default option is to check if atoms of other cysteine residues are within defined distance criteria to assign connectivity for a disulphide bridge. Cysteine residues satisfying the distance criteria are renamed CYX to distinguish them from the chemically different free cysteine residue CYS. If however a cysteine residue fulfils the disulphide criteria with two or more cysteines then the program issues a warning message and the user must assign the cysteine connectivity in the `control_parameters` file (see Section 2.3.7). Furthermore, the charge status of the N- and C-termini of different chains must be considered. Chains are determined by the presence of a TER card separator (not the chain ID). The first and last residue and any residue before or after a TER card will be charged if possible! However, for this to occur an N-terminal residue must begin with a “N” atom and a C-terminal residue must have an “OXT” atom (i.e. a terminal carboxyl group). A warning is issued if these residues are not charged, since it is easy to overlook their assignment. The charging or otherwise of all different combinations of N- and C-termini can be controlled by i) the use of TER cards which can be removed turning two separate chains into one for the purpose of the calculation (and thus eliminating the N- and C-termini) ii) the ***nterm_neural*** keyword (see section 2.3.1) which assigns a neutral N-terminus to the appropriate subunit number or iii) removal of the “OXT” atom at the C-terminus since residues without it are not charged. Lastly, where the distance cut-off value ***cut_jface*** (see section 2.3.2) excludes residues (that are not in close proximity to the interface) from the calculation, the residues start with the “CA” atom as opposed to the “N” atom. This occurs in the fixed “buffer” regions and at the beginning of chains and where the ***nterm_neural*** keyword is used. The “CA” atom is used to distinguish between charged N-termini and ones generated by the cut-off or the ***nterm_neural*** keyword.

2.2 `pdb_output`

The PDB output formatted file consists of the ATOM records used in the simulation, updated according to the refinement procedure. The mainchain conformation of the receptor molecule is kept fixed throughout with the ligand molecule undergoing rigid-body movement. In cases where sidechains are described by a multiple copy representation, the final conformation represents the highest probability rotamer conformation. This rotamer could be the same as the parent structure or one from the rotamer

library.

2.3 `Control_parameters`

The `control_parameters` file controls the refinement of the protein complex. The protein subunits belonging to the protein receptor and protein ligand must be defined in order to define the interface to be refined (see 2.3.1 below). All other control_parameters are optional since default options exist. However, the refinement procedure can be controlled by specifying values for parameters that over-ride the program defaults. All options are defined by a keyword system with no particular keyword order.

2.3.1 Defining the protein receptor and ligand

The receptor and ligand must be defined in the `control_parameters` file. This is done by using the keywords ***imobile_mol*** (for receptor protein(s)) and ***mobile_mol*** (for ligand protein(s)) with reference to the subunit number in the `pdb_input` file. Independent subunits are defined by a TER card separator and not by chain identification records.

e.g. 1) In the case of a protein-protein inhibitor interaction the PDB file might sequentially consist of the protein (receptor) chain followed by the protein inhibitor (ligand) chain. Given these two subunits are separated by a TER card the system is defined as:

imobile_mol=1
mobile_mol=2

e.g. 2) In the case of an antibody-protein antigen interface interaction the PDB file might sequentially consist of two (receptor) antibody chains (heavy and light) and a protein antigen (ligand) which is a heterodimer consisting of two chains. Given that these subunits are separated by a TER card the system is defined as:

imobile_mol=1
imobile_mol=2
mobile_mol=3
mobile_mol=4

you are advised to look at the `pdb_output` file to check the intended interface is in fact the one that has been optimised!

Also of importance is the charge status of the N-terminal residues (see Section 2.1) in a particular chain. By default the first residue in the chain the N-terminus residue is charged (+1.0 eV). However, this can be overrided if necessary (e.g. if the N-terminal residues are not in the PDB file and the protein coordinates start mid chain, or there is a non-amino acid blocking group at the N-terminus). This is done by;

nterm_neutral= 1
nterm_neutral= 2

nterm_neutral, makes the N-terminus of chain 1 and 2 neutral (starts with a CA atom) as opposed to the default of being positively charged.

2.3.2 Defining the protein-protein interface

The intermolecular interface of interest is described by a region of sidechain mobility for residues at the interface modelled by a multiple copy representation, this is surrounded by a fixed (buffer) region modelled by the existing protein sidechain conformations.

These two inclusion regions are defined in terms of distance of the given residues Cβ (Cα for glycine) from the Cβ of any residue on the interacting protein. This is done using the keywords:

cut_iface=#

cut_iface is the distance cut-off (in Ångstroms) for inclusion in the region of the multiple copy sidechain rotamer representation (default=10.0Å)

cut_jface=#

cut_jface is the distance cut-off (in Ångstroms) for inclusion as a fixed sidechain representation. Residues beyond this distance cut-off are not included in the simulation (default=20.0Å).

2.3.3 Defining the parameters that control non-bonded interactions

The control of non-bonded interactions (which are identical in the mean field optimization and energy minimization steps) are defined using the following keywords:

cut_nbond=#

cut_nbond is the atom-atom non-bonded cut-off distance (in Ångstroms) for inclusion in the residue-residue interaction energy (default=10.0Å).

d=#

d is the effective dielectric constant, ε, (no units) in V = 332.0*qi*qj / rij*ε (default=4.0)

eatmax=#

eatmax is the maximum van der Waals atom-atom interaction energy (in kcal/mol). Values above which are set to ***eatmax*** (default=2.5 kcal/mol).

cut_xx=#

cut_xx is the distance of closest approach in (in Ångstroms) for Heavy-Heavy atom contacts for evaluation of electrostatic interaction energy. Values below this are set to ***cut_xx*** (default=3.0Å)

cut_xh=#

cut_xh is the distance of closest approach for Heavy-Hydrogen atom contacts for evaluation of electrostatic interaction energy. Values below this are set to ***cut_xh*** (default=2.0Å)

cut_hh=#

cut_hh is the distance of closest approach for Hydrogen-Hydrogen atom contacts for evaluation of electrostatic interaction energy. Values below this are set to ***cut_hh*** (default=1.0Å)

2.3.4 Defining the parameters specific to mean field optimisation

Control of parameters specific to the mean field optimisation.

cut_res_nb=#

cut_res_nb is the residue-residue non-bonded cut-off distance (in Ångstroms) between adjacent Cβ atoms for inclusion in the sidechain residue-residue non-bonded pair list (default=15.0Å).

temp=#

temp is the temperature (in degrees Kelvin). This determines the value of RT used in ensemble optimisation (default=298.0 K).

lamda =#

lamda is the value for 'memory' of previous probability matrix. The value of lamda ranges from 0.0 to 1.0 the smaller the value of lamda the smaller the 'memory' of previous probability matrix (default=0.5).

rmsmax=#

rmsmax determines the convergence criteria in terms of the r.m.s. change in the probability Matrix. When rmsMAT is below ***rmsmax*** the probability matrix is deemed to have converged (default=1e-3).

emax=#

emax determines the convergence criteria in terms of change in Energy (in kcal/mol). When the change in energy from the previous step is below emax the energy is deemed to have converged (default=0.5 kcal/mol)

Note: either ***emax*** or ***rmsmax*** can be used to control the convergence of the mean field optimization. However, a minimum of ten cycles of optimisation are performed to guard against choice of inappropriate values.

2.3.5 Defining the parameters specific to rigid-body energy minimization

Parameters specific to rigid-body energy minimization

cut_lface=#

cut_lface is the residue-residue Cβ (Cα for glycine) distance cut-off (in Ångstroms) across the interface for inclusion of the residue in the calculation of non-bonded interaction between the two rigid molecules. Residues beyond this distance cut-off are not included in the energy minimization. The value should be greater than or equal to cut_res_nb the mean field residue-residue non-bonded cut-off so all residue interactions included in the mean field calculation are also included in the minimization (default=15.0Å).

stepmax=#

stepmax is the maximum allowed translation (in Ångstroms/per step), moved according to the numerical derivatives of the energy function (default=0.3Å/step).

thetamax=#

thetamax is the maximum allowed rotation (in degrees/ per step) (default=1.0º/step).

ftol=#

ftol determines the convergence criteria for energy minimization in terms of change in energy between two successive steps (in kcal/mol) (default=1e-6 kcal/mol/step).

2.3.6 Controlling the output

By default output information about the progress of the mean field optimisation and rigid-body energy minimisation is printed (1) to the screen and (2) to the `pdb_output.out` file. But this can be turned off using the following keywords.

screen_output_off

screen_output_off turns off printing (on the progress of the simulation) to the screen.

file_output_off

file_output_off prevents output of the `pdb_output.out` file.

By default detailed information about the van der Waals parameters and point charges assigned to atoms is not printed out. However, initially it may be useful to check this information.

atom_params_on

atom_params_on turns on printing of van der Waals and point charge information for each atom to the output `pdb_output.out` file.

2.3.7 Defining the disulphide bond connectivity

define_disulphides

define_disulphides allows the user to define the disulphide linkages rather than use the default distance constraints to generate the connectivity. This can be important when according to the default option a given disulphide satisfies the constraints to form more than one disulphide. *e.g.*

```
WARNING!!! auto_def: CYX I  20 makes more than one Disulphide bond
```

Thus the disulphides can be defined in the `control_parameter` file (format (A5,2(2X,A3,1X,A1,I4)) by the uppercase keyword DISUL by residue name, chain identity and residue number for the two Cyx residues bonded (see end of PDB2 output file) *e.g.*

```
DISUL  CYX I  20  CYX I  27
```

Note: even when the disulphides are assigned with the ***define_disulphides*** keyword there will still be a warning message about the automatic generation of disulphide connectivity.

2.3.8 Example of a control_parameters file

The following is an example of a control_parameters file constructed for trypsin complexed with the inhibitor from bitter gourd (pdb1mct.ent). In this case the disulphide bond connectivity must be defined by the user using the ***define_disulphides*** keyword since three of the inhibitor CYX residues can form more than one disulphide bond based on the default distance constraints used to generate connectivity.

```
# sample params.dat
# -----------------
#
# # at beginning of line indicates a comment line.
# The variable names MUST be in lower case, except
# those defining the disulphide bonds which must have
# the format (A5,2(2X,A3,1X,A1,I4)) and start with the
# upper case 'DISUL' e.g. DISUL  CYX A  42  CYX A  58
# defining the chain_id and residue number.
#
imobile_mol=1
mobile_mol=2
#nterm_neutral=1
#nterm_neutral=2
temp= 298.0
cut_jface= 200.0
cut_atom_nb= 10.0
cut_res_nb=16.0
cut_lface=16.0
ftol = 0.0001
dielectric= 4.0
eatmax= 3.0
emax=0.4
atom_params_on
#file_output_off
#screen_output_off
define_disulphides
DISUL  CYX E  22  CYX E 157
DISUL  CYX E  42  CYX E  58
DISUL  CYX E 136  CYX E 201
DISUL  CYX E 168  CYX E 182
DISUL  CYX E 191  CYX E 220
DISUL  CYX I   3  CYX I  20
DISUL  CYX I  10  CYX I  22
DISUL  CYX I  16  CYX I  27
```

2.4 pdb_output.out

This file gives information on the progress of the simulation during refinement including both mean field optimisation and energy minimisation steps. In this case α-chymotrypsin complexed with turkey ovomucoid third domain (pdb1cho.ent) is used as an example. Details include the following:
Cysteine residues in a disulphide bridge (as defined by given distance criteria, see Jackson *et al*. 1998

for further details). *e.g.*

```
Disulphide bond information defined using internal distance constraints

 CYX E  42 - CYX E  58
 CYX E 136 - CYX E 201
 CYX E 168 - CYX E 182
 CYX E 191 - CYX E 220
 CYX I   8 - CYX I  38
 CYX I  16 - CYX I  35
 CYX I  24 - CYX I  56
```

This default option can be overrided by the ***define_disulphides*** keyword if ambiguity exists using the distance criteria (see section 2.3.7 above).

If the ***atom_params_on*** keyword is used then detailed information about the atom parameters is printed to the `pdb_output.out` file. In addition to the standard Protein Data Bank ATOM records, six additional columns are included detailing the residue number in the simulation, the rotamer number (which starts at 2), the point charge on the atom, the van der Waals A_{ii} and B_{ii} parameters and lastly the molecule number. *e.g.*

```
ATOM    230  N   THR E  37A     -4.181   2.772   3.362  18   2 -0.400  735.3 24.25   1
ATOM    231  CA  THR E  37A     -2.744   2.521   3.158  18   2  0.000  769.7 21.49   1
ATOM    232  C   THR E  37A     -2.047   3.436   2.175  18   2  0.550  888.8 24.81   1
ATOM    233  O   THR E  37A     -0.892   3.151   1.759  18   2 -0.550  480.2 20.72   1
ATOM    234  CB  THR E  37A     -2.516   0.993   2.889  18   2  0.000  769.7 21.49   1
ATOM    235  OG1 THR E  37A     -3.221   0.634   1.680  18   2 -0.490  500.2 19.68   1
ATOM    236  CG2 THR E  37A     -3.125   0.172   4.034  18   2  0.000 1586.4 35.05   1
ATOM    237  HN  THR E  37A     -4.794   2.316   2.717  18   2  0.400    9.1  1.60   1
ATOM    237  HA  THR E  37A     -2.246   2.789   4.078  18   2  0.000    0.0  0.00   1
ATOM    237  HB  THR E  37A     -1.457   0.797   2.806  18   2  0.000    0.0  0.00   1
ATOM    238  HG1 THR E  37A     -2.862   1.177   0.921  18   2  0.490    9.1  1.60   1
ATOM    230  N   THR E  37B     -4.181   2.772   3.362  18   3 -0.400  735.3 24.25   1
ATOM    231  CA  THR E  37B     -2.744   2.521   3.158  18   3  0.000  769.7 21.49   1
ATOM    232  C   THR E  37B     -2.047   3.436   2.175  18   3  0.550  888.8 24.81   1
ATOM    233  O   THR E  37B     -0.892   3.151   1.759  18   3 -0.550  480.2 20.72   1
ATOM    234  CB  THR E  37B     -2.516   0.993   2.889  18   3  0.000  769.7 21.49   1
ATOM    235  OG1 THR E  37B     -3.221   0.634   1.680  18   3 -0.490  500.2 19.68   1
ATOM    236  CG2 THR E  37B     -3.125   0.172   4.034  18   3  0.000 1586.4 35.05   1
ATOM    237  HN  THR E  37B     -4.794   2.316   2.717  18   3  0.400    9.1  1.60   1
ATOM    237  HA  THR E  37B     -2.246   2.789   4.078  18   3  0.000    0.0  0.00   1
ATOM    237  HB  THR E  37B     -1.457   0.797   2.806  18   3  0.000    0.0  0.00   1
ATOM    238  HG1 THR E  37B     -3.082  -0.340   1.499  18   3  0.490    9.1  1.60   1
```

etc . . .

For each full cycle of (1) mean field optimisation and (2) energy minimisation the following output is given

(1) Mean field optimisation

i) The total number of sidechain-sidechain interactions and a summary of convergence in the mean field optimisation in terms of rmsMAT (the root mean squared deviation in the matrix from the previous step) and the change in the total energy (which is given by the probability weighted sum of sidechain rotamers). *e.g.*

```
Matrix optimisation...
 Total no. of sidechain-sidechain interactions                    71456
 Total no. of solvent sites                                           0
 No. of solvent sites incompatible with sidechain rotamers            0
Cycle No.   1  rmsMAT =  3.085201  Energy=     387.8
Cycle No.   2  rmsMAT =  1.778450  Energy=    -127.3
Cycle No.   3  rmsMAT =  1.070900  Energy=    -412.1
Cycle No.   4  rmsMAT =  1.069327  Energy=    -562.8
Cycle No.   5  rmsMAT =  0.662776  Energy=    -654.8
Cycle No.   6  rmsMAT =  0.398871  Energy=    -704.4
Cycle No.   7  rmsMAT =  0.281486  Energy=    -730.4
Cycle No.   8  rmsMAT =  0.220360  Energy=    -743.9
Cycle No.   9  rmsMAT =  0.168179  Energy=    -751.1
Cycle No.  10  rmsMAT =  0.122364  Energy=    -754.9
Cycle No.  11  rmsMAT =  0.086387  Energy=    -757.0
Cycle No.  12  rmsMAT =  0.059732  Energy=    -758.2
Cycle No.  13  rmsMAT =  0.040518  Energy=    -758.9
Cycle No.  14  rmsMAT =  0.026989  Energy=    -759.2
```

ii) A description of the final probability matrix detailing each residue rotamer with a listing of the sidechain internal energy, sidechain-backbone interaction energy, the probability weighted sidechain-sidechain interaction energy, the total sidechain energy and the probability of rotamer occupancy. *e.g.*

Einternal	Ebbone	Esidechain	Esol	Etotal	Prob(i,j)	i	j
2.726	-6.308	-1.705	0.000	-5.287	1.000000	2	2
0.186	-1.886	-1.794	0.000	-3.493	1.000000	3	2
0.983	-0.635	-1.143	0.000	-0.795	1.000000	4	2
0.000	0.000	0.000	0.000	0.000	1.000000	5	2
1.574	1.910	-3.610	0.000	-0.125	1.000000	6	2
0.821	-4.027	-5.098	0.000	-8.304	1.000000	7	2
0.000	0.000	0.000	0.000	0.000	1.000000	8	2
2.726	-4.145	-7.509	0.000	-8.927	1.000000	9	2
2.726	-6.197	-4.755	0.000	-8.225	1.000000	10	2
1.426	-4.261	-4.283	0.000	-7.118	1.000000	11	2
1.015	-1.693	-1.809	0.000	-2.487	1.000000	12	2
0.845	0.587	-3.954	0.000	-2.522	0.980503	13	2
0.845	0.398	1.999	0.000	3.242	0.000080	13	3
0.845	-0.523	-0.494	0.000	-0.171	0.019418	13	4
0.500	-4.005	-0.124	0.000	-3.628	0.270989	14	2
0.500	-4.368	0.088	0.000	-3.780	0.353371	14	3
0.754	-4.530	2.836	0.000	-0.940	0.002924	14	4
1.164	-2.913	-2.093	0.000	-3.842	0.363190	14	5
1.495	0.788	9.711	0.000	11.993	0.000009	14	6
1.906	-1.018	-2.580	0.000	-1.693	0.009509	14	7
2.076	1.321	18.372	0.000	21.770	0.000009	14	8
0.733	-3.260	-5.508	0.000	-8.035	1.000000	15	2
1.123	-13.312	-5.419	0.000	-17.608	0.994882	16	2
0.354	-3.522	-5.270	0.000	-8.438	0.000015	16	3
0.713	-6.154	-1.776	0.000	-7.217	0.000015	16	4
1.123	-10.197	-5.407	0.000	-14.481	0.005087	16	5

Important Note: Initially the crystallographically observed rotamer is placed at rotamer No. 2. In the next cycle of Mean Field Optimisation The highest probability rotamer from this cycle will be rotamer No. 2 (and rotamer No. 2 will swap places with the highest probability rotamer). e.g. above, residue i = 14, rotamer j= 5 (the highest probability rotamer) will become j=2 in the next cycle and rotamer j= 2 will become j=5. However, this makes comparison of changes in probability for a particular rotamer difficult.

iii) A summary of the probability weighted average energies for a given residue plus the net charge assigned to individual residues and an overall energy summary *e.g.*

```
ILE   16 (   2)  -5.287 kcal/mol Charge= 0.0
VAL   17 (   3)  -3.493 kcal/mol Charge= 0.0
ASN   18 (   4)  -0.795 kcal/mol Charge= 0.0
GLY   19 (   5)   0.000 kcal/mol Charge= 0.0
GLU   20 (   6)  -0.125 kcal/mol Charge=-1.0
GLU   21 (   7)  -8.304 kcal/mol Charge=-1.0
ALA   22 (   8)   0.000 kcal/mol Charge= 0.0
TRP   27 (   9)  -8.927 kcal/mol Charge= 0.0
TRP   29 (  10)  -8.225 kcal/mol Charge= 0.0
GLN   30 (  11)  -7.118 kcal/mol Charge= 0.0
VAL   31 (  12)  -2.487 kcal/mol Charge= 0.0
SER   32 (  13)  -2.476 kcal/mol Charge= 0.0
LEU   33 (  14)  -3.733 kcal/mol Charge= 0.0
GLN   34 (  15)  -8.035 kcal/mol Charge= 0.0
ASP   35 (  16) -17.592 kcal/mol Charge=-1.0
```
etc . . .

```
Sidechain Energy (sidechain, backbone & solvent) =     -759.2
Backbone-backbone Energy =      -41.0
Solvation Energy (component) =        0.0
Total Energy =     -800.2
```

(2) Rigid-body energy minimisation

A summary of convergence in the rigid-body energy minimization across the interface, in terms of the total energy, the electrostatic energy component only (QEnergy) and the number of atom-atom vdW clashes (bumps) whose energies are capped by ***eatmax***. *e.g.*

```
Performing rigid-body steepest decents minimization...
Cycle no. 1 Energy=  -91.866 QEnergy=  -27.190 kcal/mol bumps   1
Cycle no. 2 Energy=  -92.424 QEnergy=  -27.819 kcal/mol bumps   1
Cycle no. 3 Energy=  -92.878 QEnergy=  -28.358 kcal/mol bumps   1
Cycle no. 4 Energy=  -92.881 QEnergy=  -28.364 kcal/mol bumps   1
Cycle no. 5 Energy=  -93.047 QEnergy=  -28.527 kcal/mol bumps   1
Cycle no. 6 Energy=  -93.154 QEnergy=  -28.646 kcal/mol bumps   1
```

A summary of the residue-residue interaction energy across the interface in terms of the total energy, van der Waals energy, electrostatic energy and the number of bumps (the number of atom-atom interactions with a vdW energy capped by ***eatmax***) for each residue in the interface (as defined by ***cut_lface***). In addition the summed total interface interaction energy is given. *e.g.*

```
LYS   13 Res Energy=  -0.7 Vdw Energy=  -0.4 Q Energy=  -0.3 kcal/mol bumps   0
PRO   14 Res Energy=  -2.5 Vdw Energy=  -1.8 Q Energy=  -0.7 kcal/mol bumps   0
ALA   15 Res Energy=  -2.9 Vdw Energy=  -2.4 Q Energy=  -0.5 kcal/mol bumps   0
CYX   16 Res Energy=  -4.0 Vdw Energy=  -2.2 Q Energy=  -1.9 kcal/mol bumps   0
THR   17 Res Energy=  -3.3 Vdw Energy=  -3.4 Q Energy=   0.1 kcal/mol bumps   0
LEU   18 Res Energy=  -9.1 Vdw Energy=  -6.1 Q Energy=  -3.0 kcal/mol bumps   1
GLU   19 Res Energy=  -4.3 Vdw Energy=  -4.0 Q Energy=  -0.3 kcal/mol bumps   0
TYR   20 Res Energy=  -4.8 Vdw Energy=  -4.6 Q Energy=  -0.2 kcal/mol bumps   0
ARG   21 Res Energy=  -6.1 Vdw Energy=  -2.9 Q Energy=  -3.1 kcal/mol bumps   0
PRO   22 Res Energy=  -0.1 Vdw Energy=  -0.2 Q Energy=   0.1 kcal/mol bumps   0
LEU   23 Res Energy=   0.0 Vdw Energy=   0.0 Q Energy=   0.0 kcal/mol bumps   0
```
etc . . .
```
Total Energy=  -93.154  kcal/mol
```

3.0 Description of Program and input/output

The overall program consist of several independent programs which are linked by a shell script to perform the interface optimisation on a PDB file given a `control_parameters` file. These programs are independent and can be run independently if desired. The expected input/output files are given by typing the name of the program. Only the program `hydtor` essentially runs independently of the other programs to 1) add hydrogen atoms to a PDB file or 2) generate a file containing a list of torsion angles for backbone and side-chains, depending on the input flags. The program is part of `PREPI` (Protein REPresentations Interactively: distributed by Islam & Sternberg, Imperial Cancer Research Fund, London, UK), arround which the program was originally constructed. The program `mcopy` initially generates the multiple copy representation of side chains given a rotamer library of torsion angles `ROT.LIB` and library of side chain torsion angle definitions `AATOR.DAT`. Other programs generally require output from earlier programs. The programs `rearrange` and `torpdb` are just format manipulation routines, that allow the programs to process information from the program `PREPI`. The program `rotenergy` calculates the rotamer internal energies given a list of dihedrals for side chain rotamers and the rotamer occupancies (taken from the library `ROT.OCC`). Finally i) the modified PDB file with multiple copy representation of interface side-chains with added hydrogens ii) the `control_parameters` file (controlling parameters in the simulation) and iii) the internal energies of side-chain rotamers are all read into the program `multidock_main`. This program performs the mean field optimisation of side-chains followed by rigid-body energy minimisation of the interacting subunits. More specifically the routines are run in the following order as follows:

3.1 `mcopy` Correct usage: `mcopy pdb_input pdb_output control_parameters AATOR.DAT ROT.LIB`

The program expects 5 command line arguments.

The program takes ATOM records from a PDB formatted file `pdb_input` and generates a PDB formatted file `pdb_output` including multiple copies of side-chains in the specified mobile interface region (see keyword ***cut_iface***) and single copies of sidechains in the fixed (buffer) region (see ***keyword cut_jface)*** modelled by the existing protein sidechain conformations. Values for the keywords are specified in `control_parameters` file (otherwise default values are used). The file `AATOR.DAT` must be specified and defines the atom names for allowed residue types involved in a given dihedral angle ($\chi 1$, $\chi 2$ *etc*). The file `ROT.LIB` must be specified and defines the values of the side-chain dihedral angles of the rotamers generated in addition to the experimental side-chain conformation.

3.2 `hydtor` Correct usage: `hydtor -ipdb_input -opdb_output -dHYDROGENS.DAT -random > hydrogen_info`

The program expects 4 command line arguments. `hydrogen_info` is optional (otherwise the information will be printed to the screen).

The program takes ATOM records in a PDB formatted file `pdb_input` generated by `mcopy` (i.e. the `pdb_output` file from `mcopy`) and generates a `pdb_output` file with added hydrogens. The dictionary `HYDROGENS.DAT` contains information about hydrogen generation for allowed residue types, including, atom bonding information, bond lengths and angles for adding hydrogen atoms to the PDB file. The `-random` flag creates multiple hydrogen positions about the freely rotatable dihe-

dral angles of Ser, Thr, Cys and Tyr. The `-i`, `-o` and `-d` flags are required to denote the `pdb_input`, `pdb_output` and `HYDROGENS.DAT` dictionary file respectively. Information about the hydrogen atom generation is generally directed to a file `hydrogen_info`, otherwise it is writen to the screen.

3.3 `rearrange` Correct usage: `rearrange pdb_input pdb_output`

The program expects two command line arguments.

The program takes ATOM records in a PDB formatted file `pdb_input` generated by `hydtor` (i.e. the `pdb_output` file from `hydtor`) and generates a `pdb_output` file in the correct format for input into `multidock_main`.

3.4 `torpdb` Correct usage: `torpdb pdb_input pdb_output`

The program expects two command line arguments.

The program takes ATOM records in a PDB formatted file `pdb_input` generated by `rearrange` (i.e. the `pdb_output` file from `rearrange`) and generates a `pdb_output` file in the correct format for input into `hydtor -tor` (which calculates rotamer side-chain dihedral angles in the PDB file).

3.5 `hydtor -tor` Correct usage: `hydtor -tor -ipdb_input -otorsion_output > torsion_angle_info`

The program expects three command line arguments. `torsion_angle_info` is optional (otherwise the run information will be printed to the screen).

The program takes ATOM records in a PDB formatted file `pdb_input` generated by `torpdb` (i.e. the `pdb_output` file from `torpdb`) and generates a `torsion_output` file which contains formatted information about the side-chain (and mainchain) dihedral angles of the rotamers present. The `-tor` flag is required to denote the torsion option the `-i and -o` flags are required to denote the `pdb_input` and `torsin_output` files respectively.

3.6 `rotenergy` Correct usage: `rotenergy torsion_input tor_energy_output ROT.OCC`

The program expects three command line arguments.

The program takes a file which contains formatted information about the side-chain dihedral angles generated by `hydtor -tor` (i.e. the `torsion_output` file from `hydtor -tor`) and generates a `tor_energy_output` file which contains formatted information about the side-chain rotamer internal energies. The file `ROT.OCC` must be specified and defines the occupancies (i.e. as a fraction of 1.0) of the side-chain rotamers (as defined by the $\chi 1$, $\chi 2$ ... dihedral angles). The program takes into account rotational symmetry about symmetric Tyr, Phe, Asp and Glu dihedral angles.

3.7 `multidock_main` Correct usage: `multidock_main pdb_input pdb_output pdb_output.out control_parameters tor_energy_input`

The program expects five command line arguments.

The program takes ATOM records in a PDB formatted file `pdb_input` generated by `rearrange` (i.e. the `pdb_output` file from `rearrange`) and generates a `pdb_output` file following mean field optimisation and rigid-body minimisation. The `pdb_output.out` file gives details of the simulation (described in section 2.4). The definition of the ligand and receptor chains (compulsory) and the values for the other keywords (optional) are specified in the `control_parameters` input file (described in section 2.3). The `tor_energy_input` file contains formatted information about the side-chain rotamer internal energies generated in `rotenergy` (i.e. the `tor_energy_output` file from `rotenergy`).

4.0 Dimensions

At the moment the program can simulate up to 450 residues with a maximum of 22 side chain rotamers per residue. However, these are not the only limitation on the program dimension. The total number of non-zero energy sidechain-sidechain interactions is also limited to 300,000. However, judicious choice of cut_iface (in particular) and cut_jface, should mean this limitation is not a problem. Also of technical importance is the limitation on the number of non-zero energy rotamer-rotamer interactions for a particular side chain rotamer which is limited to 1000. Note: this essentially limits the number of side chain rotamers per residue. However, for practical reason this value is controlled by cut_res_nb (the residue-residue non-bonded cut-off for inclusion in the sidechain residue-residue non-bonded pair list) and cut_nbond the atom-atom non-bonded cut-off distance, since the limitation is based on the number of non-zero energy rotamer interactions. Thus as the cut_nbond is increased the numbers of non-zero contributions in the pair list will also increase.

The present release does not include the “soft” sphere Langevin dipole solvation contribution described in Jackson et al., 1998.

5.0 Additional Comments

i) The proline residue is incorrectly charged if at the N-terminus of a chain. If the residue is assigned as neutral the charge imbalance is 0.1 electron volts. This is due to the fact all N-terminus residues defined as a result of the cut-off or by the ***nterm_neutral*** keyword start at Cα instead of N. If the N-terminus of the proline residue is assigned as net positively charged the charge imbalance is 0.8 electron volts (hence beware!). This problem will be addressed in a later release.

ii) MULTIDOCK is writen in standard Fortran77, however, has only been tested on Silicon Graphics Inc. computers under Irix 5.3 and above. Please report bugs and other “undocumented features” to the author at jackson@biochem.ucl.ac.uk.

6.0 Reference

Jackson R.M., Gabb, H.A. & Sternberg, M.J.E. (1998) “Rapid Refinement of Protein Interfaces Incorporating Solvation: Application to the Docking Problem.” *J. Mol. Biol.*, **276**(1), 265-285.